

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

**BİR MÜHENDİSLİK PROBLEMİ İÇİN KAOTİK ÇOKLU SÜRÜ
PARÇACIK SÜRÜ OPTİMİZASYONU: KAYNAKLI KİRİŞ
TASARIMI**

Shahad Odah Feneaker FENEAKER

**Danışman
Jüri Üyesi
Jüri Üyesi**

**Dr. Öğr. Üyesi Kemal AKYOL
Dr. Öğr. Üyesi Yasemin GÜLTEPE
Dr. Öğr. Üyesi Ümit ATİLA**

**YÜKSEK LİSANS TEZİ
MÜHENDİSLİK YÖNETİMİ ANA BİLİM DALI**

KASTAMONU – 2019

TEZ ONAYI

Shahad Odah Feneaker FENEAKER tarafından hazırlanan "**Bir Mühendislik Problemi İçin Kaotik Çoklu Sürü Parçacık Sürü Optimizasyonu: Kaynaklı Kiriş Tasarımı**" adlı tez çalışması aşağıdaki jüri üyeleri önünde savunulmuş ve **oy birliği** ile Kastamonu Üniversitesi Fen Bilimleri Enstitüsü **Mühendislik Yönetimi Ana Bilim Dalı**'nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Danışman

Dr. Öğr. Üyesi. Kemal AKYOL
Kastamonu Üniversitesi



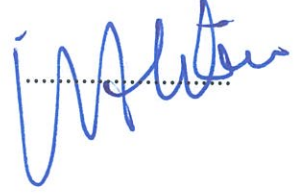
Jüri Üyesi

Dr. Öğr. Üyesi Yasemin GÜLTEPE
Kastamonu Üniversitesi



Jüri Üyesi

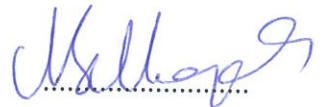
Dr. Öğr. Üyesi Ümit ATİLA
Karabük Üniversitesi



01/11/2019

Enstitü Müdürü

Doç. Dr. Nur BELKAYALI



TAAHHÜTNAME

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, ayrıca tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildirir ve taahhüt ederim.

Shahad Odah Feneaker FENEAKER



ÖZET

Yüksek Lisans Tezi

BİR MÜHENDİSLİK PROBLEMİ İÇİN KAOTİK ÇOKLU SÜRÜ PARÇACIK SÜRÜ OPTİMİZASYONU: KAYNAKLI KİRİŞ TASARIMI

Shahad Odah Feneaker FENEAKER
Kastamonu Üniversitesi
Fen Bilimleri Enstitüsü
Mühendislik Yönetimi Ana Bilim Dalı

Danışman: Dr. Öğr. Üyesi Kemal Akyol

Tasarım optimizasyonu önemli bir mühendislik tasarım etkinliğidir. Genel olarak, tasarım optimizasyonu kısıtlamaları yerine getirerek amaç fonksiyonunu optimize edecek şekilde tasarım değişkenleri için gerekli değerleri belirler.

Parçacık Sürü Optimizasyonu algoritması, yerel arama ve küresel arama arasında dengesizlik yaşar. Toplantı Odası yaklaşımı, Parçacık Sürü Optimizasyon algoritmasını iyileştirmek için çok sürülü bir model olarak öne sürülmüştür. Ancak, Çoklu Sürü Parçacık Sürü Optimizasyonu olarak adlandırılan bu yaklaşımda algoritma kötü bir pozisyondan başlayabilir. Bu sebeple algoritma yavaş bir yakınsamaya sahip olabilir. Kaotik lojistik haritası ile oluşturulan bir pozisyon kullanılarak bu sorun aşılabilmektedir.

Bir mühendislik problemi olan Kaynaklı Kiriş Tasarımı temel olarak, yükleme yükü, kayma gerilmesi, eğilme gerilmesi ve son sapma üzerindeki kısıtlamalara bağlı olarak kiriş maliyetinin en aza indirilmesini amaçlar. Bu tezde amaç, bu problemin çözümünde Kaotik Çoklu Sürü Parçacık Sürü Optimizasyonu algoritmasının performansını değerlendirmektir. Farklı sürü boyutları ve yineleme sayıları ile deneysel çalışmalar gerçekleştirilmiştir. Elde edilen sonuçlara göre, Kaotik Çoklu Sürü Parçacık Sürü Optimizasyonu algoritması hızlı bir yakınsama süreci sergileyerek iyi bilinen diğer algoritmalara kıyasla daha iyi performans göstermiştir.

Anahtar Kelimeler: Mühendislik Tasarım Problemi, Kaynaklı Kiriş Tasarımı, Optimizasyon, Kaotik Çoklu Sürü Parçacık Sürü Optimizasyonu.

2019, 72 Sayfa
Bilim Kodu: 90

ABSTRACT

MSc. Thesis

CHAOTIC MULTI-SWARM PARTICLE SWARM OPTIMIZATION FOR AN ENGINEERING PROBLEM: WELDED BEAM DESIGN

Shahad Odah Feneaker FENEAKER
Kastamonu University
Institute Of Science
Department of Engineering Administration

Supervisor: Assist. Prof. Dr. Kemal Akyol

Design optimization is an important engineering design activity. In general, design optimization determines the values required for design variables by satisfying constraints to optimize the objective function.

Particle Swarm Optimization algorithm suffers unbalanced local search and global search. Meeting Room approach was introduced as a multi-swarm model to improve the Particle Swarm Optimization algorithm. However, in this approach called Multiple Swarm Particle Swarm Optimization, the algorithm can start from a bad position. Therefore, the algorithm may have a slow convergence. This problem can be overcome by using a position created with a chaotic logistics map.

Welded Beam Design, which is an engineering problem, mainly aims to minimize the beam cost due to constraints on loading load, shear stress, bending stress and final deflection. The purpose of this thesis is to evaluate the performance of the Chaotic Multiple Swarm Particle Swarm Optimization algorithm in solving this problem. Experimental studies were carried out with different swarm sizes and iteration numbers. According to the results obtained, the Chaotic Multiple Swarm Particle Swarm Optimization algorithm showed a fast convergence process and performed better than other well-known algorithms.

Key Words: Engineering Design Problem, Welded Beam Design, Optimization, Chaotic Multi-Swarm Particle Swarm Optimization.

2019, 72 Pages
Science Kod: 90

TEŞEKKÜR

Bu çalışmaya büyük katkıda bulunan aşağıdaki kişilere yürekten teşekkür ediyorum ve minnettarlığımı sunuyorum. Öncelikle, engin bilgi birikimi ve yaratıcı düşüncesiyle beni yönlendiren danışmanım Dr. Öğr. Üyesi Kemal AKYOL'a derin ve içten teşekkürlerimi ifade etmek istiyorum. Onun gözetiminde çalışmak benim için her zaman büyük bir değer olmuştur. Detaylı ve yapıcı yorumları ve bu çalışma boyunca yaptığı kritik destek için derinden minnettarım. Yüksek Lisans Projemin değerlendirme kurulu üyeleri olan Dr. Öğr. Üyesi Yasemin GÜLTEPE ve Dr. Öğr. Üyesi Ümit ATİLA'ya sıcak ve içten teşekkürlerimi sunmak isterim; Sunumum sırasında vermiş olduğunuz destek, rehberlik ve önerilerinizi derinden takdir ediyorum. Ayrıca, araştırmalarını tamamlamalarına yardımcı olmak adına öğrencilere iyi bir ortam ve imkânlar sağladığı için Kastamonu Üniversitesi'ne teşekkür ediyorum. Son olarak, onların sürekli ve kararlılıkla beni cesaretlendirdikleri ve üniversitedeyken geçirdiğim süre boyunca bana sevgiyle yaklaşmak lütfunda bulundukları için muazzam minnettarlığım ailemedir. Sonsuz teşvikleri ve inançları bana ilham verdi, veriyor ve ilerde de verecek, bu sebeple onlara her zaman borçlu olacağım.

Shahad Odah Feneaker FENEAKER
2019

İÇİNDEKİLER

	Sayfa
TEZ ONAYI.....	ii
TAAHHÜTNAME	iii
ÖZET	iv
ABSTRACT	v
TEŞEKKÜR	vi
İÇİNDEKİLER.....	vii
TEKNİK TERİMLER.....	viii
ŞEKİLLER DİZİNİ	ix
TABLolar DİZİNİ	x
1. GİRİŞ	1
2. ARAŞTIRMANIN AMACI	4
3. LİTERATÜR TARAMASI	5
4. OPTİMİZASYON ALGORİTMALARI.....	10
4.1. Ateş Böceği Algoritması.....	10
4.2. Guguk Kuşu Arama Algoritması.....	10
4.3. Çiçek Tozlaşma Algoritması.....	11
4.4. Yerçekimi Arama Algoritması	11
4.5. Çoklu Evren Optimizasyonu Algoritması.....	11
5. PARÇACIK SÜRÜ OPTİMİZASYONU VE ÇEŞİTLERİ.....	12
5.1. Hibrit Parçacık Sürü Optimizasyonu	14
5.2. Uyarlanır Parçacık Sürü Optimizasyonu	14
5.3. Çok Amaçlı Parçacık Sürü Optimizasyonu	14
5.4. Ayrık Parçacık Sürü Optimizasyonu	15
5.5. Çoklu Sürü Parçacık Sürü Optimizasyonu	15
6. KAYNAKLI KİRİŞ TASARIMI PROBLEMİ	18
6.1. Kaynaklı Kiriş Tasarımı.....	18
6.2. Kaynaklı Kiriş Tasarımı Problemi Çalışmaları	19
6.3. Kaynaklı Kiriş Tasarımı İçin Matematiksel Formülasyon	20
6.4. Kaotik Çoklu Sürü Parçacık Sürü Optimizasyonu	22
6.5. Kaynaklı Kiriş Tasarımı Problemi İçin Önerilen Yaklaşım.....	23
7. ELDE EDİLEN SONUÇLAR VE TARTIŞMA	27
7.1. Optimizasyon Problemleri için Kaotik Çoklu Sürü Parçacık Sürü Optimizasyonu	27
7.2. Kaynaklı Kiriş Tasarımı Problemi için Kaotik Çoklu Sürü Parçacık Sürü Optimizasyonu	33
8. SONUÇLAR VE GELECEKTEKİ ÇALIŞMALAR	37
KAYNAKLAR.....	39
EKLER.....	46
ÖZGEÇMİŞ.....	72

TEKNİK TERİMLER

TÜRKÇE

Kısaltmalar

PSO	Parçacık Sürü Optimizasyonu
EKPSO	Efendi-Köle Parçacık Sürü Optimizasyonu
KÇSPSO	Kaotik Çoklu Sürü Parçacık Sürü Optimizasyonu
KKT	Kaynaklı Kiriş Tasarımı
ÇSPSO	Çoklu Sürü Parçacık Sürü Optimizasyonu
ÇAPSO	Çok Amaçlı Parçacık Sürü Optimizasyonu
ABA	Ateş Böceği Algoritması
GKAA	Guguk Kuşu Arama Algoritması
ÇEO	Çoklu Evren Optimizasyonu
ÇTA	Çiçek Tozlaşma Algoritması
YAA	Yerçekimi Arama Algoritması
TO	Toplantı Odası

İNGLİZCE

Kısaltmalar

PSO	Particle Swarm Optimization
M-SPSO	Master-slave Particle Swarm Optimization
CMSPSO	Chaotic Multi-Swarm Particle Swarm Optimization
WBD	Welded Beam Design
MSPSO	Multi-swarm Particle Swarm Optimization
MOPSO	Multi-Objective Particle Swarm Optimization
FA	Firefly Algorithm
CSA	Cuckoo Search Algorithm
MVO	Multi Verse Optimizer
FPA	Flower Pollination Algorithm
GSA	Gravitational Search Algorithm
MR	Meeting Room

ŞEKİLLER DİZİNİ

	Sayfa
Şekil 5.1. PSO'nun bir tasviri.....	13
Şekil 5.2. PSO tabanlı TO yaklaşımı	16
Şekil 6.1. Kaynaklı Kiriş Tasarımı	19
Şekil 6.2. Lojistik haritanın çatallanması (iki kola ayrılması)	23
Şekil 6.3. Araştırma metodolojisinin akış şeması.....	24
Şekil 6.4. KÇSPSO'nun Çerçevesi.....	24
Şekil 7.1. Test fonksiyonları; a) Sphere b) Griewank c) Rastirign d) Ackley	29
Şekil 7.2. Sphere test fonksiyonu için algoritmaların yakınsaklıkları	31
Şekil 7.3. Griewank test fonksiyonu için algoritmaların yakınsaklıkları	31
Şekil 7.4. Rastirign test fonksiyonu için algoritmaların yakınsaklıkları	32
Şekil 7.5. Ackley test fonksiyonu için algoritmaların yakınsaklıkları	32
Şekil 7.6. Algoritmaların hata dağılımı grafiği	33
Şekil 7.7. Meta sezgisel algoritmaların performanslarının karşılaştırılması	34
Şekil 7.8. KKT problemi için algoritmaların yakınsama eğrileri	35

TABLÖLER DİZİNİ

	Sayfa
Tablo 7.1. Önerilen algoritmada kullanılan parametreler ve değerleri.	27
Tablo 7.2. Değerlendirme için kullanılan optimizasyon test fonksiyonları	28
Tablo 7.3. Karşılaştırma testi fonksiyonları için sonuçlar	29
Tablo 7.4. KKT problemi için algoritma simülasyon sonuçları	34
Tablo 7.5. KKT problemi için önerilen çalışmanın literatürdeki diğer çalışmalar ile karşılaştırılması	36
Tablo 7.6. KKT problem için en iyi, en kötü ve ortalama değerleri bakımından önerilen çalışmanın literatürdeki diğer çalışmalar ile karşılaştırılması	36

1. GİRİŞ

Hesaplamalı bir bilim dalı olarak optimizasyon, temel olarak herhangi bir soruna mümkün olan en iyi çözümleri üretmenin olası yollarını ele alır (Gen ve Yun, 2006; Keshtegar ve Hao, 2018) ve bu çözümler eldeki probleme dayalı sayısal değerler olarak gösterilebilir. Bu nedenle, optimizasyon işlemlerinin amacı, mevcut olan çok sayıda çözümden herhangi bir soruna en iyi çözümü sunmaktır. Hesaplamalı bilim alanı yakın zamanda minimum ağırlıkta uçak tasarımları, en uygun petrol rafine etme, en uygun füze yörüngeleri, karlı ticari faaliyetler, fiziksel bilimler, biyolojik bilimler, kimya bilimleri, mühendislik, işletme, ekonomi bilimi ve mimarlık gibi birçok bilimsel ve mühendislik alanında uygulama bulmuştur (Fadaee ve Radzi, 2012; Jayabarathi vd., 2017). Optimizasyon problemleri mevcut çeşitli algoritmalar veya teknikler kullanılarak çözülebilmektedir.

Yıldız, gezegen ve galaktik sistemler de dâhil olmak üzere tüm yaşam biçimleri doğaya karşı sorumludur ve doğanın önemli bir özelliği hem bilinen hem de bilinmeyen araçları kullanarak dengeyi koruyabilmesidir. Çünkü doğada en uygun arama kavramında belirgin özelliği olanlar karakterize edilebilir. Optimuma ulaşmak yaşamda temel bir kavramdır (Arango vd., 2013; Chow, 2014) ve en uygunu ararken belirli hedefler ve kısıtlamalar karşılanmalı ve tatmin edilmelidir (Aldwaik ve Adeli, 2014; Chen vd., 2014). Optimum ararken arama süreci bir optimizasyon problemi (Peng ve Ouyang, 2014; Smith vd., 2014) olarak temsil edilebilirken bazı bilgisayar ve mühendislik uygulamalarında genellikle bir probleme özgü bir amaç fonksiyon ile en uygun çözüm sunulur ve bu süreç azaltılabilir (Luo vd., 2013).

Sezgisel optimizasyon algoritmalarının avantajları son zamanlarda onları çekici kılmış olup (Beheshti ve Shamsuddin, 2013) bu algoritmalar gerçek dünyadaki birçok oluşumun simülasyonu için ve bir popülasyonun ortak davranışını öğrenmek için kullanılmıştır. Yıllar boyunca Guguk Kuşu Arama Algoritması, Harmoni Arama Algoritması, Meyve Sineği Optimizasyonu Algoritması ve Ateş Böceği Algoritması gibi çeşitli meta-sezgisel algoritmalar önerilmiş olup çeşitli mühendislik tasarım problemlerini çözmek için kullanılmıştır.

Bu tez çalışmasında önerilen yaklaşımın temeli olan Parçacık Sürü Optimizasyonu (PSO) algoritması, kuşların ve balıkların avlanmasından veya yaşama davranışından ilham alınarak geliştirilmiştir (Yang, 2010a). PSO, çeşitli karmaşık optimizasyon problemlerini çözmede oldukça yaygın bir şekilde kullanılmaktadır. İlk ortaya çıkışından bu yana farklı PSO algoritmaları geliştirilmiştir. Bu algoritmaların amacı, belirli optimizasyon sorunlarına daha iyi çözümler elde etmenin daha iyi yollarını bulmaktır. PSO'nun arama ve kullanma yetenekleri arasında bir denge sağlamak en iyi performans için gereklidir ve bu nedenle Salih ve Alsewari Toplantı Odası (TO) yaklaşımını çoklu sürü yaklaşımı olarak önermiştir. Çalışmalarında, yazarlar, Çoklu Sürü Parçacık Sürü Optimizasyonunu (ÇSPSO) önermiş ve bu algoritmanın hem büyük ölçekli hem de normal optimizasyon problemlerindeki performansını değerlendirmişlerdir (Salih ve Alsewari 2018).

Değişkenlerin tasarım aşamasında ele alınması çoğu zaman bir sorundur ve çoğu geleneksel optimizasyon algoritmasının bir kısıtlaması olarak süreksiz ve farklılaşamayan problemleri başarılı bir şekilde çözemezler. Bu nedenle, süreksiz ve farklılaşamayan problemleri çözmek için meta-sezgisel algoritmalar yaygın olarak kullanılmaktadır (Džugan vd., 2018; Towler ve Sinnott, 2013).

Bu tez çalışmasında, bir optimizasyon problemi olan Kaynaklı Kiriş Tasarımı (KKT) problemini simüle etmek için yeni önerilen ÇSPSO'nun uygulanabilirliği araştırılmıştır. KKT'deki optimizasyon tasarımı temel olarak en düşük işletme maliyetinin belirlenmesini amaçlamaktadır. Bu çerçevede, bu çalışma sekiz ana bölümden oluşmaktadır ve aşağıda bu bölümler hakkında kısaca bilgiler verilmiştir.

- Birinci bölüm, tez konusu hakkında genel bir bilgi sunar.
- İkinci bölüm, araştırmanın hedeflerini ve kapsamını genel bir bakış açısıyla sunar ve detaylandırır.
- Üçüncü bölüm, literatürdeki PSO ve çeşitleri ile gerçekleştirilen çalışmaları sunar.
- Dördüncü bölüm, optimizasyon algoritmaları hakkında bilgi sunar.
- Beşinci bölüm, Parçacık Sürü Optimizasyonu ve çeşitleri hakkında bilgi sunar.

- Altıncı bölüm, Kaynaklı Kiriş Tasarımı problemine ve bu problemi çözmek için önerilen Kaotik Çoklu Sürü Parçacık Sürü Optimizasyonu (KÇSPSO) algoritmasının metodolojisine, tasarımına ve uygulamasına odaklanır.
- Yedinci bölüm, önerilen algoritmanın en uygun çözümlerini ve elde edilen sonuçları sunar.
- Sekizinci bölüm, araştırmanın sonucunu ortaya koyar ve gelecekteki çalışmalara ilişkin tavsiyeler ve öneriler sunar.



2. ARAŞTIRMANIN AMACI

Optimizasyon yöntemleri bir çok uygulamada, özellikle bilgisayar mühendisliği alanında oldukça önemli bir yere sahiptir. Evrimsel algoritmalar karmaşık optimizasyon problemlerini ele almak için geliştirilmiştir. PSO, Genetik Algoritma ve Ateş Böceği Algoritması gibi çeşitli evrimsel algoritmalar çoğunlukla çeşitli optimizasyon problemlerini çözmek için kullanılmaktadır. Yalnızca bir parametrenin optimizasyonunu gerektiren optimizasyon problemleri tek fonksiyonlu optimizasyon problemi, birden fazla değişkenin optimize edilmesini gerektiren problemler çok amaçlı optimizasyon problemi olarak ele alınır (Yang vd., 2013).

Bilim ve mühendislik alanında karşılaşılan problemlerin çoğu karmaşık ve doğrusal olmayan optimizasyon problemleridir, çünkü her zaman birkaç doğrusal ve / veya doğrusal olmayan kısıtlamadan oluşurlar. Bu nedenle, nesnel işlevi optimize ederken bu gibi kısıtlamaları çözebilecek optimizasyon algoritmaları geliştirmeye ihtiyaç vardır. KKT problemi temel olarak, yüklemeye yükü, kayma gerilmesi, eğilme gerilmesi ve son sapma üzerindeki kısıtlamalara bağlı olarak kiriş maliyetinin en aza indirilmesini amaçlar. Bununla birlikte, kiriş kalınlığı, kiriş genişliği, kaynak uzunluğu ve kaynak kalınlığı değişkenleri sürekli tasarım değişkenleridir. Bu değişkenler için en uygun değerlerinin elde edilmesi kısıtlı bir optimizasyon problemi olarak karşımıza çıkmaktadır (Mirjalili vd., 2014; Yang, 2010b).

KKT probleminin karmaşık yapısını araştırmak ve anlamak için yapılan bu tez çalışması, bu problemin çözümünde KÇSPSO yaklaşımının performansını değerlendirmeyi hedeflemektedir ve aşağıdaki soruları cevaplamayı amaçlamaktadır:

- 1) Kısıtlanmış mühendislik tasarım sorunları üzerindeki sezgisel yöntemlerin etkisini araştırmak,
- 2) Kaynaklı Kiriş Tasarımı problemini çözmek için KÇSPSO algoritmasını geliştirmek ve bu algoritmayı kullanarak KKT tasarım değişkenleri için en iyi sonuçları tespit etmek,
- 3) Önerilen algoritmayı test etmek, değerlendirmek ve sonuçları diğer optimizasyon algoritmaları ile karşılaştırmak.

3. LİTERATÜR TARAMASI

Son otuz yıl boyunca, karmaşık mühendislik problemlerini çözmek için birçok optimizasyon algoritması önerilmiştir. Doğa, bu tür algoritmalar için daima esin kaynağı olmuştur. Bu algoritmaların birçoğu canlı hayvanların akıllı davranışlarından esinlenerek yazılmıştır. PSO, makine çizelgesi problemi, dökümde proses parametre optimizasyonu, makine toleransı, eşzamanlı optimizasyon ve dişli çarkının optimum ağırlık tasarımı gibi mekanik alanda farklı çözümleri optimize etmede başarıyla kullanılmıştır (Savsani, 2014). Bu bölümde, literatürdeki çalışmalardan bazıları kronolojik bir sırada sunulmuştur.

Gao ve arkadaşları, işleme ve tasarım toleranslarını optimize etmek için PSO'yu kullanmışlardır. Bu işleme toleransı, fiili maliyetleri çok etkilediği için ürünün geliştirilmesinde çok kritik öneme sahiptir. Burada, PSO, piston silindir düzeneğini tasarlamak için kullanılmıştır. Parçacığın tüm kısıtlamaları yerine getirip getirmediğini kontrol etmek için kullanılan Partikül fizibilite faktörü olarak adlandırılan bir faktör ile PSO modifiye edilmiştir. Yazarların bu çalışmasına göre PSO'nun yakınsaması genetik algoritmaya göre daha hızlı olmuştur (Gao vd., 2006).

Spina, enjeksiyonla kalıplanmış kısmın enjeksiyon kalıbı örtüsünün optimize edilmesinde PSO ve Yapay Sinir Ağları yaklaşımını kullanmıştır (Spina, 2006). Yazar, bu yaklaşım ile parçanın hacimsel dolumunu ve büzülmesini azaltmayı hedeflemiştir. İşlenen parametreler, paketleme basıncı, kalıp sıcaklığı, son parça ağırlığı, ortalama farkı, erime sıcaklığı, ortalama hacimsel büzülme, akış hızı, kayma gerilmesi ve kütle sıcaklığıdır.

Anghinolfi ve arkadaşları, planlama ve üretimdeki gecikmenin toplam ağırlığını en aza indirmek için Ayırık PSO kullanmışlardır (Anghinolfi vd., 2007). Yazarlar, çalışmalarında elde ettiği sonuçları Karınca Kolonisi optimizasyonu çalışması (Lian vd., 2006) ile karşılaştırmıştır. Yazarların çalışmasına göre, Ayırık PSO'nun en iyi performansı Lian ve ark. tarafından sunulan çalışmadan daha başarılı olmuştur.

Shu, çinko alüminyum alaşımında PSO'yu yapay sinir ağları ile birlikte kullanarak

dökme işleme optimizasyon parametrelerini biraraya getirmiştir. Basınç, Basınç-zaman, dökme sıcaklığı ve kalıp sıcaklığı olarak seçilen dört parametre bulunmaktadır. Yazarın bu çalışmasına göre, geliştirilmiş PSO optimize edilmiş işlem parametrelerini verir ve öngörülen mekanik özellikler gerçek mekanik özellikler ile yakın bir uyum içindedir (Shu, 2007).

Guo ve ark. imalat planlama optimizasyonunu elde etmek için ve proses çizelgeleme ve planlamanın birleşimi üretim planlama optimizasyonunu sağlamak için PSO kullanmışlardır. Elde edilen sonuçlar, hesaplama verimliliği açısından PSO algoritmasının benzetimli tavlama ve genetik algoritmaya göre daha verimli olduğunu göstermiştir (Guo vd., 2009).

Ramanathan ve arkadaşları, Ni-Diamond kompozit kaplamanın sertliğini etkileyen parametre ve sertliği optimize etmede PSO metodunu kullanmışlardır. İşlenen parametreler, pH, sıcaklık ve akım yoğunluğudur. Yazarlar, 431,9470 optimizasyon değerinde “Vickers Sertliği” elde etmişlerdir (Ramanathan vd., 2009).

Sarayloo ve Tavakkoli-Moghaddam, hücresel imalatta dinamik hücre oluşturma kısıtında Çok Amaçlı Parçacık Sürü Optimizasyonu (ÇAPSO) kullanmışlardır. Bu çalışma hücreler arası ve iş yükünün standart sapmasının toplamını, hücre içi hareket katsayısını ve toplam maliyeti olmak üzere üç hedef içerir. ÇAPSO'nun sonuçları, Baskın Olmayan Sıralama Genetik Algoritması II ile rastgele bazı test vakaları üzerinde karşılaştırılmıştır. Bu karşılaştırma zaman ölçütüne, bir dizi baskın olmayan çözüme ve bir kalite ölçütüne ilişkin olarak ÇAPSO'nun daha üstün olduğunu göstermiştir (Sarayloo ve Tavakkoli-Moghaddam, 2010).

Venketesan, bir çalışmada sıkma kalıbının optimal tasarım şeklini belirlemek için PSO kullanmıştır. Amaç, sıkma oranının ve kalıp koni açısının optimizasyonu yoluyla sıkma kuvvetini azaltmaktır. Sıkma oranı ve kalıp konisi açısı için en uygun değerler Benzetilmiş Tavlama algoritması ve PSO kullanılarak belirlenmiştir. PSO ve Benzetilmiş Tavlama algoritmaları kullanılarak elde edilen minimum kalıp basıncı sırasıyla 118,77 Megapaskal ve 123,141 Megapaskal'dır. Bunlar PSO'nun çok amaçlı optimizasyon açısından Benzetilmiş Tavlama algoritmasına göre daha verimli

olduğunu göstermiştir (Venketesan, 2010).

Manjunath ve arkadaşları Bor/Epoksi kompozit sürücü yelpazesi ve tek parça E-Cam/Epoksi sıralamasının en iyileştirmesini PSO algoritması ile gerçekleştirmişlerdir. Yazarlar, çalışmalarında şaft uzunluğuna, şaft yoğunluğuna ve şaft uzunluğunun dış ve iç çapına bağlı olan kompozit sürücü ağırlığını azaltmak işlemlerini amaç fonksiyon olarak tanımlamışlardır. Kısıtlamalar; cevap, yığınlama sıralaması ve daha az sayıda vuruş kalınlığıdır. Elde edilen sonuçlar, Bor/Epoksi ve E- Cam/Epoksi için en uygun katmanların sırasıyla 15 ve 16 olduğunu göstermiştir. Dahası, yazarlar bileşik malzemelerden %50 ile %85 arasında önemli bir ağırlık tasarrufu elde etmişlerdir (Manjunath vd., 2011).

Noor ve arkadaşlarının lazer ışını kesim işleminde akrilik levhalarda optimum yüzey pürüzlülüğü elde etmeye çalışmışlardır. Yazarlar, malzeme kalınlığı 9 mm, güç %95, uç mesafesi 9,7 mm ve kesme hızı 2600 darbe/s olan işlem faktörlerini optimize etmek için PSO kullanmışlardır (Noor vd., 2011).

Srirat ve ark. boş bir levhada kırılma ve yırtılma gibi kusurları önlemek için ağır bir çizim prosedüründe en uygun taslak tutucu aralığını ve şeklini belirlemek için PSO kullanmışlardır. Bu çalışmada dikkate alınan parametreler ortalama kalınlıktan standart sapma, kalınlık sapması ve minimum ve maksimum kalınlık arasındaki fark olup istenilen taslak tutucu aralığı elde edilmiştir (Srirat vd., 2011).

Yarmohamadi çalışmasında, mobil hedeflere ve engellere sahip dinamik bir ortamda robot yolunun uzunluğunun azaltılması için PSO kullanmıştır. Yazar, çalışmasında verimli yakınsama sağladığı için PSO algoritması içine bir mutasyon operatörü kullanmıştır. Elde edilen sonuçlar, PSO tekniğinin gelişmiş ceza fonksiyon tekniğiyle karşılaştırıldığında daha verimli olduğunu göstermiştir (Yarmohamadi, 2011).

Chen ve ark. çok delikli sıkma işlemlerinin işlem faktörlerini optimize etmek için Taguchi tekniğiyle birleştirilmiş mutasyon kullanan bir PSO geliştirmişlerdir. Ayrıca, yazarlar eksantriklik oranı, sıkma hızı, kalıp ara yüzündeki sürtünme katsayısı ve kütük sıcaklığı gibi işlem faktörlerini optimize etmek için Taguchi tabanlı Çoklu Sürü PSO algoritmasını kullanmışlardır (Chen vd., 2013).

Liu ve arkadaşları, dikey elektrikli ısıtma basınçlandırma ağırlığını optimize etmek için modifiye bir Nelder-Mead simpleks algoritması ile PSO'nun hibritlenmesini içeren bir çalışma önermişlerdir. Toplam ağırlığın en aza indirilmesi için tek bir amaç fonksiyonunu içeren etkili bir yerel optimizasyon algoritmasının gerekli olduğu bu çalışmadaki tasarım parametreleri reaktör çıkış soğutma suyu sıcaklığı, primer döngü işletme basıncı, basınçlandırıcı iç çapı ve reaktör giriş soğutma suyu sıcaklığıdır. Bu çalışmada, basınçlandırıcı su hacmi, basınçlandırıcı yüksekliği, minimizasyon buhar hacmi ve basınçlandırıcı buhar hacmi kısıtlamalarını içeren bu hibrit model ile asgari 0,830779 basınç elde edilmiştir (Liu vd., 2013).

Wang ve Qiu, Çin'in Bohai Körfezi'nde bulunan bir petrol sahasında termal yağ geri kazanımının optimize edilmesine odaklanmışlardır. Yazarlar çalışmalarında, ağır bir yağ deposunun nihai yağ geri kazanımını optimize etmek için Canonical, Levy Bare-bone ve Gaussian Bare-bone PSO içeren üç çeşitli PSO algoritmasını test etmişlerdir. Elde edilen sonuçlar, 19,31 değerinde bir geri kazanım değeri ile geniş alan optimizasyonu zorlukları için Levy Bare-bone'nin verimli olduğunu göstermiştir (Wang ve Qiu, 2013).

Garg ve arkadaşları, sondaj sırasındaki kanadın aşınmasını öngörmek için gerekli olan önemli makine fonksiyonlarının analizinde PSO kullanmışlardır. Yapay sinir ağı eğitimi için PSO algoritmasını kullanmışlar ve önerdikleri modeli standart Geri Yayılımlı Sinir Ağı ile karşılaştırmışlardır. Elde edilen sonuçlar PSO'nun Geri Yayılımlı Sinir Ağından daha üstün bir performansa sahip olduğunu göstermiştir (Garg vd., 2014).

Tambolia ve arkadaşları, ağır işlerde uygulanan sarmal dişli çiftinin küçültülmüş bir hacime ulaşması için PSO ile optimize etmişlerdir. Çalışmanın odak noktası, sarmal açısı, modülü, dişli, pinyondaki diş sayısı ve yüz genişliğini içeren beş tasarım faktörüne göre hacim küçültmesini formüle etmektir. Bununla birlikte, kısıtlamalar diş kırılması ve çukurlaşmadan korunma parametresi, sarmal açısı kısıtlamaları, yüz genişliği kısıtlamaları ve dişlilerin çakışmaması için minimum diş sayısı olmak üzere dört tanedir (Tambolia vd., 2014).

Son olarak, bir başka çalışmada Bonyadi ve Michalewicz, PSO'yu bulanık sistem kontrolü, yapay sinir ağı eğitimi gibi farklı optimizasyon problemlerinde başarıyla kullanmışlardır (Bonyadi ve Michalewicz, 2017).



4. OPTİMİZASYON ALGORİTMALARI

Optimizasyon algoritmaları, hayvan gruplarının veya böceklerin akıllı davranışlarına dayanarak tasarlanmış meta-sezgisel algoritmalarlardır. Bu tez çalışmasında önerilen KÇSPSO algoritmasının performansını karşılaştırmak için kullanılan optimizasyon algoritmaları hakkında aşağıda kısaca bilgiler sunulmuştur.

4.1. Ateş Böceği Algoritması

Ateş Böceği Algoritması (ABA), tropikal ateş böceklerinin davranışlarına ve hafif yanıp sönen desenlerine dayanan bir optimizasyon algoritmasıdır. Kullanımı kolay, basit ve esnek bir algoritmadır. İdeal olarak, şu kurallara dayanarak çalışır (Yang, 2009; Yang, 2010c):

- i) Ateş böcekleri arasında cinsiyet değişikliği yoktur, ateş böcekleri cinsiyetlerine bakılmaksızın diğer ateş böcekleri tarafından cezbedilebilir.
- ii) Yayılan ışığın parlaklığı bir ateş böceğinin çekiciliğini belirler. Her iki parametre de ateş böcekleri arasındaki mesafe arttıkça düşer ve daha parlak ateş böceklerinin yokluğunda sürü rastgele hareket eder.
- iii) Bir ateş böceği tarafından yayılan ışığın parlaklığı, nesnel bir işlevdir

4.2. Guguk Kuşu Arama Algoritması

Guguk Kuşu Arama Algoritması (GKAA), guguk kuşlarının üreme davranışlarından esinlenilmiş yeni bir metasezgisel algoritmadır (Yang ve Deb, 2009). Guguk kuşları temelde yumurtalarını diğer kuşların yuvalarına bırakır. Bu yumurtalar, konakçı kuş tarafından tahrip edilir veya kuş kendi yumurtalarının olmadığını fark ettiğinde kuş yuva ile birlikte yumurtaları terk eder (Fister vd., 2013). Bu davranışı bir optimizasyon aracı olarak uygulamak için şu üç kural göz önünde bulundurulur:

- i. Her guguk kuşu rastgele bir yuvaya atılan yalnızca bir yumurtayı bırakabilir; bu yumurta bir çözüm kümesini temsil eder;
- ii. Gelecek nesillerde en iyi yumurtaları içeren yuvaların bir kısmı en iyi çözümler

temsil edilecektir;

- iii. Sabit sayıda yuva vardır ve yabancı bir yumurtayı keşif eden konakçı kuşun bulunma olasılığı vardır ve bu gerçekleşirse yumurta ya atılabilir ya da bütün yuva terkedilir. Böylece yeni bir konumda yeni bir yuva inşa edilir (Yang ve Deb, 2009).

4.3. Çiçek Tozlaşma Algoritması

Çiçek Tozlaşma Algoritması (ÇTA) yeni ve popüler bir populasyon tabanlı algoritma olup çiçekli bitkilerin tozlaşma davranışını simüle eder. Optimum bitkinin üreme stratejisi, en uygun olanın ve optimal üreme bitkilerinin sayı bakımından hayatta kalmasını içerir. Algoritma, pek çok sayısal ve mühendislik optimizasyon problemini çözmek için yaygın olarak uygulanmaktadır (Yang, 2012).

4.4. Yerçekimi Arama Algoritması

Sezgisel optimizasyon algoritmaları bir çok mühendislik probleminin çözümünde başarılı bir şekilde kullanılmaktadır. Bu algoritmalarından biri de Newton'un evrensel çekim kanunundan esinlenerek geliştirilmiş olan Yerçekimsel Arama Algoritması (YAA)'dır. Bu algoritma arama uzayında hem global hem yerel arama yaparak daha fazla yüzey taraması yapmaktadır (Kazak vd., 2015).

4.5. Çoklu Evren Optimizasyonu Algoritması

Çoklu Evren Optimizasyon (ÇEO) algoritması, optimizasyon ve mühendislik problemleri için önerilen bir algoritmadır. Kara delik, sıcak delik ve beyaz delik olmak üzere üç ana bileşen bu algoritmada matematiksel olarak modellenmiştir. Bu algoritma, "Birden fazla büyük patlama var ve biri bir evrenin doğuşuna neden olur" teorisini taklit eder ve farklı optimizasyon problemlerini çözmek için kullanılır (Mirjalili vd., 2016).

5. PARÇACIK SÜRÜ OPTİMİZASYONU VE ÇEŞİTLERİ

Optimizasyon problemlerine ve algoritmalarına genel bir bakış önceki bölümde sunulmuştur. Bu bölümde, PSO ayrıntılı olarak açıklanmaktadır. Ayrıca, bu bölüm PSO algoritmalarının çeşitlerini ve bu çerçevede mühendislik optimizasyonu problemlerinin çözümü için diğer araştırmacılar tarafından yapılan önemli çalışmaları gözden geçirmektedir.

PSO, Eberhart ve Kennedy tarafından 1995 yılında önerilen popülasyona bağlı stokastik optimizasyon tekniğidir. Balık sürüleri ve kuş sürüleri de dâhil olmak üzere mikro organizmaların sosyal davranışlarından ilham almış olan bu teknikte, hız; bir parçacığın konumundaki bir değişiklik olarak bilinir. Bu parçacıklar zamanla konumlarını değiştirirler. Uçuş sırasında parçacıkların hızı etrafına ve önceki en iyi yerel çözümüne doğru rastgele hızlandırılır (Eberhart ve Kennedy, 1995; Palupi Rini vd., 2011; Van Den ve Engelbrecht, 2002). PSO, kuş sürüsü davranışını simüle eder. Örneğin, bir kümes hayvanının stokastik olarak yiyecekleri aradığı bir lokasyonda keşfedilmesi gereken sadece bir yiyecek türü vardır. Bu yüzden yiyecek aramak için en etkili yöntem, yiyeceğe en yakın kuşları izlemektir. PSO, bu örnekten yola çıkarak optimizasyon zorluklarını çözmede kullanılır. PSO'daki kuş, arama alanındaki her bir çözümü temsil eder ve "Parçacık" olarak adlandırılır (Salih vd., 2019a).

PSO'da, potansiyel bir çözüm, çok boyutlu aranan boşluklardan akan bir parçacık olarak tanımlanmaktadır. Bir parçacığın arama alanındaki konumlarındaki herhangi bir değişiklik bir bireyin sosyal psikolojisinin başka bir bireyin zaferini taklit etme potansiyeline bağlı olacaktır. Parçanın sürü içindeki pozisyonlarındaki herhangi bir değişiklik komşularının bilgi ve tecrübelerinden de etkilenebilir. Bu kümeler olası bir çözümü işaret eden bir grup parçacıktan oluşur. Bir parçacık, komşularına ve kendi deneyimlerine dayanarak her parçanın konumunu değiştirmek için hiper uzayda uçar (Shi ve Eberhart, 1998; Tanweer vd., 2015; Van Den, 2006).

Her parçacık, uygunluk optimizasyonu rolü ile incelenen uygunluk değerine ve parçacığın uçmasını yönlendiren hıza sahiptir. Bir parçacık, en son en iyi parçacıkları takip eder. PSO, rastgele ayarlanan çözümler ile başlatılır ve daha sonra en iyi

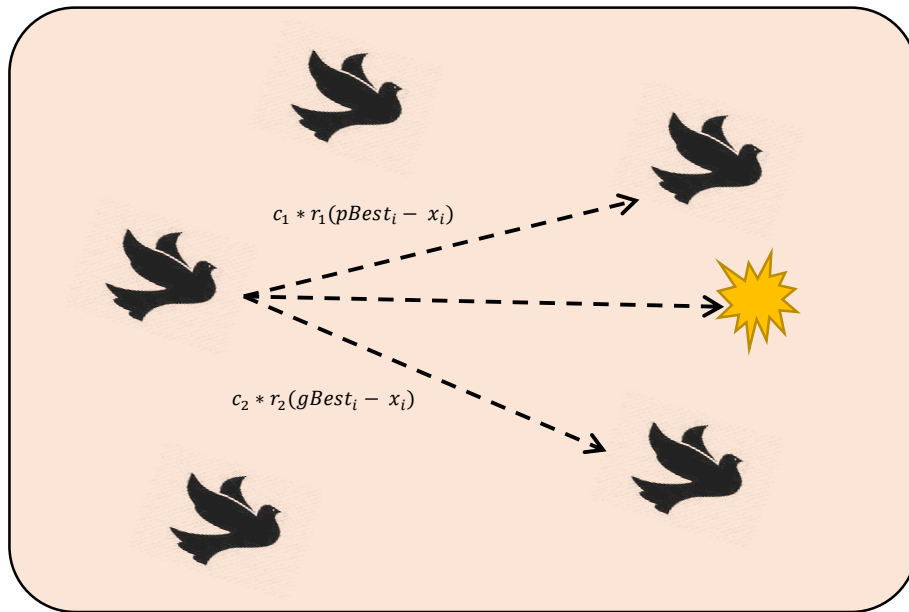
yükseltme nesnesini arar. Herhangi bir yinelemede parçacıklar bu iki "optimal" değer ile yükseltilir. En iyi uygunluk o ana kadar elde edilen ve **Pbest** adı verilen uygunluk değerleri korunur. Parçacık sürü optimizasyonunun izlediği diğer "en iyi" değerler popülasyon içindeki herhangi bir parçacıkta türetilmiş en iyi değerlerdir. Bir parçacık topolojik komşularını popülasyonları olarak tuttuğunda bu en iyi değerler en iyisi olarak bilinen yerel yerlerdir. En iyi iki değeri aradıktan sonra bu parçacıklar (5.1) ve (5.2) denklemlerini izleyerek konumlarını ve hızlarını günceller (Eberhart ve Kennedy, 1995; Poli vd., 2007):

$$v_i(t + 1) = v_i(t) + r_1 c_1 (P_i - x_i(t)) + r_2 c_2 (P_g - x_i(t)) \quad (5.1)$$

$$x_i(t + 1) = x_i(t) + v_i(t) \quad (5.2)$$

$$w = w_{max} - \frac{w_{max} - w_{min}}{itr_{max}} \times itr \quad (5.3)$$

Burada, $v(k + 1)$ yeni parçacık hızıdır, $x_i(k)$ mevcut parçacıktır (çözüm), $pbest$ ve $gbest$, $rand_1$ and $rand_2$ olarak tanımlanan (0, 1) arasındaki rasgele sayılardır. c_1 ve c_2 öğrenme faktörleridir. Şekil 5.1, PSO algoritmasının temel davranışını göstermektedir.



Şekil 5.1. PSO'nun bir tasviri

5.1. Hibrit Parçacık Sürü Optimizasyonu

Hibrit Parçacık Sürü Optimizasyonu, global optimum noktaya en az iterasyonda en kısa zamanda yakınsamasını sağlamak için geliştirilen algoritmadır. Bu algoritmada temel hedef, popülasyondaki farklılıkları aşağıdaki maddeler çerçevesinde geliştirmektir (Palupi Rini vd., 2011):

- 1) Atalet ağırlığı, hızlanma sabiti ve daralma faktörlerini içeren kendinden uyarlamalı parametrelerin kullanılması,
- 2) Parçacıkların birbirine çok yakın hareket etmesini engelleyerek parçacıkların çarpışmasının önlenmesi.

5.2. Uyarlanır Parçacık Sürü Optimizasyonu

Uyarlanır PSO yaklaşımında aşağıdaki adımlar çerçevesinde PSO algoritmasındaki parametrelere ayarlamalar yapılır (Janson ve Middendorf, 2005).

- a) Atalet ağırlıklarına bazı randomize bileşenler eklenmesi,
- b) Mantıksal bulanıklığın kullanılması,
- c) En iyi parametrenin elde edilmesi,
- d) Q-öğrenme ile iyileştirme yapılması.

Buradaki, Q-öğrenme pekiştirmeli öğrenmenin en çok bilinen algoritmalarından biridir. Bu algoritmadaki temel amaç bir sonraki hareketleri inceleyip yapacağı hareketlere göre kazanacağı ödülü görmek ve bu ödülü maksimize ederek buna göre hareket etmektir (Watkins, 1989).

5.3. Çok Amaçlı Parçacık Sürü Optimizasyonu

Klasik PSO, çeşitli objektif zorlukların çözümünde, ağırlık ve dinamik değişimleri göz önünde bulundurarak optimizasyon işlemleri sırasında tek bir amaç için kullanılmaktadır.

Çok amaçlı zorlukları çözmeye etkili olduğu görülen ÇAPSO' da dikkat edilmesi gereken en önemli koşul en iyi optimale ulaşmak ve bireysel değişimleri sürdürmek

için sosyal liderlerin ve bilişsellersin (*pbest* ve *lbest*) seçilmesidir (Janson ve Middendorf, 2005).

5.4. Ayrık Parçacık Sürü Optimizasyonu

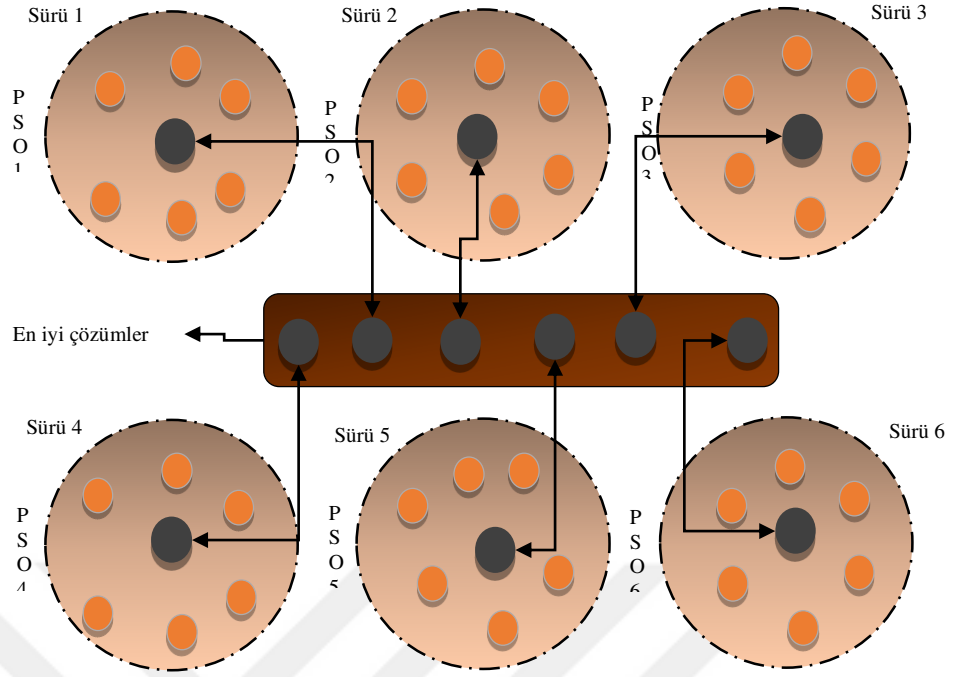
Bu yaklaşımda, algoritmaların performanslarını çeşitli fonksiyonlara dönüştürmek için PSO'nun ikili bir versiyonunda yapılan modifikasyonlar ele alınır. Bu algorithmada 3 strateji önerilmektedir:

- 1) Doğrudan yaklaşım: Bu strateji için klasik PSO algoritmaları kullanılır ve güçlü kod çözme süreçleri kullanılarak çözüm bit dizisinde değiştirilir.
- 2) Eğilim vektörü yaklaşımı: Bu yöntem için rastgele bir şekilde uygunluk fonksiyonlarının değerlerine dayalı olasılık kullanılarak hız belirlenir.
- 3) Karma arama yaklaşımı: Bu tür bir yaklaşım için her tanecik dinamik olarak PSO'nun genel veya yerel sürümlerine uyarlandığı çoklu kümelere bölünebilir.

5.5. Çoklu Sürü Parçacık Sürü Optimizasyonu

Çoklu Sürü Parçacık Sürü Optimizasyonu (ÇSPSO), aynı anda tek bir sürü yerine en iyi çözüm için arama alanında arama yapar. Daha sonra her bir kümenin en iyisi algoritmanın genel arama yeteneğini arttırmak için bir toplantı odasına gönderilir. İlk olarak odadaki en iyi çözüm sunan en iyi lider olarak seçilirken diğerleri normal liderleri temsil eder. Her lider için pozisyonlar ve kontrol parametreleri en iyi lider temelinde güncellenir. Ardından, sürülerine geri dönen lider en iyi çözümü bulmak için yeni bir yineleme başlatırlar. ÇSPSO çeşitli normal ve büyük ölçekli problemlere dayanarak değerlendirilmiş ve bu yaklaşımın standart PSO algoritmasına göre daha iyi olduğu kanıtlanmıştır (Salih vd., 2019a).

Çoklu kümelenmenin temel fikri çözüm ararken birkaç grup arasındaki etkileşimdir. ÇSPSO şeması birkaç kümeden oluşur ve her bir küme grup üyelerini temsil eden birkaç çözümden oluşur. En iyi üye bir sürünün lideri olarak belirlenir. Lider, sürünün üyelerini zaman veya nereye gidileceği açısından kontrol eder (Salih vd., 2018; Salih vd., 2019b).



Şekil 5.2. PSO tabanlı TO yaklaşımı

Her nesilde liderler genel olarak en iyi liderin kendi konum bilgisine dayanarak diğer normal liderlerin pozisyonlarını güncellediği bir odada buluşur. Bu bilgi paylaşımı davranışı keşif aşamasını PSO'nun keşif aşaması ile dengelemeye yardımcı olur. Bu çok alanlı yaklaşım TO yaklaşımı olarak adlandırılır. PSO tabanlı TO modeli Şekil 5.2'de gösterilmiştir. Bu şekilde her bir küme konumsal ve hız güncellemesinin yanı sıra yeni yerel popülasyon nesnesini içeren tek bir PSO araması gerçekleştirir. Tüm kümeler için yeni nesiller kuran her küme liderini toplantı odasına "en iyi çözüm" olarak gönderir. Toplantı odasındaki tüm liderler arasında en iyisi genel olarak en iyi lider olarak seçilir. Yeni en iyi lider konum bilgisini (5.4), (5.5) ve (5.6) eşitliklerini kullanarak diğer liderlerle paylaşır (Salih ve Alsewari, 2018):

$$w^{Ln} = \left(\frac{w^{Lg} - w^{Ln}}{itr} \right) \times rand() \quad (5.4)$$

$$v_i^{Ln}(t+1) = w^{Ln} \times v_i^{Ln}(t) + rc \left(P_g^L - P_n^L(t) \right) \quad (5.5)$$

$$x_i^{Ln}(t+1) = x_i^{Ln}(t) + v_i^{Ln}(t) \quad (5.6)$$

Burada, Ln normal liderleri temsil eder, Lg genel olarak en iyi lideri temsil eder, x_i^L

normal liderin konumunu temsil eder, v_i^{Ln} normal liderin hızını gösterir, w^{Lg} ve w^{Ln} sırasıyla en iyi liderin ve normal liderin atalet ağırlığını temsil eder. Her jenerasyondan sonra üyelerin pozisyonlarındaki değişikliklerden dolayı her bir sürü için yeni bir lider seçilir. ÇSPSO, TO yaklaşımın ana fikrini kullanır ve onu PSO'nun davranışına uygular.



6. KAYNAKLI KİRİŞ TASARIMI PROBLEMİ

Mühendislik tasarım problemlerini simüle etmek ve çözmek için çeşitli optimizasyon algoritmaları önerilmiştir. Bu algoritmaların çoğu, kuşlar, karıncalar, arılar ve hatta yarasalar gibi bazı sürülerin davranışları gibi doğadan ilham almıştır. Bu tezde amaç, KKT problemini çözmek için kaotik harita yönteminden yararlanan KÇSPSO tabanlı yeni bir çözüm sunmaktır. Bu çerçevede bu bölümde sırasıyla aşağıdaki öğelere yer verilmiştir.

- KKT hakkında bilgi
- KKT üzerine yapılmış çalışmalar
- KKT probleminin matematiksel formüllere dayalı olarak açıklanması
- KÇSPSO hakkında bilgi
- KKT problemi için önerilen yaklaşım

6.1. Kaynaklı Kiriş Tasarımı

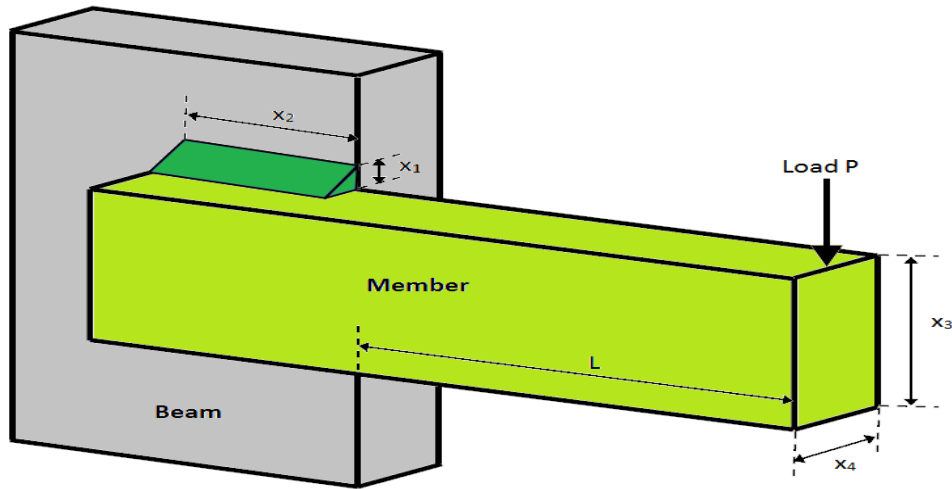
Kaynak, metalik parçaların basınç uygulanarak veya uygulanmadan uygun bir sıcaklığa ısıtılmasıyla birleştirilmesi işlemi olarak tanımlanabilir. Kaynak, metal parçaların kalıcı bir birleşimini elde etmenin ekonomik ve verimli bir yöntemidir. Kaynak işlemi genel olarak aşağıdaki iki gruba ayrılır:

1. İki parçayı birleştirmek için yalnız ısı kullanan kaynak işlemi.
2. İki parçayı birleştirmek için ısı ve basınç kombinasyonunu kullanan kaynak işlemi.

Kaynaklı bağlantıların iki farklı uygulaması vardır:

1. Kaynak yapılmış bir bağlantı perçinlenmiş bağlantı yerine kullanılabilir.
2. Kaynaklanmış bir yapı döküm veya dövme için alternatif bir yöntem olarak kullanılabilir.

Sadece ısı kullanan kaynak işlemine füzyon kaynak işlemi denir. Bu yöntemde birleştirilecek parçalar yerinde tutulur ve erimiş metal bağlantıya verilir. Erimiş metal Parent metal olarak adlandırılan parçalardan veya ek yerine verilen dış dolgu metalinden gelebilir. İki parçanın birleştirme yüzeyi plastik hale gelir veya ısı etkisi altında eritilir. Eklem sertleştğinde iki parça tek bir üniteye birleşir. Bir kiriş, uzunlamasına, enlemesine uygulanan ve üyenin bükülmesine neden olan yüke maruz kalmış bir elemandır. Kirişler sıklıkla desteklere veya reaksiyonlara göre sınıflandırılır. Uçlardaki pimler, makaralar veya pürüzsüz yüzeyler tarafından desteklenen kirişe basit kiriş denir. Bir kiriş projesinin iki ucu veya her ikisi de destekçilerin ötesindeyse çıkıntı yapan basit bir kiriş adı verilir. Bir kiriş sert bir elemana kaynaklanır. Kaynaklanmış kiriş çelikten oluşur ve 6000 pound-kuvveti destekler. Kirişin boyutları sistem maliyeti minimize edilecek şekilde seçilmelidir. (Mirjalili vd., 2014; Mirjalili vd., 2016; Nematollahi vd., 2017; Yang, 2010b). KKT'nin şeması Şekil 6.1'de gösterilmektedir



Şekil 6.1. Kaynaklı Kiriş Tasarımı

6.2. Kaynaklı Kiriş Tasarımı Problemi Çalışmaları

Bu problemin çözümü için yapılmış çalışmalardan bazıları şu şekildedir: Coello, 2000 yılında, evrimsel algoritmalar kullanarak kısıtlamaları ele almış ve belirtilen KKT problemini çözmek için çok hedefli bir optimizasyon kullanmıştır (Coello, 2000).

Ray ve Liew, KKT problemini çözmek için sosyal davranışa dayalı yeni bir optimizasyon algoritması önermişlerdir (Ray ve Liew, 2003).

He ve arkadaşları, optimizasyon problemlerinin özellikle KKT problemi ve diğer makine mühendisliği problemlerinin kısıtlarını ele almak için iyileştirilmiş bir PSO sürümü önermişlerdir. Önerilen PSO sürümü, karışık değişkenli doğrusal olmayan optimizasyon problemleri üzerinde doğrulanmış ve standart PSO'ya kıyasla daha etkin bir performans göstermiştir (He vd., 2004).

Mezura-Montes ve Coello, KKT gibi optimizasyon problemlerini çözmek için farklı evrim stratejileri türleri arasında bir karşılaştırma çalışması sunmuşlardır (Mezura-Montes ve Coello, 2008).

Savsani, KKT problemi için Yapay Arılar Kolonisi algoritmasının yakınsama hızı iyileştirilmiş bir versiyonunu önermiştir (Savsani, 2014).

Mirjalili ve arkadaşları KKT gibi mühendislik problemlerini çözmek için gri kurtlardan esinlenen Gri Kurt algoritması adında yeni bir meta-sezgisel bir algoritma önermiştir. Bu algoritma, liderlik hiyerarşisini ve gri kurdun avlanma mekanizmasını taklit eder (Mirjalili vd., 2016).

6.3. Kaynaklık Kiriş Tasarımı İçin Matematiksel Formülasyon

Şekil 6.1 bir kirişe kaynaklanmış sert bir elemanı göstermektedir. Üyenin sonuna bir yük uygulanır. Kiriş, kaynak ve eleman boyutlarını $X = (x_1, x_2, x_3, x_4)$ değiştirerek minimum maliyet için optimize edilecektir. Bu parametreler, kayma gerilmesinin sınırlarını, eğilme gerilmesini, burkulma yükünü ve uç sapmasını içerir. Burada amaç, kaynaklı kiriş tasarım probleminin maliyetini en aza indirmektir. Bu çerçevede, temel olarak kaynak işçiliği ve malzeme maliyetini içeren toplam imalat maliyeti olan $f(x)$ amaç fonksiyonunun matematiksel formülasyonu (6.1) eşitliğindeki gibi olup kısıtlar (6.2) ve (6.8) arasında verilmiştir.

$$\text{Min } f(x) = 1,10471x_1^2x_2 + 0,04811x_3x_4(14,0 + x_2) \quad (6.1)$$

$$g_1(x) = \tau(x) - \tau_{max} \leq 0 \quad (6.2)$$

$$g_2(x) = \sigma(x) - \sigma_{max} \leq 0 \quad (6.3)$$

$$g_3(x) = x_1 - x_4 \leq 0 \quad (6.4)$$

$$g_4(x) = 1,10471x_1^2 + 0,04811x_3x_4(14,0 + x_2) - 5,0 \leq 0 \quad (6.5)$$

$$g_5(x) = 0,125 - x_1 \leq 0 \quad (6.6)$$

$$g_6(x) = \delta(x) - \delta_{max} \leq 0 \quad (6.7)$$

$$g_7(x) = P - P_c(x) \leq 0 \quad (6.8)$$

Bu kısıtlar için gerekli parametreler (6.9) ve (6.18) eşitlikleri arasında verilmiştir.

$$\tau(x) = \sqrt{(\tau')^2 + 2\tau'\tau'' \frac{x_2}{2R} + (\tau'')^2} \quad (6.9)$$

$$\tau' = \frac{P}{\sqrt{2} x_1 x_2} \quad (6.10)$$

$$\tau'' = \frac{MR}{J} \quad (6.11)$$

$$M = P(L + \frac{x_2}{2}) \quad (6.12)$$

$$R = \sqrt{\frac{x_2^2}{4} + (\frac{x_1 + x_3}{2})^2} \quad (6.13)$$

$$J = 2 (\sqrt{2} x_1 x_2 \left[\frac{x_2^2}{12} + \left(\frac{x_1 + x_3}{2} \right)^2 \right]) \quad (6.14)$$

$$\sigma(x) = \frac{6PL}{x_4 x_3^2} \quad (6.15)$$

$$\delta(x) = \frac{4PL^3}{Ex_3^3 x_4} \quad (6.16)$$

$$P_c(x) = \frac{4.013 \sqrt{(EGx_3^2 x_4^6)/36}}{L^2} \left(1 - \frac{x_3}{2L} \sqrt{\frac{E}{4G}} \right) \quad (6.17)$$

$$P(x) = 6000x_2 x_4 \quad (6.18)$$

Burada, $L=14$ inç, $E = 30 \times 10^6$ psi, $G = 12 \times 10^6$ psi, $\delta_{max} = 0.25$ inç, $\tau_{max} = 13600$ pounds per square inch (psi), $\sigma_{max} = 30000$ psi'dir. 13600 katsayısı tasarımın kesme gerilmesi $\tau(\max)$ değerini, 30000 katsayısı tasarımın kiriş malzemesinin normal stres $\sigma(\max)$ değerini, 0,25 katsayısı Maksimum sapma $\delta(\max)$ değerini göstermektedir. Değişkenlerin birinci sınır aralığı ise $0.1 \leq x_1 \leq 2$, ikinci ve

üçüncü sınır aralığı ise $0.1 \leq x_2, x_3 \leq 10$, dördüncü değişken aralığı ise $0.1 \leq x_4 \leq 2$ ’ dir. İlk kısıtlama g1, geliştirilen maksimum kesme gerilmesinin kaynak malzemesinin emniyetli kesme gerilmesinden daha az olmasını sağlar. İkinci kısıtlama g2, geliştirilen maksimum normal gerilmenin kirişte emniyetli normal gerilmeden daha düşük olduğunu kontrol eder. Üçüncü kısıtlama g3, kiriş kalınlığının kaynağından daha yüksek olmasını sağlar. Dördüncü ve beşinci kısıtlamalar g4 ve g5, negatif uzunlukları veya kalınlıkları önlemek için pratik kontrollerdir. Altıncı kısıtlama g6, kiriş üzerindeki yükün emniyetli burkma yükünden daha büyük olmamasını sağlar. Yedinci kısıtlama g7, kaynak kalınlığının belirli bir minimumun üstünde olduğunu kontrol eder (Rekliatis vd., 1983). Burada, M, kirişin sapmasını, P Kiriş üzerindeki kuvvet, J kirişteki bükülme gerilimini, Pc çubuk üzerindeki burkulma yükünü temsil etmektedir (Smarandache vd., 2017).

6.4. Kaotik Çoklu Sürü Parçacık Sürü Optimizasyonu

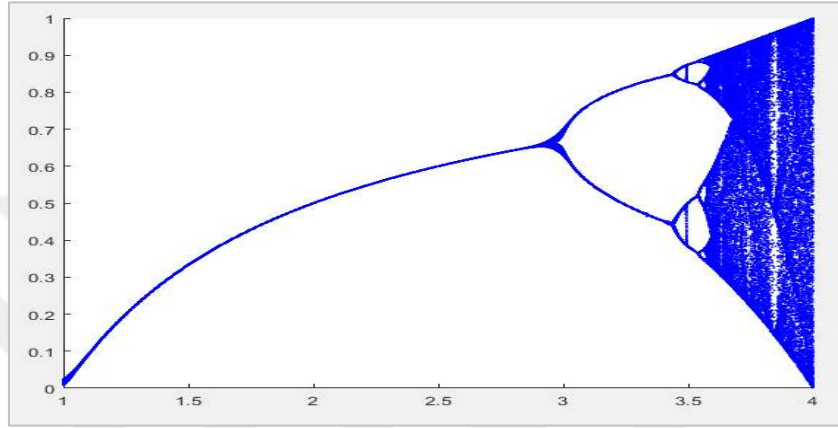
ÇSPSO, her ne kadar normal ve büyük ölçekli optimizasyon problemlerini çözmek için test edilmiş olsa da her bir kümedeki parçacıkların başlangıç aşamasında kullanılan homojen dağılım denklemi nedeniyle yerel optimumda sıkışma ihtimali vardır. İlk aşamada her bir kümedeki her bir parçacık eşit dağılım denklemi kullanılarak başlatılır. Bu nedenle, parçacıklar arama alanında arama alanını azaltan farklı bir başlangıç konumuna sahip olmayabilir.

Bu tezin ana katkısı, kaotik lojistik haritası ile oluşturulan bir pozisyon kullanarak ÇSPSO algoritmasının başlangıç aşamasını daha iyi bir noktaya getirmektir. Lojistik harita, biyolojik popülasyonların sözde rastgele sayı üreticinin ve şifreleme uygulamalarının kullanımını doğrulamak için belirli bir matematiksel formül ile temsil edilen tek bir bileşen üzerine inşa edilmiş iyi bilinen bir kaotik haritadır. Ayrıca, lojistik haritanın sağladığı matematiksel basitlik onu kaos teorisindeki yeni kavramların yanı sıra veri güvenliğinde sınamak için önemli bir yöntem haline getirmiştir. Basitleştirilmiş matematiksel formül Eşitlik 6.19’da görülmektedir (Gandomi vd., 2013a; Gandomi ve Yang, 2014):

$$X_{i+1} = \mu X_i (1 - X_i), \quad (6.19)$$

Burada, $X_i \in (0,1)$, $X_0 \in (0,1)$ başlangıç durumunu koruyarak farklı durumu belirtirken, $\mu \in (3,999 - 4)$ kontrol parametresini temsil eder ve $i \geq 0$ yineleme sayısıdır (Chai vd., 2017; Pareek vd., 2006).

Şekil 6.2’de yatay eksen μ parametresinin değerlerini ve dikey eksen, X_n 'in olası uzun vadeli değerlerini ifade eder (Chai vd., 2017; Pareek vd., 2006).



Şekil 6.2. Lojistik haritanın çatallanması (iki kola ayrılması)

Lyapunov karakteristik üssü bir lojistik haritadaki kaotik yörüngeyi doğrulamakta önemli olan pozitif bir değer elde ettikten sonra kaos ve yörüngesel sapmayı tanımlayan sayısal bir yöntemdir. $x_{n+1} = f(x_n)$ için Lyapunov üssü, λ (6.20) eşitliğindeki gibi tanımlanır (Murillo-Escobar vd., 2017):

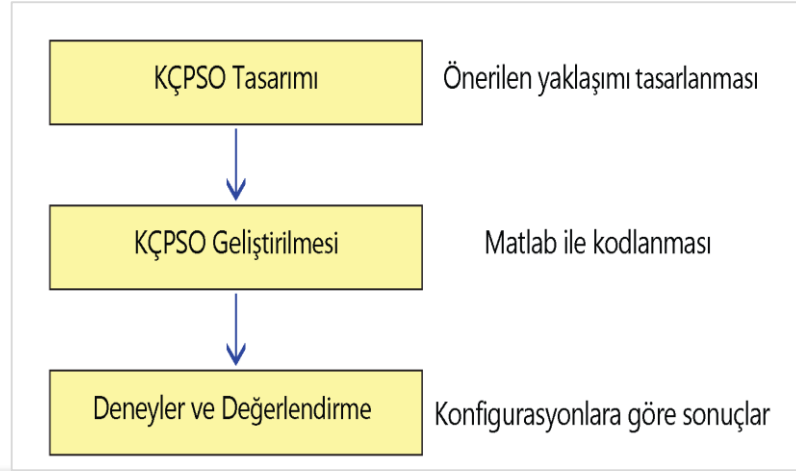
$$\lambda = \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=0}^{n-1} \ln |f'(x_n)| \quad (6.20)$$

Kaotik Lojistik Harita, modern bilimde yeni bir kavramdır ve araştırmacılar için operasyonel basitliği, yüksek hızı, başlangıç koşullarına, kontrol parametrelerine, kararlılık ve döngelik parametrelerine ve güvenlik sistemlerinin tasarımında gerekli olan tüm özelliklere duyarlı olan ideal bir tercihtir (Gandomi vd., 2013b).

6.5. Kaynaklık Kiriş Tasarımı Problemi İçin Önerilen Yaklaşım

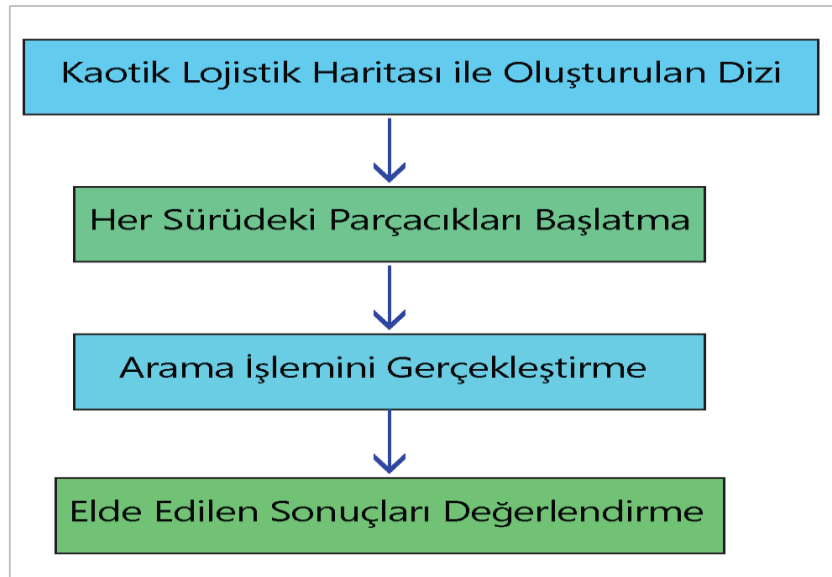
Bu tez çalışmasında önerilen yaklaşım; Kaotik ÇSPSO tasarımını, uygulamasını, deneylerini ve değerlendirmesini içeren temel aşamaları içermektedir. Metodolojinin

akış şeması Şekil 6.3'te gösterilmiştir.



Şekil 6.3. Araştırma metodolojisinin akış şeması

Önerilen algoritmanın genel çerçevesi Şekil 6.4'te verilmiştir. Buna göre, birinci aşamada, kaotik lojistik harita kullanılarak bir ilk dizi üretilir. İkinci aşamada, tüm parçacıklar üretilen kaotik diziye göre başlatılır. Daha sonra, üçüncü aşamada parçacıklar daha iyi çözümler bulmak için arama işlemine başlar. Son bileşen olan dördüncü aşama, her bir parçacığın nesnel işlev açısından değerlendirilmesi ile ilgilidir. KKT problemi için önerilen KÇSPSO çalışmasını gerçeklemek için Matlab tümleşik geliştirme ortamı kullanılmıştır.



Şekil 6.4. KÇSPSO'nun Çerçevesi

Bu çerçevede, önerilen algoritmanın ana adımları aşağıda açıklanmaya çalışılmıştır.

- Adım 1: İlk (x_i) ve kontrol (μ) parametreleriyle Kaotik Lojistik Harita yöntemi tanımlanır. Burada, $x \in [0, 1]$ ve $\mu \in [0, 4]$ olacaktır.
- Adım 2: Önceki adımda tanımlanan lojistik haritaya dayanarak iki kaotik dizi oluşturulur.
- Adım 3: Amaç fonksiyonunu tanımlanır. Bu çalışmada amaç işlevi temel kısıtlamalar dâhil olmak üzere KKT probleminin matematiksel formülasyonudur.
- Adım 4: Her parçacığın pozisyonları ilk üretilen kaotik diziyi temel alarak her bir sürüde sıfırlanır. Buna karşın hızlar, üretilen ikinci sıraya göre başlatılır.
- Adım 5: Her bir parçacık amaç fonksiyonuna göre değerlendirilir.
- Adım 6: Her bir sürüdeki en iyi partikülü, bu sürü için genel en iyi çözüm ($gbest$) ve yerel en iyi çözüm ($lbest$) olarak ayarlanılır.
- Adım 7: Her bir sürüdeki diğer normal parçacıkları en iyi çözümlerine ($gbest$) ve ($lbest$) doğru kaydırılır. Her parçacık için hız, (6.21) eşitliği ile güncellenir:

$$v_i(t + 1) = v_i(t) + r_1 c_1 (lbest_i - x_i(t)) + r_2 c_2 (gbest - x_i(t)) \quad (6.21)$$

Burada, v_i tek bir parçacık için hızı temsil eder, r_1 ve r_2 , $[0, 1]$ aralığında iki rastgele değerdir. c_1 ve c_2 , genel ve yerel en iyi çözümler için iki kontrol parametresidir ve değerleri $[1, 2]$ aralığındadır. Son olarak x_i pozisyonu temsil eder. Her parçanın pozisyonu (6.22) eşitliği ile güncellenir:

$$x_i(t + 1) = x_i(t) + v_i(t) \quad (6.22)$$

- Adım 8: Her bir kümedeki her bir parçacık yeni bir pozisyon aldıktan sonra parçacık amaç fonksiyon ile değerlendirilir.
- Adım 9: Yeni uygunluk bir öncekinden daha iyi ise $lbest$ yenisiyle güncellenir, aksi takdirde $lbest$ 'in önceki değeri korunur.
- Adım 9: Her parçacık için yeni $gbest$ ayarlanılır.
- Adım 10: Her bir sürü içindeki her $gbest$ toplantı odasını temsil eden genel sürüye gönderilir. Ardından en iyiyi $gbest$ lider olarak ayarlanılır.
- Adım 11: (6.23) ve (6.24) ile genel lidere dayalı olarak diğer normal liderlerin

hızı ve konumu güncellenir:

$$v_i^{Ln}(t+1) = w^{Ln} \times v_i^{Ln}(t) + rc \left(P_g^L - P_n^L(t) \right) \quad (6.23)$$

$$x_i^{Ln}(t+1) = x_i^{Ln}(t) + v_i^{Ln}(t) \quad (6.24)$$

- Adım 12: Her lider sürülerine geri konulur ve kaotik lojistik haritası ile parçacıklar yeniden başlatılır.
- Adım 13: Belirli bir yinelemenin ardından oluşan durma koşulu sağlandığında algoritma durur.

7. ELDE EDİLEN SONUÇLAR VE TARTIŞMA

Bu bölümde, KKT probleminin en uygun tasarım parametrelerini bulmak için önerilen algoritma iki deney türüne dayanarak değerlendirilmiş ve tartışılmıştır.

- a) Çeşitli optimizasyon problemleri üzerinde gerçekleştirilen testler.
- b) KKT probleminin çözümü üzerinde farklı sürü boyutları ve yineleme sayıları ile gerçekleştirilen testler.

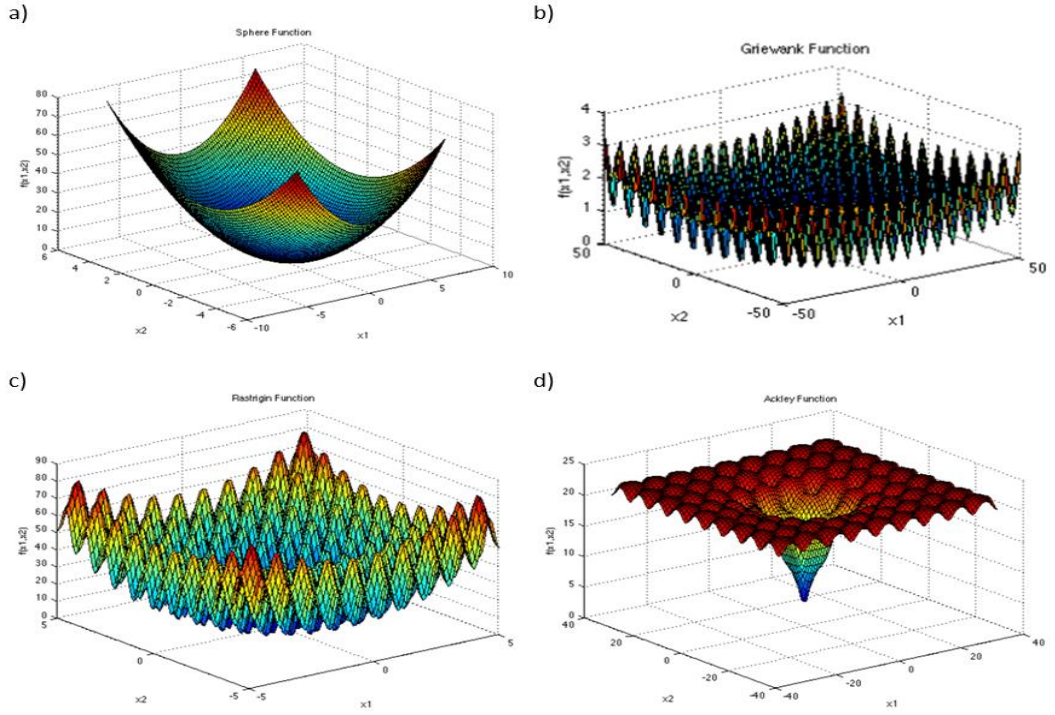
KKT probleminin çözümü için önerilen KÇSPSO yaklaşımı ve diğer algoritmalar ile yürütülen deneysel çalışmalar 64 bit mimariye sahip Windows 7 işletim sistemi ve Intel 2,4 GHz işlemci ve 8GB RAM donanım özelliklerine sahip bir bilgisayarda Matlab 2016b kullanılarak gerçekleştirilmiştir. KÇSPSO algoritmasını uygulamak için gerekli olan parametreler ve değerleri Tablo 7.1’de sunulmuştur.

Tablo 7.1. Önerilen algorithmada kullanılan parametreler ve değerleri.

N	Parametre	Sembol	Değer
1	Lojistik Harita: Başlangıç değeri	x_0	Rastgele [0;1]
2	Lojistik Harita: Değişim	μ	3,99999854
3	KÇSPSO: Atalet Ağırlığı	w	0,74
4	KÇSPSO: Bilişsel Ağırlık	c_1	1,4
5	KÇSPSO: Sosyal Ağırlık	c_2	1,4

7.1. Optimizasyon Problemleri için Kaotik Çoklu Sürü Parçacık Sürü Optimizasyonu

Bu bölümde KÇSPSO'nun performansı Şekil 7.1’de sunulan Sphere (f_1), Griewank (f_2), Rastirign (f_3) ve Ackley (f_4) optimizasyon problemleri (URL-1) üzerinde ele alınmış ve değerlendirilmiştir.



Şekil 7.1. Test fonksiyonları; a) Sphere b) Griewank c) Rastirign d) Ackley

Tablo 7.2’de bilgileri verilen bu optimizasyon problemleri birçok ÇSPSO araştırmasında kullanılmıştır (Salih ve Alsewari, 2018). Bu tabloda, sırasıyla fonksiyon bilgisi, problemin matematiksel formülasyonu, üst ve alt sınır (U_b, L_b) bilgileri bulunmaktadır.

Tablo 7.2. Değerlendirme için kullanılan optimizasyon test fonksiyonları (URL-1)

f_n	Fonksiyon	U_b, L_b
f_1	$f_1 = \sum_{i=1}^d x_i^2$	-5,12; 5,12
f_2	$f_2 = \sum_{i=1}^d \frac{x_i^2}{4000} - \prod_{i=1}^d \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	-600; 600
f_3	$f_3 = 10d + \sum_{i=1}^d [x_i^2 - 10 \cos(2\pi x_i)]$	-5,12; 5,12
f_4	$f_4 = -a \exp\left(-b \sqrt{\frac{1}{d} \sum_{i=1}^d x_i^2}\right) - \exp\left(\frac{1}{d} \sum_{i=1}^d \cos(cx_i)\right) + a + \exp(1)$ (Önerilen a, b ve c değerleri sırasıyla 20, 0.2 ve 2π ’ dir.)	-32,768; 32,768

Önerilen Kaotik ÇSPSO’nun verimliliğini araştırmak için her fonksiyon için farklı boyutlarda popülasyon büyüklükleri kullanılmıştır. Ayrıca, Kaotik ÇSPSO’nun

performansı PSO, Efendi-Köle PSO ve ÇSPSO ile karşılaştırılmıştır. Şekil 7.1’de sunulan 4 karşılaştırma fonksiyonunun 30 deney koşuturulması sonrasında parçacıkların en iyi ve ortalama uygunluk değerleri Tablo 7.3’te verilmiştir. Bu tabloda görüldüğü üzere Kaotik ÇSPSO neredeyse tüm durumlarda diğer algoritmalarından daha iyi sonuç sunmuştur. Bu deneylerde, Kaotik ÇSPSO 5 sürüye sahiptir; her bir sürü 10 parçacıktan oluşur ancak toplantı odasında sadece 5 parçacık etkileşime girer. Bu nedenle, Kaotik ÇSPSO’nun daha az hesaplama karmaşıklığı ile en iyi çözümü bulma konusunda daha iyi bir performansa sahip olduğu ifade edilebilir.

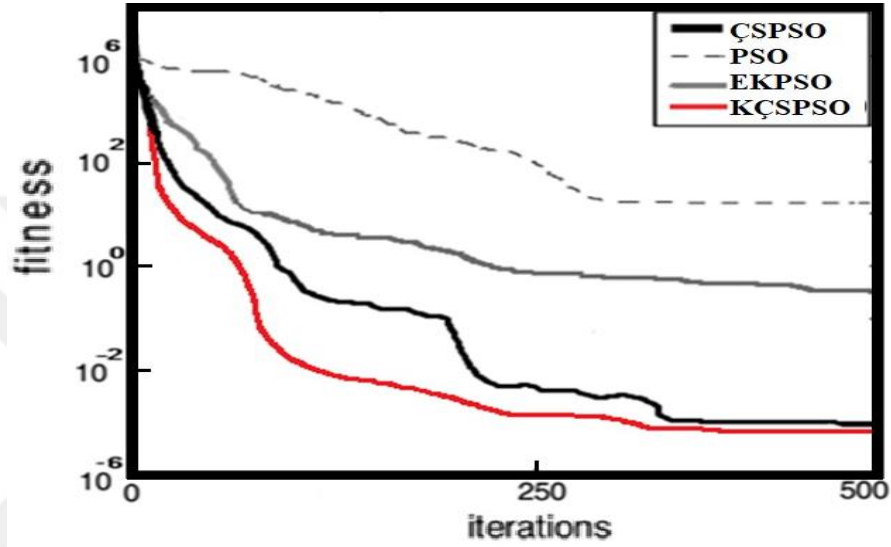
Tablo 7.3. Karşılaştırma testi fonksiyonları için sonuçlar

Boyut	f_n	Algoritma	En iyi	Ortalama	Standard Sapma
100	f_1	PSO	2,5457521	2,7647845	0,0784516
		EKPSO	0,9854126	1,0154784	0,0014784
		ÇSPSO	0,0007845	0,0008748	0,0000184
		Kaotik ÇSPSO	0,0005784	0,0007741	0,0000106
	f_2	PSO	0,0884741	0,0964587	0,0078478
		EKPSO	0,0078414	0,0087789	0,0009874
		ÇSPSO	0,0000897	0,0000997	0,0000658
		Kaotik ÇSPSO	0,0000564	0,0000845	0,0000621
	f_3	PSO	21,695847	27,947512	0,0847896
		EKPSO	2,0018977	2,6647845	0,0078487
		ÇSPSO	0,0004687	0,0045214	0,0000144
		Kaotik ÇSPSO	0,0003850	0,0040184	0,0000945
	f_4	PSO	16,4875218	26,110161	0,0238484
		EKPSO	1,99847	2,5869124	0,0084578
		ÇSPSO	0,0002648	0,0017636	0,0000584
		Kaotik ÇSPSO	0,0002645	0,0014213	0,0000315
500	f_1	PSO	7,2456571	2,7647845	0,0784516
		EKPSO	2,4859157	1,0154784	0,0014784
		ÇSPSO	0,0026472	0,0008748	0,0000184
		Kaotik ÇSPSO	0,0021645	0,0007945	0,0000021

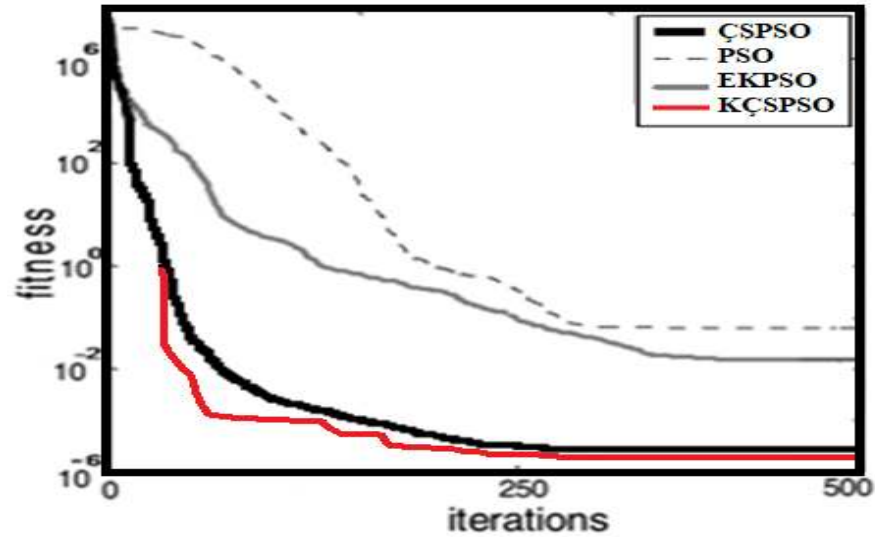
Tablo 7.3. devamı

500	f_2	PSO	1,2785781	0,0964587	0,0078478
		EKPSO	0,9045472	0,0087789	0,0009874
		ÇSPSO	0,0041816	0,0000997	0,0000658
		Kaotik ÇSPSO	0,0041443	0,0000487	0,0000584
	f_3	PSO	48,995751	27,947512	0,0847896
		EKPSO	7,2214945	2,6647845	0,0078487
		ÇSPSO	0,0784457	0,0045214	0,0000144
		Kaotik ÇSPSO	0,0706241	0,0040012	0,0000115
	f_4	PSO	37,125475	26,110161	0,0238484
		EKPSO	3,35847	2,5869124	0,0084578
		ÇSPSO	0,1778499	0,0017636	0,0000584
		Kaotik ÇSPSO	0,1084123	0,0014871	0,0000484
1000	f_1	PSO	16,422422	2,7647845	0,0784516
		EKPSO	4,7923729	1,0154784	0,0014784
		ÇSPSO	1,0749752	0,0008748	0,0000184
		Kaotik ÇSPSO	1,0054121	0,0007984	0,0000101
	f_2	PSO	3,0899761	0,0964587	0,0078478
		EKPSO	5,2574914	0,0087789	0,0009874
		ÇSPSO	0,9177297	0,0000997	0,0000658
		Kaotik ÇSPSO	0,9003451	0,0000854	0,0000584
	f_3	PSO	21,695847	27,947512	0,0847896
		EKPSO	2,0018977	2,6647845	0,0078487
		ÇSPSO	0,9563589	0,0045214	0,0000144
		Kaotik ÇSPSO	0,9034581	0,003945	0,0000112
	f_4	PSO	16,4875218	26,110161	0,0238484
		EKPSO	1,99847854	2,5869124	0,0084578
		ÇSPSO	0,48758311	0,0017636	0,0000584
		Kaotik ÇSPSO	0,48184724	0,0017603	0,0000587

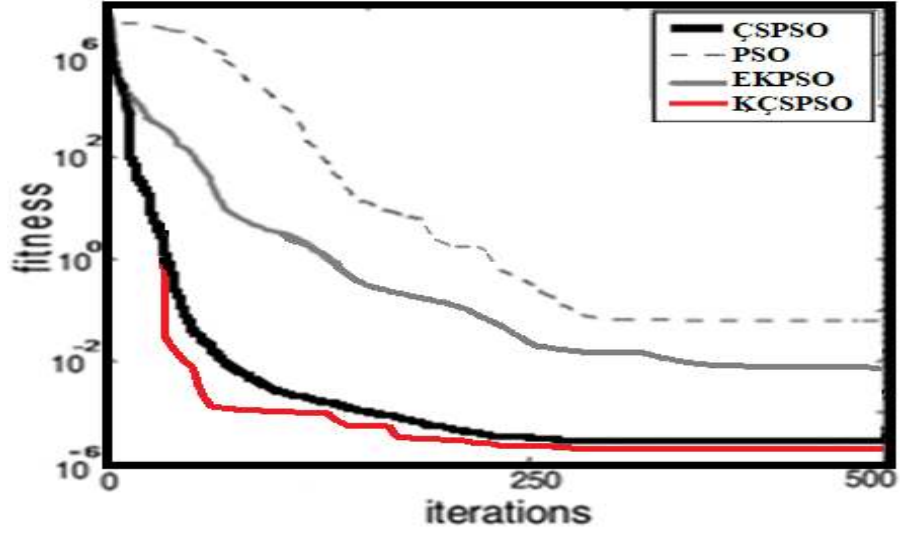
Şekil 7.2, 7.3, 7.4 ve 7.5, algoritmaların test fonksiyonları üzerindeki yakınsama eğrilerini sunmaktadır. Bu şekillere göre, önerilen KÇSPSO tüm test fonksiyonlarında diğer optimizasyon algoritmalarına kıyasla daha hızlı yakınsamaya sahiptir. KÇSPSO, özellikle başlangıçta orijinal ÇSPSO'dan daha iyi yakınsama göstermiştir. Bu durum, kaotik lojistik haritanın kullanılmasının algoritmaya katkı sağladığı ve orijinal ÇSPSO'ya kıyasla daha başarılı olduğu anlamına gelmektedir.



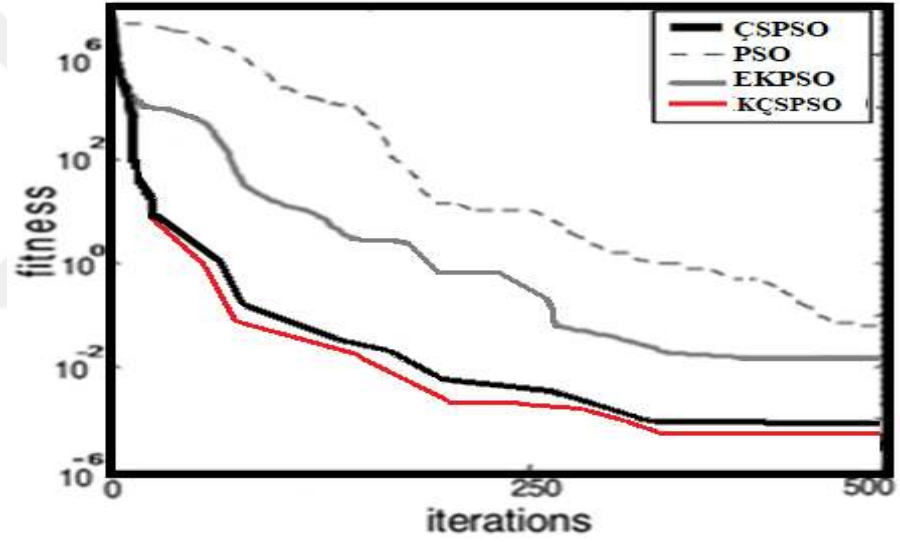
Şekil 7.2. Sphere test fonksiyonu için algoritmaların yakınsaklıkları



Şekil 7.3. Griewank test fonksiyonu için algoritmaların yakınsaklıkları

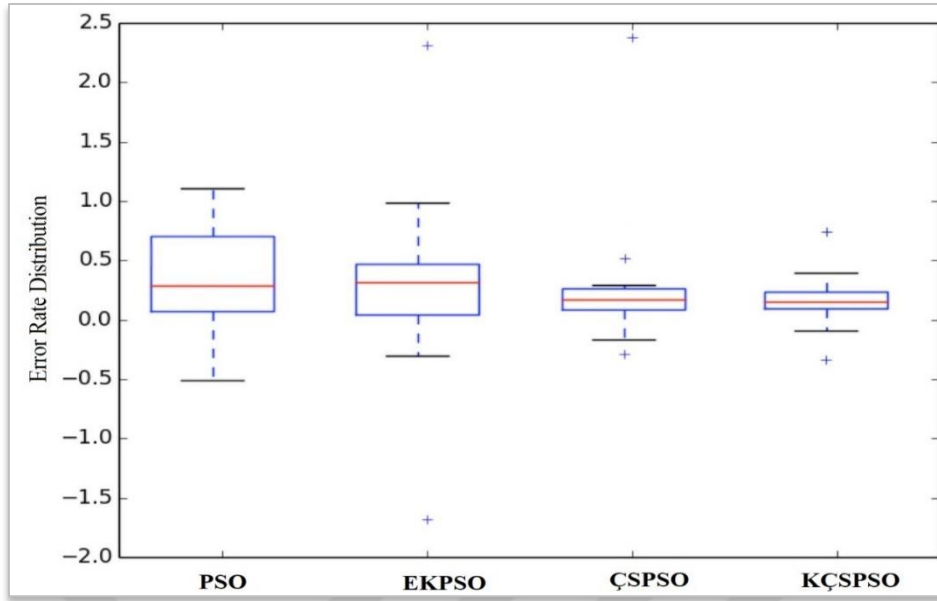


Şekil 7.4. Rastigrin test fonksiyonu için algoritmaların yakınsaklıkları



Şekil 7.5. Ackley test fonksiyonu için algoritmaların yakınsaklıkları

Şekil 7.6 algoritmaların istatistiki olarak karşılaştırılması için kutu kesitlerini göstermektedir. Bu şekilde de görüldüğü gibi KÇSPSO ve ÇSPSO algoritmalarının original PSO ve EKPSO algoritmalarından daha kararlı olduğu açıktır.



Şekil 7.6. Algoritmaların hata dağılımı grafiği

7.2. Kaynaklık Kiriş Tasarımı Problemi için Kaotik Çoklu Sürü Parçacık Sürü Optimizasyonu

KKT problemi için önerilen KÇSPSO'nun performansı aşağıdaki adımlar çerçevesinde değerlendirilmiştir.

1. Sürü Boyutu: Parçacık sayısının veya sürü büyüklüğünün etkisini test etmek için deneylerde 10, 20, 30, 40 ve 50 boyutunda sürüler kullanılmıştır.
2. Yineleme Sayısı: KKT problemini çözerken yinelemelerin önerilen algoritma üzerindeki etkilerini değerlendirmek için 100, 200, 250, 300 ve 500 değerlerinde yinelemeler kullanılmıştır.

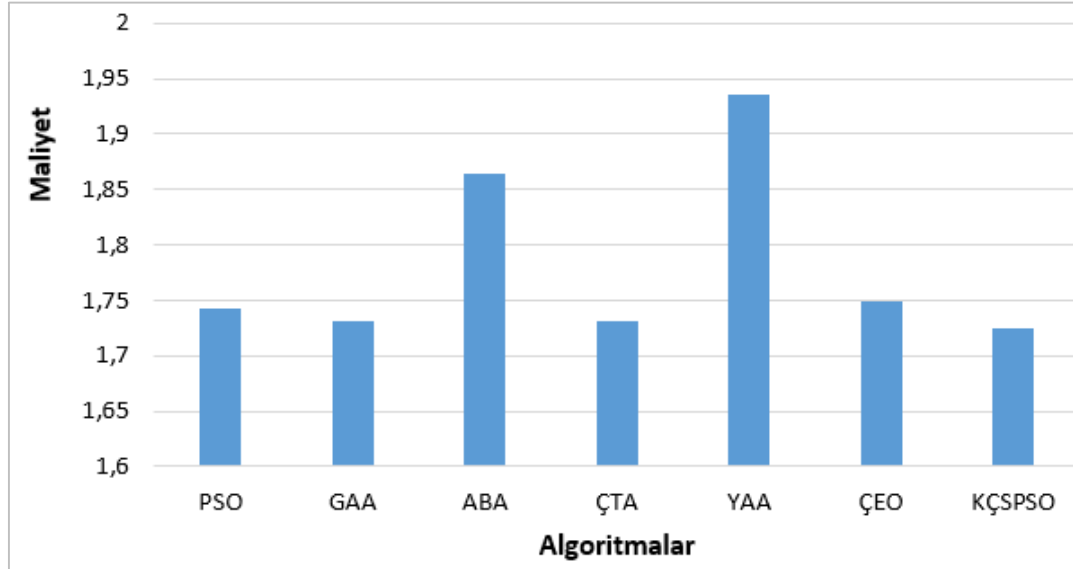
Ek-1'de her bir sürü boyutu ve yineleme sayısı ikilisi için önerilen algorithmadan elde edilen sonuçlar sunulmuştur. KÇSPSO tarafından elde edilen en iyi çözüm, 7 kısıtlama koşulları $g = (-13026,40322, -28740,99403, -0,0001, -3,385997356, -0,0805, -0,2494, -3796489,9974)$ ile $x_1=0,2055, x_2=3,4703, x_3=9,0700$ ve $x_4=0,2056$ olmak üzere 1,7292'dir.

Bu çalışmada ele alınan problem diğer optimizasyon yöntemleri kullanılarak da çözülmüş ve Kaotik lojistik harita tabanlı ÇSPSO'nun performansı bu algoritmalar ile

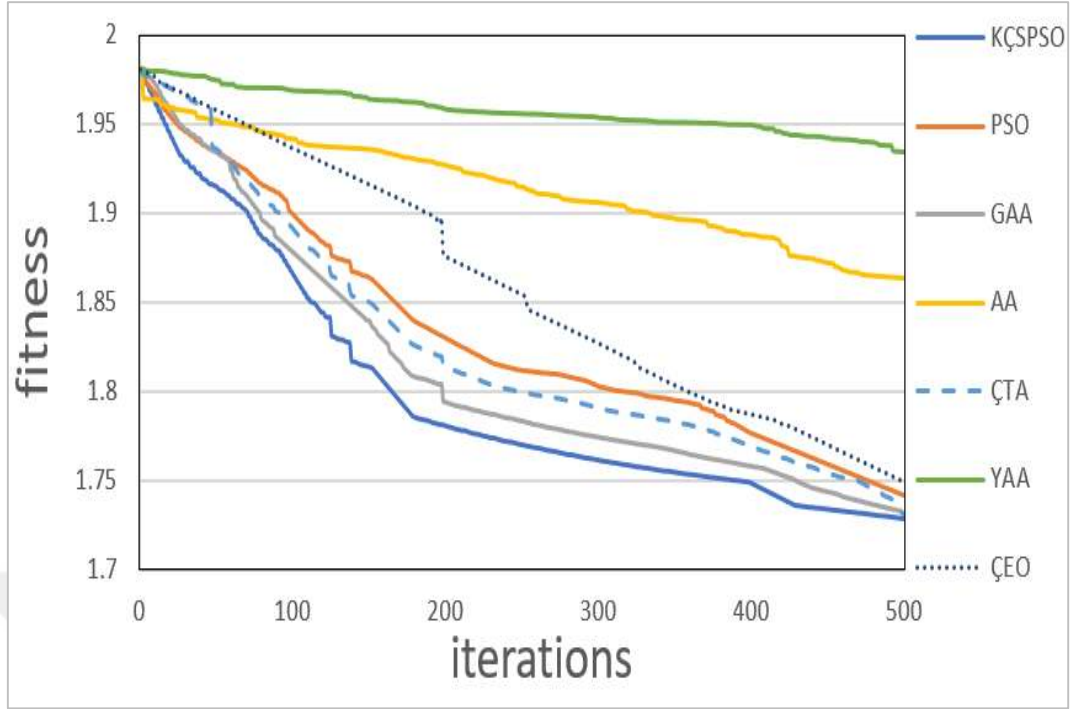
karşılaştırılmıştır. Önerilen KÇSPSO ile diğer mevcut meta sezgisel algoritmalar arasındaki karşılaştırmanın sonucu Tablo 7.4'te verilmiş ve ayrıca Şekil 7.7'de gösterilmiştir. Tablo 7.4'te önerilen KÇSPSO algoritmasının en düşük maliyet sunduğu görülmektedir. İkinci sırayı 1,7315 maliyet değeri ile GKAA ve ÇTA algoritmaları almıştır. Elde edilen sonuçlar diğer algoritmalarla kıyasla KÇSPSO'nun daha iyi bir performans sunduğunu bize göstermektedir.

Tablo 7.4. KKT problemi için algoritmaların simülasyon sonuçları

Metod	Tasarım Değişkenleri				
	x_1	x_2	x_3	x_4	Maliyet
PSO	0,2064	3,5283	8,9884	0,2080	1,7423
GKAA	0,2057	3,5194	9,0366	0,2057	1,7315
ABA	0,2416	3,6552	8,5071	0,2344	1,8641
ÇTA	0,2057	3,5195	9,0366	0,2057	1,7315
YAA	0,2195	4,7283	8,5009	0,2715	1,9350
ÇEO	0,1990	3,6529	9,1144	0,2054	1,7498
KÇSPSO	0,2055	3,4703	9,0700	0,2056	1,7292



Şekil 7.7. Meta sezgisel algoritmaların performanslarının karşılaştırılması



Şekil 7.8. KKT problemi için algoritmaların yakınsama eğrileri

Şekil 7.8 popülasyon boyutu 50, iterasyon sayısı 500 olarak belirlendiğinde KKT problemi için algoritmaların yakınsama eğrilerini göstermektedir. Her deney için, iterasyon tekrarı ile elde edilen sonuçlar kaydedilerek algoritmaların yakınsama eğrileri elde edilmiştir. Yakınsama eğrileri KÇSPSO algoritmasının üstünlüğünü ve onun 1.7292 maliyeti ile en iyi değeri sunduğunu bize göstermektedir.

KKT problemi için önerilen KÇSPSO yaklaşımının performansı literatürdeki diğer çalışmalar ile karşılaştırılmış ve bu karşılaştırma bilgileri Tablo 7.5'te sunulmuştur. Görüldüğü üzere, Toplum ve Medeniyet Algoritması (TMA) (Ray ve Liew 2003) ve PSO (He vd., 2004) hariç Yapay Arı Kolonisi (YAK) (Savsani, 2014), Evrim Stratejisi (ES) (Mezura-Montes ve Coello 2008), Gri Kurt Algoritması (GKA) (Mirjalili vd., 2014) ve Genetik Algoritma (GA) (Coello, 2000) çalışmaları ile önerilen yaklaşımın performansı birbirine oldukça yakın çıkmıştır. En iyi sonucu ise YAK algoritması vermiştir.

Tablo 7.5. KKT problemi için önerilen çalışmanın literatürdeki diğer çalışmalar ile karşılaştırılması

Metod	Tasarım Değişkenleri				
	x_1	x_2	x_3	x_4	Maliyet
GA (Coello, 2000)	0,2088	3,4205	8,9975	0,21	1,7483
ES (Mezura-Montes ve Coello, 2008)	0,1997	3,6120	9,0375	0,2060	1,7373
TMA (Ray ve Liew 2003)	0,2444	6,2379	8,2885	0,2445	2,3854
PSO (He vd., 2004)	0,2443	6,2175	8,2914	0,2443	2,3809
YAK (Savsani, 2014)	0,2057	3,4705	9,0366	0,2057	1,7248
GKA (Mirjalili vd., 2014)	0,2056	3,4783	9,0368	0,20577	1,7262
KÇSPSO	0,2055	3,4703	9,0700	0,2056	1,7292

Ayrıca, bu çalışmada ele alınan problem için en iyi, en kötü ve ortalama değerlerin literatürdeki diğer çalışmalar ile karşılaştırması Tablo 7.6’da verilmiştir. KKT problemi için bu tez çalışmasında önerilen KÇSPSO yöntemi YAK ve GKA yöntemlerinin bulduğu en iyi sonuca yakın sonuçlar bulmuştur. Bununla birlikte, bu yöntemin kararlılığının YAK ve GKA ile kıyaslandığında bu yöntemlerden daha iyi olduğu görülmektedir.

Tablo 7.6. KKT problem için en iyi, en kötü ve ortalama değerleri bakımından önerilen çalışmanın literatürdeki diğer çalışmalar ile karşılaştırılması

Metod	En iyi	Ortalama	En kötü	Std Sapma
GA (Coello, 2000)	1,7483	1,7719	1,7858	0,0112
ES (Mezura-Montes ve Coello, 2008)	1,7373	1,8133	1,9946	0,0705
TMA (Ray ve Liew 2003)	2,3854	3,2551	6,3996	0,96
PSO (He vd., 2004)	2,3809	2,3819	2,6458	0,0052
YAK (Savsani, 2014)	1,7248	1,7754	-	0,104087
GKA (Mirjalili vd., 2014)	1,7262	1,7255	-	0,100286
KÇSPSO	1,7292	1,7484	1,7339	0,0082

8. SONUÇLAR VE GELECEKTEKİ ÇALIŞMALAR

Yıllar boyunca çeşitli meta sezgi algoritma biçimleri önerilmiş ve geliştirilmiştir. Bunlar birçok mühendislik tasarımı problemini çözmek için kullanılmıştır. Meta sezgisel algoritmalar, doğadan esinlenen algoritmalar ve doğadan esinlenmeyen algoritmalar olmak üzere iki sınıfa ayrılabilir. Birinci sınıfa ait olan algoritmalar, kolonilerinden yiyecek kaynağına en kısa yolu bulmaya çalıştıklarında karıncaların davranışları gibi doğada akıllı bir davranıştan ilham alır. Ayrıca, çalışan arıların hareketlerinde, yiyeceklerini almak için iyi bir çiçek bulmaya çalıştıklarında ya da daha iyi kaynaklar bulmak için mevcut çiçeklerden uzak noktalarda aramaya çalıştıklarında akıllı bir davranış görülebilir. Bu iki örnek ve dahası araştırmacılar tarafından yeni metabilşimi geliştirmek için ilham kaynağı olarak kullanılmaktadır. Metasezgisel optimizasyon algoritmalarının avantajları, son zamanlarda birçok gerçek dünya oluşumunun simülasyonu ve bir popülasyonun ortak davranışını öğrenmek için kullanıldıkları için onları çekici hale getirmiştir.

Bu tez çalışmasında, bilinen bir mühendislik problemi olan KKT probleminin çözülmesi için doğadan esinlenmiş olan çok yaygın bir meta-sezgisel yaklaşım üzerinde durulmuş ve buna dayalı algoritma geliştirilmiştir. Bu araştırmanın ana katkısı, KKT problemini çözmek için kaotik lojistik harita denklemine dayalı KÇSPSO algoritmasını geliştirmek ve uygulamaktır.

ÇSPSO, her ne kadar normal ve büyük ölçekli optimizasyon problemlerini çözmek için test edilmiş olsa da her bir kümedeki parçacıkların başlangıç aşamasında kullanılan homojen dağılım denklemi nedeniyle yerel optimumda sıkışma ihtimali vardır. Bu çalışmada, tekdüze dağılım yerine lojistik bir harita olan kaotik harita denklemi kullanılmıştır. ÇSPSO için önerilen bu iyileştirme KÇSPSO olarak adlandırılmaktadır. Önerilen algoritma iki tür deney temelinde değerlendirilmiştir. İlk deneyde, KÇSPSO dört optimizasyon problemine dayanarak test edilmiş ve değerlendirilmiştir. En iyi, ortalama ve standart sapma hesaplanmıştır. İkinci deneyde, 25 farklı deneyde KKT problemini çözmek için KÇSPSO uygulanmıştır. Her deney için farklı sürü büyüklüğü (10, 20, 30, 40 ve 50) ve yineleme sayısı (100, 200, 250,

300 ve 500) kullanılmıştır. Bu deneylerin tümü Ekler bölümünde sunulmuştur. Son olarak, elde edilen en iyi sonuçlar, diğer optimizasyon algoritmalarıyla karşılaştırılmıştır. Genel olarak, çalışmadan aşağıdaki sonuçlar çıkarılmıştır:

- Önerilen KÇSPSO algoritması, optimizasyon problemlerinde uygulandığında PSO ve ÇSPSO algoritmalarına kıyasla en uygun çözümü sunmuştur.
- KÇSPSO hızlı yakınsama eğrisine sahiptir, genel ve yerel arama esnasında KÇSPSO'nun ÇSPSO ve PSO'ya göre daha dengeli olduğu sonucuna varılmıştır.
- Deneyler, arama sürecindeki yinelenmelerin sayısının belirgin bir etkisi olduğunu göstermiştir. Başka bir deyişle yineleme sayısı arttıkça sonuçlar daha da iyileşmiştir. Ayrıca, deneyler sürü büyüklüğünün algoritmanın performansını biraz etkilediğini de göstermiştir.
- KKT probleminde, KÇSPSO diğer algoritmalara kıyasla daha iyi sonuçlar sunmuştur.

Gelecekte aşağıdaki çalışmalara odaklanılacaktır:

- Basınçlı Kap Tasarımı ve Zaman-Maliyeti Dengelemesi gibi başka bir mühendislik optimizasyon problemini çözmek için KÇSPSO algoritmasının uygulanması hedeflenmektedir.
- KÇSPSO, lojistik haritasını başlangıç aşamasında kaotik bir harita olarak kullanmıştır. Gelecekte, çadır haritası veya parça bilge lojistik haritası gibi farklı kaotik haritaların incelenmesi planlanmaktadır.

Ayrıca, çoklu sürünün gelişmiş başlatma adımına sahip mimarisi, YAK gibi diğer optimizasyon algoritmalarının dengelenmesini geliştirmek için kullanılabilir. KÇSPSO, özellik seçimi veya ileri besleme sinir ağı gibi farklı optimizasyon problemlerinin çözümü için uygulanabilir.

KAYNAKLAR

- Aldwaik, M., & Adeli, H. (2014). Advances in optimization of highrise building structures. *Structural and Multidisciplinary Optimization*, 50(6), 899–919. <https://doi.org/10.1007/s00158-014-1148-1>
- Anghinolfi, D., Boccalatte, A., Grosso, A., Paolucci, M., & Passadore, Andrea Vecchiola, C. (2007). A Swarm Intelligence Method Applied to Manufacturing Scheduling, 8th IA/TABOO Joint Workshop “From Objects to Agents. In *Agents and Industry: Technological Applications of Software Agents*.
- Arango, C., Cortes, P., Onieva, L., & Escudero, A. (2013). Simulation-optimization models for the dynamic berth allocation problem. *Computer-Aided Civil and Infrastructure Engineering*, 28(10), 769–779. <https://doi.org/10.1111/mice.12049>
- Beheshti, Z., & Shamsuddin, S. M. H. (2013). A review of population-based meta-heuristic algorithm. *International Journal of Advances in Soft Computing and Its Applications*, 5(1), 1–35. <https://doi.org/2074-8523>
- Bonyadi, M. R., & Michalewicz, Z. (2017). Particle Swarm Optimization for Single Objective Continuous Space Problems: A Review *The MIT Press Journals* 1063-6560 https://doi.org/10.1162/EVCO_r_00180
- Chai, X., Chen, Y., & Broyde, L. (2017). A novel chaos-based image encryption algorithm using DNA sequence operations. *Optics and Lasers in Engineering*, 88, 197–213. <https://doi.org/10.1016/j.optlaseng.2016.08.009>
- Chen, X., Zhang, L., He, X., Xiong, C., & Li, Z. (2014). Surrogate-Based Optimization of Expensive to Evaluate Objective for Optimal Highway Toll Charges in Transportation Network. *Computer-Aided Civil and Infrastructure Engineering*, 29(5), 359–381. <https://doi.org/10.1111/mice.12058>
- Chen, W. J., Su, W. C., Nian, F. L., Lin, J. R., & Chen, D. C. (2013). Application of ANOVA and Taguchi-Based Mutation Particle Swarm Algorithm for Parameters Design of Multi-Hole Extrusion Process. *Research Journal of Applied Sciences, Engineering and Technology*, 6(13), 2316–2325.
- Chow, J. Y. J. (2014). Activity-based travel scenario analysis with routing problem reoptimization. *Computer-Aided Civil and Infrastructure Engineering*, 29(2), 91–106. <https://doi.org/10.1111/mice.12023>

- Coello, C. A. C. (2000). Treating constraints as objectives for single-objective evolutionary optimization. *Engineering Optimization*, 32(3), 275–308. <https://doi.org/10.1080/03052150008941301>
- Džugan, J., Španiel, M., Prantl, A., Konopík, P., Růžicka, J., & Kuželka, J. (2018). Identification of ductile damage parameters for pressure vessel steel. *Nuclear Engineering and Design*. <https://doi.org/10.1016/j.nucengdes.2015.12.014>
- Eberhart, R., & Kennedy, J. (1995). A new optimizer using particle swarm theory. In *Micro Machine and Human Science, 1995. MHS'95., Proceedings of the Sixth International Symposium on* (pp. 39–43).
- Fadaee, M., & Radzi, M. A. M. (2012). Multi-objective optimization of a stand-alone hybrid renewable energy system by using evolutionary algorithms: A review. *Renewable and Sustainable Energy Reviews* 16(5), 3364–3369. <https://doi.org/10.1016/j.rser.2012.02.071>
- Fister, I., Yang, X. S., Fister, D., & Fister, I. (2013). Cuckoo search: A brief literature review. In *Studies in Computational Intelligence*, 516, 49–62. https://doi.org/10.1007/978-3-319-02141-6_3
- Gandomi, A. H., Yang, X. S., Talatahari, S., & Alavi, A. H. (2013a). Firefly algorithm with chaos. *Communications in Nonlinear Science and Numerical Simulation*, 18(1), 89–98. <https://doi.org/10.1016/j.cnsns.2012.06.009>
- Gandomi, A. H., Yun, G. J., Yang, X. S., & Talatahari, S. (2013b). Chaos-enhanced accelerated particle swarm optimization. *Communications in Nonlinear Science and Numerical Simulation*, 18(1), 327–340. <https://doi.org/10.1016/j.cnsns.2012.07.017>
- Gandomi, A. H., & Yang, X. S. (2014). Chaotic bat algorithm. *Journal of Computational Science*, 5(2), 224–232. <https://doi.org/10.1016/j.jocs.2013.10.002>
- Gao, L., Zhou, C., & Zan, K. (2006). Particle Swarm Optimization for Simultaneous Optimization of Design and Machining Tolerances. In *SEAL Proceedings of the 6th International Conference on Simulated Annealing and learning* (pp. 321–329).
- Garg, S., Patra, K., & Pal, S. K. (2014). Particle swarm optimization of a Neural Network Model in a Machining Process. *Sadhana Publications*, 39(3), 533–548.
- Gen, M., & Yun, Y. S., (2006). Soft computing approach for reliability optimization: State-of-the-art survey. *Reliability Engineering and System Safety*, 91(9), 1008–1026. <https://doi.org/10.1016/j.res.2005.11.053>

- Guo, Y. W., Li, W. D., Mileham, A. R., & Owen, G. W. (2009). Applications of Particle Swarm Optimization in Integrated Process Planning and Scheduling. *Robotics and Integrated Manufacturing*, 25(2), 280–288.
- He, S., Prempan, E., & Wu, Q. H. (2004). An improved particle swarm optimizer for mechanical design optimization problems. *Engineering Optimization*, 36(5), 585–605. <https://doi.org/10.1080/03052150410001704854>
- Janson, S., & Middendorf, M. (2005). A Hierarchical Particle Swarm Optimizer and Its Adaptive Variant. *IEEE Transactions on Systems, Man, and Cybernetics—part b: Cybernetics*, 35(6), 1272–1280.
- Jayabarathi, T., Raghunathan, T., & Gandomi, A. H. (2017). The bat algorithm, variants and some practical engineering applications: A review. In *Studies in Computational Intelligence (Book Chapter)* 774, 313–330. https://doi.org/10.1007/978-3-319-67669-2_14
- Kazak, A., Alpaslan, N., & Alpaslan, D. (2015). Geliştirilmiş Yerçekimsel Arama Algoritması: MSS-GSA. / *ISITES2015Valencia -Spain* 1823-1830.
- Keshtegar, B., & Hao, P. (2018). Enriched self-adjusted performance measure approach for reliability-based design optimization of complex engineering problems. *Applied Mathematical Modelling*, 57(1), 37–51. <https://doi.org/10.1016/j.apm.2017.12.030>
- Lian, Z., Gu, X., & Jiao, B. (2006). A similar particle swarm optimization algorithm for permutation flowshop scheduling to minimize makespan. *Applied Mathematics and Computation* (175), 773–785.
- Liu, C. Y., Yan, C. Q., Wang, J. J. (2013). Hybrid Particle Swarm Optimization Algorithm and its Application in Nuclear Engineering. *Annals of Nuclear Energy*, 64, 276–286.
- Luo, D., Ibrahim, Z., Ismail, Z., & Xu, B. (2013). Optimization of the geometries of biconical tapered fiber sensors for monitoring the early-age curing temperatures of concrete specimens. *Computer-Aided Civil and Infrastructure Engineering*, 28(7), 531–541. <https://doi.org/10.1111/mice.12022>
- Manjunath K., S., Mohan, K., & Channakeshava, K. R. (2011). Optimization of Ply Stacking Sequence of Composite Drive Shaft Using Particle Swarm Algorithm. *Journal Of Engineering Science And Technology*, 6(3), 323 – 331.
- Mezura-Montes, E., & Coello, C. A. C. (2008). An empirical study about the usefulness of evolution strategies to solve constrained optimization problems. *International Journal of General Systems* 37(4), 443–473.

<https://doi.org/10.1080/03081070701303470>

- Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in Engineering Software*, 69, 46–61. <https://doi.org/10.1016/j.advengsoft.2013.12.007>
- Mirjalili, S., Mirjalili, S. M., & Hatamlou, A. (2016). Multi-Verse Optimizer: a nature-inspired algorithm for global optimization. *Neural Computing and Applications*, 27(2), 495–513. <https://doi.org/10.1007/s00521-015-1870-7>.
- Murillo-Escobar, M. A., Cruz-Hernández, C., Cardoza-Avendaño, L., & Méndez-Ramírez, R. (2017). A novel pseudorandom number generator based on pseudorandomly enhanced logistic map. *Nonlinear Dynamics*, 87(1), 407–425. <https://doi.org/10.1007/s11071-016-3051-3>
- Nematollahi, A. F., Rahiminejad, A., & Vahidi, B. (2017). A novel physical based meta-heuristic optimization method known as Lightning Attachment Procedure Optimization. *Applied Soft Computing Journal*, 59, 596–621. <https://doi.org/10.1016/j.asoc.2017.06.033>
- Noor, M. M., Kadirgama, K., & Rahman, M. M. (2011). Particle swarm Optimization Prediction Model for Surface Roughness, *International Journal of the Physical Sciences*, 6(13), 3082–3090.
- Palupi Rini, D., Shamsuddin, M., & Yuhaniz, S. (2011). *Particle Swarm Optimization: Technique, System and Challenges*. *International Journal of Computer Applications*, 14(1), 19–27. <https://doi.org/10.5120/1810-2331>
- Pareek, N. K., Patidar, V., & Sud, K. K. (2006). Image encryption using chaotic logistic map. *Image and Vision Computing*, 24(9), 926–934. <https://doi.org/10.1016/j.imavis.2006.02.021>
- Peng, F., & Ouyang, Y. (2014). Optimal clustering of railroad track maintenance jobs. *Computer-Aided Civil and Infrastructure Engineering*, 29(4), 235–247. <https://doi.org/10.1111/mice.12036>
- Poli, R., Kennedy, J., & Blackwell, T. (2007). Particle swarm optimization. *Swarm* 200, 1(1), 33–57. <https://doi.org/10.1007/s11721-007-0002-0>
- Ramanathan, K., Periasamy, V. M., Pushpavanam, M., & Natarajan, U. (n.d.). (2009). Particle Swarm Optimization of Hardness in Nickel Diamond Electro Composites. *International Journal of Computational Materials Science and Surface Engineering*, 1(4), 232–236.
- Ray, T., & Liew, K. M. (2003). Society and civilization: an optimization algorithm

- based on the simulation of social behavior. *IEEE Transactions on Evolutionary Computation*, 7(4), 386–396.
<https://doi.org/10.1109/TEVC.2003.814902>
- Rekliatis, G. V., Ravindrab, A., and Ragsdell, K. M. (1983). *Engineering optimisation methods and applications*, (Wiley, New York).
- Salih, S. Q., & Alsewari, A. A. (2018). Solving large-scale problems using multi-swarm particle swarm approach. *International Journal of Engineering & Technology*, 7(3), 1725–1729.
<https://www.sciencepubco.com/index.php/IJET>
- Salih, S. Q., Alsewari, A. A., Al-Khateeb, B., & Zolkipli, M. F. (2019a). Novel Multi-swarm Approach for Balancing Exploration and Exploitation in Particle Swarm Optimization. In F. Saeed, N. Gazem, F. Mohammed, & A. Busalim (Eds.), *Recent Trends in Data Science and Soft Computing* (pp. 196–206). Cham: Springer International Publishing.
- Salih, S. Q., Alsewari, A. A., Al-Khateeb, B., & Zolkipli, M. F. (2019b). Novel Multi-swarm Approach for Balancing Exploration and Exploitation in Particle Swarm Optimization. In *Recent Trends in Data Science and Soft Computing (Book Chapter)* (pp. 196–206). Cham: Springer International Publishing.
- Sarayloo, F., & Tavakkoli-Moghaddam, R. (2010). Multi Objective Particle Swarm Optimization for a Dynamic Cell Formation Problem. In *Proceedings of the World Congress on Engineering*, London, U.K., (pp. 1–6).
- Savsani, V. (2014). Implementation of modified artificial bee colony (ABC) optimization technique for minimum cost design of welded structures. *International Journal for Simulation and Multidisciplinary Design Optimization*, 5. <https://doi.org/10.1051/smdo/2013008>
- Shi, Y., & Eberhart, R. (1998). A Modified Particle Swarm Optimizer. *Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence.*, The 1998 IEEE International Conference On, 69–73.
<https://doi.org/10.1109/ICEC.1998.699146>
- Shu, F. (2007). Aluminum-zinc Alloy Squeeze Casting Technological Parameters Optimization based on PSO and ANN. *China Foundry*, 4(3), 202–205.
- Smarandache, F., Abdel-Basset, M., & Chang, V. (2017) *Neutrosophic Operational Research* Pons Publishing House / Pons asbl Quai du Batelage, 5 1000 - Bruxelles Belgium DTP: George Lukacs ISBN 978-1-59973-5375
- Smith, R., Ferrebee, E., Ouyang, Y., & Roesler, J. (2014). Optimal Staging Area Locations and Material Recycling Strategies for Sustainable Highway

Reconstruction. *Computer-Aided Civil and Infrastructure Engineering*, 29(8), 559–571. <https://doi.org/10.1111/mice.12089>

Spina, R. (2006). Optimization of Injection Moulded Parts by using ANN-PSO approach. *Journals of Achievements in Materials and Manufacturing Engineering*, 15(1–2), 146–152.

Srirat, J., Yamazaki, K., & Kitayama, S. (2011). The Optimization of Blank Holder Gap and Segmented Blank Holder Shape in Deep-Drawing Process Considering Failures and Product Quality. In *9th World Congress on Structural and Multidisciplinary Optimization*.

Tambolia, K., Patel, S., George, P. M., & Sanghvia, R. (2014). Optimal Design of a Heavy Duty Helical Gear Pair using Particle Swarm Optimization Technique. In *2nd International Conference on Innovations in Automation and Mechatronics Engineering*, ICIAME, 14, 513–519.

Tanweer, M. R., Suresh, S., & Sundararajan, N. (2015). Improved SRPSO algorithm for solving CEC 2015 computationally expensive numerical optimization problems. In *2015 IEEE Congress on Evolutionary Computation, CEC 2015 - Proceedings* (pp. 1943–1949). <https://doi.org/10.1109/CEC.2015.7257123>

Towler, G., & Sinnott, R. (2013). Design of Pressure Vessels. In *Chemical Engineering Design: Principles, Practice and Economics of Plant and Process Design* (pp. 563–627). <https://doi.org/10.1016/B978-0-08-096659-5.00014-6>

URL-1. Optimization Test Problems, 05/07/2019 tarihinde <https://www.sfu.ca/~ssurjano/optimization.html> adresinden alınmıştır.

Van Den, B. (2006). An Analysis of Particle Swarm Optimizers.

Van Den, B. & F. Engelbrecht, A. P. (2002). A New Locally Convergent Particle Swarm Optimizer. In *Proceedings of the IEEE International Conference on systems, man and cybernetics* (pp. 6–9).

Venkatesan, R. (2010). Extrusion Die profile Design using Simulated Annealing and Particle Swarm Optimization. *International Journal of Engineering Science and Technology*, 2(8), 3758–3761.

Wang, X., & Qiu, X. (2013). Application of Particle Swarm Optimization for Enhanced Cyclic Steam Stimulation in a Offshore Heavy Oil Reservoir. *International Journal of Information Technology, Modeling and Computing*, 1(2).

- Watkins, C. J. C. H. Learning from delayed rewards. PhD Thesis, University of Cambridge England, 1989.
- Yang, X. S. (2009). Firefly algorithms for multimodal optimization. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 5792 LNCS, 169–178. https://doi.org/10.1007/978-3-642-04944-6_14
- Yang, X. S., & Deb, S. (2009). Cuckoo search via lévy flights. In 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC) (pp. 210–214). <https://doi.org/10.1109/NABIC.2009.5393690>
- Yang, X. S. (2010a). Firefly algorithm, stochastic test functions and design optimisation. *International Journal of Bio-Inspired Computation*, 2(2), 78–84.
- Yang, X. S. (2010b). A new metaheuristic bat-inspired algorithm. In *Studies in Computational Intelligence*, 284, 65–74. https://doi.org/10.1007/978-3-642-12538-6_6
- Yang, X. S. (2010c). Firefly algorithm, Levy flights and global optimization. *Research and Development in Intelligent Systems*, 135–146. <https://doi.org/10.1007/978-1-84882-983-1>
- Yang, X. S. (2012) Flower pollination algorithm for global optimization, in: *Unconventional Computation and Natural Computation 2012, Lecture Notes in Computer Science*, vol. 7445, pp. 240–249 (2012). https://link.springer.com/chapter/10.1007/978-3-642-32894-7_27
- Yang, X. S., Deb, S., & He, X. (2013). Eagle strategy with flower algorithm. In *Proceedings of the 2013 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2013* (pp. 1213–1217). <https://doi.org/10.1109/ICACCI.2013.6637350>
- Yarmohamadi, M., Javadi, H. S., & Erfani, H., (2011). Improvement of Robot Path Planning using Particle Swarm Optimization in Dynamic Environments with Mobile Obstacles and Target-. *Advanced Studies in Biology*, 3(1), 43–53.

EKLER

EK-1. Deneysel Çalışma Sonuçları



EK-1. Deneysel Çalışma Sonuçları

Deney 1:

Sürü büyüklüğü 10'a eşit ve yineleme sayısı 100'e eşit olduğu 10 deneme sonuçları aşağıdaki Tablo EK-1.1'de sunulmuştur.

Tablo Ek-1.1. Birinci deneyin sonuçları

Deneme	X_1	X_2	X_3	X_4	Maliyet
1	0,2064	3,5283	8,9884	0,2066953	1,74212089
2	0,205571	3,526654	8,993295	0,2069494	1,74222327
3	0,206497	3,525009	8,998005	0,2060911	1,74225121
4	0,206125	3,521511	8,995195	0,2072078	1,74219312
5	0,205808	3,528856	8,992545	0,2074401	1,74229993
6	0,206406	3,521995	8,993771	0,20652	1,74205301
7	0,205933	3,527818	8,996057	0,2068917	1,74218234
8	0,206347	3,528094	8,999209	0,2065191	1,7422409
9	0,205992	3,524897	8,997875	0,2068523	1,74224643
10	0,206164	3,522699	8,991319	0,2070359	1,74203932

Deney 2:

Sürü büyüklüğü 20'ye eşit ve yinleme sayısı 100'e eşit olduğu 10 deneme sonuçları aşağıdaki Tablo EK-1.2'de sunulmuştur.

Tablo EK-1.2. İkinci deneyin sonuçları

Deneme	X_1	X_2	X_3	X_4	Maliyet
1	0,205996763	3,523719953	8,992612733	0,206052255	1,742035963
2	0,20639312	3,523918349	8,99844976	0,207259366	1,742298406
3	0,206286372	3,529019415	8,999027538	0,206358471	1,742170484
4	0,206170819	3,52227591	8,999154162	0,206525728	1,742235506
5	0,206160184	3,529402254	8,992403662	0,206630878	1,742227231
6	0,206091975	3,521025081	8,990230712	0,206398172	1,7420773
7	0,206037807	3,528401913	8,998374405	0,206804595	1,742139838
8	0,206423213	3,520932498	8,999027077	0,207331105	1,742253871
9	0,205556802	3,523231794	8,996150306	0,206320886	1,742020679
10	0,206203198	3,521790938	8,997627954	0,206295489	1,742078512

Deney 3:

Sürü büyüklüğü 30'a eşit ve yineleme sayısı 100'e eşit olduğu 10 deneme sonuçları aşağıdaki Tablo EK-1.3'te sunulmuştur.

Tablo EK-1.3. Üçüncü deneyin sonuçları

Deneme	X_1	X_2	X_3	X_4	Maliyet
1	0,206342799	3,52955487	8,999006541	0,207486842	1,742073343
2	0,205516935	3,52669728	8,995749167	0,207438581	1,742058863
3	0,206462202	3,52758543	8,992933985	0,206353253	1,742195186
4	0,206147497	3,525897691	8,99563259	0,207010347	1,742000078
5	0,206309103	3,520865036	8,997140604	0,207010947	1,742155868
6	0,205681327	3,523687868	8,99014212	0,206693633	1,742278521
7	0,206014563	3,523959996	8,999004424	0,207365564	1,742127195
8	0,20562974	3,52791373	8,994110501	0,207206839	1,74225873
9	0,2056835	3,520474697	8,995022516	0,207019946	1,742289603
10	0,206342799	3,52955487	8,999006541	0,207486842	1,742073343

Deney 4:

Sürü büyüklüğü 40'a eşit ve yineleme sayısı 100'e eşit olduğu 10 deneme sonuçları aşağıdaki Tablo EK-1.4'te sunulmuştur

Tablo EK-1.4. Dördüncü deneyin sonuçları

Deneme	X_1	X_2	X_3	X_4	Maliyet
1	0,205614	3,524788	8,990785	0,2070208	1,74202087
2	0,205915	3,523012	8,9929	0,2064809	1,74214116
3	0,205862	3,52576	8,988978	0,2071368	1,74215784
4	0,20594	3,529917	8,993136	0,2071252	1,74224641
5	0,205804	3,526823	8,993116	0,2072306	1,74200779
6	0,206164	3,528141	8,99306	0,2069961	1,74200934
7	0,205598	3,522195	8,991428	0,2063085	1,74202779
8	0,205682	3,52843	8,995111	0,2072083	1,74211996
9	0,206068	3,52104	8,990486	0,2063525	1,74206665
10	0,205614	3,524788	8,990785	0,2070208	1,74202087

Deney 5:

Sürü büyüklüğü 50'ye eşit ve yineleme sayısı 100'e eşit olduğu 10 deneme sonuçları aşağıdaki Tablo EK-1.5'te sunulmuştur.

Tablo EK-1.5. Beşinci deneyin sonuçları

Deneme	X_1	X_2	X_3	X_4	Maliyet
1	0,205636	3,520118	8,989195	0,2067235	1,74202489
2	0,205517	3,526418	8,993853	0,2074352	1,74205067
3	0,206035	3,528585	8,994841	0,2069792	1,74226705
4	0,20585	3,52915	8,997579	0,2060175	1,74219456
5	0,205721	3,521225	8,992407	0,2064178	1,74225737
6	0,206242	3,527125	8,993248	0,2070916	1,74212221
7	0,206455	3,522465	8,989625	0,2068261	1,742025
8	0,205681	3,527986	8,989478	0,2068081	1,74210115
9	0,206067	3,523956	8,995011	0,2073723	1,74208667
10	0,206366	3,524336	8,996382	0,2069422	1,74206708

Deney 6:

Sürü büyüklüğü 10'a eşit ve yineleme sayısı 200'e eşit olduğu 10 deneme sonuçları aşağıdaki Tablo EK-1.6'da sunulmuştur.

Tablo EK-1.6. Altıncı deneyin sonuçları

Deneme	X_1	X_2	X_3	X_4	Maliyet
1	0,206253	3,518224	8,999874	0,2074498	1,74187247
2	0,205825448	3,519112114	9,000114321	0,206260525	1,741808196
3	0,20584941	3,52075473	8,999956559	0,206925208	1,741998654
4	0,205830463	3,521071806	9,000078634	0,206638519	1,74171674
5	0,206761868	3,518952398	8,999992858	0,207406835	1,741778076
6	0,206038391	3,524696763	8,999935049	0,207174412	1,741703967
7	0,206252531	3,518172587	8,999915228	0,206312468	1,741811738
8	0,205522589	3,519933146	8,999827141	0,206883119	1,74177354
9	0,206666536	3,518949702	8,999816728	0,206250846	1,74175859
10	0,206601589	3,518904369	8,999951366	0,206293042	1,741877186

Deney 7:

Sürü büyüklüğü 20'ye eşit ve yinleme sayısı 200'e eşit olduğu 10 deneme sonuçları aşağıdaki Tablo EK-1.7'de sunulmuştur.

Tablo EK-1.7. Yedinci deneyin sonuçları

Deneme	X_1	X_2	X_3	X_4	Maliyet
1	0,206169501	3,519054968	8,999994188	0,207389966	1,741797816
2	0,206180565	3,524391347	8,999831321	0,207356055	1,741859187
3	0,205645614	3,524439529	9,000096283	0,20737751	1,741725923
4	0,206427265	3,522216898	8,999829175	0,206965652	1,741878848
5	0,206211203	3,52075873	8,999893208	0,207047057	1,741982837
6	0,205964431	3,524522025	9,000088846	0,207235769	1,74170236
7	0,206482633	3,518212012	8,99994474	0,206644524	1,741906076
8	0,206263353	3,523463229	9,000045587	0,206741154	1,741980631
9	0,205619748	3,520263986	9,000148247	0,206576774	1,741824889
10	0,205807075	3,520454675	8,999884597	0,206013855	1,741780345

Deney 8:

Sürü büyüklüğü 30'a eşit ve yineleme sayısı 200'e eşit olduğu 10 deneme sonuçları aşağıdaki Tablo EK-1.8'de sunulmuştur.

Tablo EK-1.8. Sekizinci deneyin sonuçları

Deneme	X_1	X_2	X_3	X_4	Maliyet
1	0,205728703	3,523198656	9,000130067	0,206317108	1,741996546
2	0,205637056	3,521929196	8,999901135	0,206300555	1,741945817
3	0,206781885	3,523888119	8,999981248	0,2067907	1,741733439
4	0,206510834	3,51819381	9,00007173	0,206238963	1,741956747
5	0,205542537	3,519793688	9,000033085	0,206464041	1,741947569
6	0,206713705	3,524460598	9,000115372	0,206775119	1,741963336
7	0,206517695	3,51889633	9,000171825	0,207022673	1,741946064
8	0,206125525	3,521057713	8,999848725	0,206577328	1,741957449
9	0,206729042	3,521343367	8,999842399	0,206712437	1,741827596
10	0,205614853	3,518494583	8,999904899	0,207361393	1,741889885

Deney 9:

Sürü büyüklüğü 40'a eşit ve yineleme sayısı 200'e eşit olduğu 10 deneme sonuçları aşağıdaki Tablo EK-1.9'da sunulmuştur.

Tablo EK-1.9. Dokuzuncu deneyin sonuçları

Deneme	X ₁	X ₂	X ₃	X ₄	Maliyet
1	0,205728703	3,523198656	9,000130067	0,206317108	1,741996546
2	0,205637056	3,521929196	8,999901135	0,206300555	1,741945817
3	0,206781885	3,523888119	8,999981248	0,2067907	1,741733439
4	0,206510834	3,51819381	9,00007173	0,206238963	1,741956747
5	0,205542537	3,519793688	9,000033085	0,206464041	1,741947569
6	0,206713705	3,524460598	9,000115372	0,206775119	1,741963336
7	0,206517695	3,51889633	9,000171825	0,207022673	1,741946064
8	0,206125525	3,521057713	8,999848725	0,206577328	1,741957449
9	0,206729042	3,521343367	8,999842399	0,206712437	1,741827596
10	0,205614853	3,518494583	8,999904899	0,207361393	1,741889885

Deney 10:

Sürü büyüklüğü 50'ye eşit ve yineleme sayısı 200'e eşit olduğu 10 deneme sonuçları aşağıdaki Tablo EK-1.10'da sunulmuştur.

Tablo EK-1.10. Onuncu deneyin sonuçları

Deneme	X_1	X_2	X_3	X_4	Maliyet
1	0,205382387	3,524749819	8,999869831	0,206066396	1,741044307
2	0,206454782	3,522555374	9,000155426	0,206954469	1,741857555
3	0,206153098	3,518813335	8,999924374	0,206158949	1,741931503
4	0,20619675	3,521719157	8,999825954	0,206661145	1,741926336
5	0,205515389	3,519544067	9,000188044	0,20682967	1,741785751
6	0,206330563	3,521315895	9,000006425	0,206601644	1,741706495
7	0,205642592	3,521687954	9,000036498	0,206268004	1,741787121
8	0,205520524	3,522400039	9,000156677	0,206557846	1,741930999
9	0,206180764	3,519550248	8,99981866	0,206915446	1,741949792
10	0,206628708	3,522554967	9,000103761	0,206110182	1,741732175

Deney 11:

Sürü büyüklüğü 10'a eşit ve yineleme sayısı 250'ye eşit olduğu 10 deneme sonuçları aşağıdaki Tablo EK-1.11'de sunulmuştur.

Tablo EK-1.11. On birinci deneyin sonuçları

Deneme	X ₁	X ₂	X ₃	X ₄	Maliyet
1	0,205382387	3,524749819	8,999869831	0,206066396	1,741044307
2	0,206454782	3,522555374	9,000155426	0,206954469	1,741857555
3	0,206153098	3,518813335	8,999924374	0,206158949	1,741931503
4	0,20619675	3,521719157	8,999825954	0,206661145	1,741926336
5	0,205515389	3,519544067	9,000188044	0,20682967	1,741785751
6	0,206330563	3,521315895	9,000006425	0,206601644	1,741706495
7	0,205642592	3,521687954	9,000036498	0,206268004	1,741787121
8	0,205520524	3,522400039	9,000156677	0,206557846	1,741930999
9	0,206180764	3,519550248	8,99981866	0,206915446	1,741949792
10	0,206628708	3,522554967	9,000103761	0,206110182	1,741732175

Deney 12:

Sürü büyüklüğü 20'e eşit ve yineleme sayısı 250'ye eşit olduğu 10 deneme sonuçları aşağıdaki Tablo EK-1.12'de sunulmuştur.

Tablo EK-1.12. Onikinci deneyin sonuçları

Deneme	X_1	X_2	X_3	X_4	Maliyet
1	0,205816499	3,518344682	9,009199721	0,206530408	1,737256154
2	0,206580482	3,502010038	9,003680318	0,206785989	1,737160982
3	0,205988292	3,514434682	9,006858596	0,207083585	1,735433026
4	0,206491904	3,499353614	9,003196423	0,206747805	1,73535187
5	0,206254053	3,500126512	9,005028237	0,207250262	1,735261621
6	0,205752442	3,51956308	9,006330143	0,207268194	1,736878427
7	0,206182272	3,505071104	9,005689492	0,2070505	1,736305119
8	0,206366771	3,510645722	9,011690993	0,206277756	1,735486325
9	0,206020869	3,509828511	9,01141649	0,206528422	1,738313167
10	0,20648801	3,518934789	9,011801912	0,206264272	1,735769501

Deney 13:

Sürü büyüklüğü 30'a eşit ve yineleme sayısı 250'ye eşit olduğu 10 deneme sonuçları aşağıdaki Tablo EK-1.13'te sunulmuştur.

Tablo EK-1.13. Onüçüncü deneyin sonuçları

Deneme	X_1	X_2	X_3	X_4	Maliyet
1	0,20637752	3,5007898	9,001673741	0,206576044	1,736950968
2	0,206465099	3,519281057	9,000802613	0,20677963	1,738698249
3	0,206438378	3,50791144	9,009326319	0,20668956	1,737190221
4	0,206132844	3,51821636	9,009314056	0,207194201	1,738242992
5	0,206688564	3,505280005	9,002259542	0,206395913	1,738608485
6	0,206237868	3,499383594	9,002631845	0,207395733	1,738586406
7	0,2056019	3,506596707	9,008676865	0,206914213	1,736171249
8	0,20592165	3,502432262	9,003854788	0,206165162	1,737993441
9	0,205508362	3,524851489	9,007659496	0,207247065	1,736290663
10	0,206309303	3,520824322	9,001075438	0,207065772	1,736871902

Deney 14:

Sürü büyüklüğü 40'a eşit ve yineleme sayısı 250'ye eşit olduğu 10 deneme sonuçları aşağıdaki Tablo EK-1.14'te sunulmuştur.

Tablo EK-1.14. On dördüncü deneyin sonuçları

Deneme	X_1	X_2	X_3	X_4	Maliyet
1	0,205095	3,506822	9,006699	0,2073009	1,73598445
2	0,206418	3,501392	9,011209	0,206485	1,7379288
3	0,206454	3,517555	9,005876	0,207124	1,73623566
4	0,205942	3,512614	9,004452	0,2073952	1,737943
5	0,205834	3,521757	9,006504	0,2068138	1,7360849
6	0,20671	3,509608	9,010275	0,2070072	1,73710375
7	0,205946	3,501874	9,005209	0,2069008	1,73761871
8	0,205714	3,514114	9,00861	0,2062399	1,73583607
9	0,206497	3,523869	9,001286	0,2066688	1,73840083
10	0,205733	3,505348	9,009833	0,2071317	1,73846027

Deney 15:

Sürü büyüklüğü 50'ye eşit ve yineleme sayısı 250'ye eşit olduğu 10 deneme sonuçları aşağıdaki Tablo EK-1.15'te sunulmuştur.

Tablo EK-1.15. On beşinci deneyin sonuçları

Deneme	X_1	X_2	X_3	X_4	Maliyet
1	0,206092461	3,504757472	9,005134821	0,206076663	1,737535307
2	0,205574099	3,51582449	9,000806466	0,206103313	1,736547631
3	0,206129617	3,510905005	9,003741357	0,207432542	1,735748225
4	0,206435717	3,516662733	9,002965969	0,207136983	1,736606262
5	0,206371564	3,503112079	9,000485562	0,207383892	1,737966814
6	0,206230298	3,515266175	9,005854183	0,207362813	1,736867182
7	0,206054346	3,515942271	9,007453066	0,207469497	1,737691749
8	0,206693753	3,522666631	9,010071637	0,206106488	1,73848588
9	0,206741959	3,520045806	9,008207926	0,206850758	1,738609206
10	0,206081483	3,509035329	9,000361861	0,207388123	1,735209408

Deney 16:

Sürü büyüklüğü 10'a eşit ve yineleme sayısı 300'e eşit olduğu 10 deneme sonuçları aşağıdaki Tablo EK-1.16'da sunulmuştur.

Tablo EK-1.16. On altıncı deneyin sonuçları

Deneme	X_1	X_2	X_3	X_4	Maliyet
1	0,204933662	3,478846433	9,011947339	0,206656906	1,735509841
2	0,206706656	3,50632751	9,005522877	0,20707967	1,738177131
3	0,206225453	3,519957516	9,000828901	0,206039647	1,737393124
4	0,206307259	3,519085062	9,007681782	0,207271979	1,738245816
5	0,206288997	3,519306979	9,001583859	0,206232774	1,736992352
6	0,206163088	3,505068618	9,008700109	0,206490221	1,735618174
7	0,20618315	3,522153176	9,008285496	0,206114265	1,738162349
8	0,205701117	3,513490764	9,009075381	0,206091621	1,736535576
9	0,205895467	3,503490556	9,000117271	0,206154995	1,736138886
10	0,205726726	3,523464757	9,012136553	0,206455058	1,737044015

Deney 17:

Sürü büyüklüğü 20'ye eşit ve yinleme sayısı 300'e eşit olduğu 10 deneme sonuçları aşağıdaki Tablo EK-1.17'de sunulmuştur.

Tablo EK-1.17. On yedinci deneyin sonuçları

Deneme	X ₁	X ₂	X ₃	X ₄	Maliyet
1	0,20617963	3,499535536	9,007582369	0,207272014	1,73577409
2	0,206125418	3,508146515	9,00407645	0,206772408	1,736039039
3	0,206384033	3,523405705	9,010613507	0,206461346	1,736048848
4	0,206552626	3,521823551	9,010639995	0,207480915	1,738823205
5	0,205741339	3,502544125	9,011517692	0,207397266	1,736307217
6	0,206595094	3,519608967	9,007920019	0,206192382	1,73750206
7	0,206022721	3,499006032	9,008848	0,206232258	1,736626606
8	0,206385679	3,5118871	9,00565045	0,207395344	1,735608216
9	0,206481223	3,504048748	9,007404165	0,2067622	1,736368786
10	0,206238019	3,520717883	9,011550015	0,206936826	1,73612571

Deney 18:

Sürü büyüklüğü 30'a eşit ve yineleme sayısı 300'e eşit olduğu 10 deneme sonuçları aşağıdaki Tablo EK-1.18'de sunulmuştur.

Tablo EK-1.18. On sekizinci deneyin sonuçları

Deneme	X_1	X_2	X_3	X_4	Maliyet
1	0,204706304	3,483158245	9,028074984	0,206538803	1,738626854
2	0,206467171	3,51213679	9,001532814	0,207284842	1,738704334
3	0,205665488	3,524263037	9,00666006	0,207460465	1,73743389
4	0,205906828	3,50637321	9,010947902	0,206567221	1,737814078
5	0,205618621	3,506186836	9,007845659	0,207135425	1,738675498
6	0,205950274	3,509888867	9,004868021	0,207467676	1,737095645
7	0,205546912	3,521132398	9,006847414	0,207023158	1,737495254
8	0,205706177	3,518616563	9,009313234	0,20643554	1,735299728
9	0,206365273	3,503220653	9,005583472	0,207108769	1,73625605
10	0,205779052	3,512680842	9,004504446	0,207004245	1,736564679

Deney 19:

Sürü büyüklüğü 40'a eşit ve yineleme sayısı 300'e eşit olduğu 10 deneme sonuçları aşağıdaki Tablo EK-1.19'da sunulmuştur.

Tablo EK-1.19. On dokuzuncu deneyin sonuçları

Deneme	X_1	X_2	X_3	X_4	Maliyet
1	0,206129461	3,505881635	9,003122935	0,206390196	1,738199165
2	0,206129943	3,52072253	9,009973903	0,206909841	1,738854791
3	0,206188573	3,500445474	9,010977878	0,206779257	1,736334943
4	0,206144322	3,500150532	9,004792858	0,20712297	1,73536674
5	0,206737412	3,521405173	9,000284163	0,206367888	1,73887177
6	0,205731271	3,508902859	8,999878409	0,207400913	1,73712173
7	0,206237877	3,508119792	9,001585349	0,207450086	1,738975813
8	0,205595656	3,501717318	9,004452514	0,206411289	1,73708576
9	0,206432121	3,503508408	9,009633251	0,206711694	1,738207586
10	0,206005101	3,50290271	9,003456981	0,206633175	1,738077002

Deney 20:

Sürü büyüklüğü 50'ye eşit ve yinleme sayısı 300'e eşit olduğu 10 deneme sonuçları aşağıdaki Tablo EK-1.20'de sunulmuştur.

Tablo EK-1.20. Yirminci deneyin sonuçları

Deneme	X_1	X_2	X_3	X_4	Maliyet
1	0,204858368	3,481904626	9,027346833	0,207156087	1,738886308
2	0,206426715	3,505710042	9,008578996	0,207392733	1,736935101
3	0,20679048	3,51418117	9,004762623	0,207458575	1,736554636
4	0,206404106	3,507449299	9,003934745	0,206703879	1,737916318
5	0,206559396	3,500048643	9,012131557	0,206681536	1,737949601
6	0,206214009	3,513417839	8,999812285	0,206210045	1,735882336
7	0,206708201	3,523509417	9,012187341	0,207392533	1,737171429
8	0,205918538	3,507378642	9,005271833	0,206399768	1,738364801
9	0,205744415	3,524681626	9,000529909	0,206839756	1,737486645
10	0,20575528	3,511049876	9,009059848	0,206574106	1,735935089

Deney 21:

Sürü büyüklüğü 10'a eşit ve yineleme sayısı 500'e eşit olduğu 10 deneme sonuçları aşağıdaki Tablo EK-1.21'de sunulmuştur.

Tablo EK-1.21. Yirmi birinci deneyin sonuçları

Deneme	X_1	X_2	X_3	X_4	Maliyet
1	0,205157434	3,484637233	9,036606553	0,20584776	1,724304926
2	0,206261631	3,506892859	9,004620965	0,207225637	1,737110936
3	0,206029551	3,524949261	9,00813402	0,206246299	1,736017199
4	0,206704438	3,499469536	9,004215368	0,206145547	1,738771629
5	0,206541758	3,509415299	9,011734179	0,206633307	1,735966154
6	0,206188385	3,500787641	9,007210743	0,206371323	1,736391155
7	0,206488828	3,504119347	9,008262617	0,206473562	1,736950027
8	0,205573254	3,520023942	9,002445542	0,206939165	1,735817859
9	0,206416669	3,499483259	9,006225961	0,206257296	1,737610677
10	0,2063335	3,510278181	9,007405284	0,206063012	1,737874591

Deney 22:

Sürü büyüklüğü 20'ye eşit ve yineleme sayısı 500'e eşit olduğu 10 deneme sonuçları aşağıdaki Tablo EK-1.22'de sunulmuştur.

Tablo EK-1.22. Yirmi ikinci deneyin sonuçları

Deneme	X_1	X_2	X_3	X_4	Maliyet
1	0,205551371	3,499378638	9,000907955	0,207398111	1,736234702
2	0,205798784	3,513725698	9,00198157	0,207197114	1,736840626
3	0,206422247	3,499681093	9,003738785	0,206267428	1,737632286
4	0,206357566	3,500192041	9,000812565	0,206689001	1,735781447
5	0,206568656	3,500140521	9,008550831	0,207004958	1,735760048
6	0,20588216	3,503250388	9,00260454	0,206901539	1,735399284
7	0,205551715	3,500585695	9,003754629	0,207256242	1,736207436
8	0,205687942	3,505849326	9,006025493	0,206567086	1,736289228
9	0,206198044	3,502498686	9,00043546	0,207162998	1,737597936
10	0,206359677	3,524091611	9,010002341	0,20677953	1,73841072

Deney 23:

Sürü büyüklüğü 30'a eşit ve yineleme sayısı 500'e eşit olduğu 10 deneme sonuçları aşağıdaki Tablo EK-1.23'te sunulmuştur.

Tablo EK-1.23. Yirmi üçüncü deneyin sonuçları

Deneme	X_1	X_2	X_3	X_4	Maliyet
1	0,204628019	3,481428657	9,034725951	0,20611421	1,726060611
2	0,206106551	3,518058988	9,010283528	0,206030295	1,735704057
3	0,205893302	3,500936221	9,001563965	0,20607438	1,737722722
4	0,206795281	3,503582484	9,00391075	0,206120395	1,736259805
5	0,206582258	3,503913721	9,001346594	0,206718929	1,737547427
6	0,205614718	3,51651618	9,004372288	0,207388527	1,735729948
7	0,205954085	3,522555579	9,005054753	0,206957716	1,738736302
8	0,205979191	3,502764978	9,000340294	0,207366131	1,73520844
9	0,205623481	3,499557123	9,00471424	0,206527204	1,736428854
10	0,205813593	3,507669159	9,004358783	0,20611407	1,7369082

Deney 24:

Sürü büyüklüğü 40'a eşit ve yineleme sayısı 500'e eşit olduğu 10 deneme sonuçları aşağıdaki Tablo EK-1.24'te sunulmuştur.

Tablo EK-1.24. Yirmi dördüncü deneyin sonuçları

Deneme	X ₁	X ₂	X ₃	X ₄	Maliyet
1	0,206391365	3,505873753	9,009337701	0,206690179	1,735215167
2	0,205788126	3,504762722	8,999801866	0,206064734	1,738538053
3	0,205987882	3,510028612	9,002761054	0,206316263	1,735651325
4	0,206384819	3,520708441	9,005283211	0,206453108	1,735641921
5	0,205500609	3,513189616	9,011414725	0,206037073	1,736041405
6	0,205579284	3,508330013	9,001878352	0,20616572	1,7373699
7	0,20639883	3,51677607	9,002542709	0,207349332	1,737707847
8	0,206451791	3,51981408	9,003655182	0,207128215	1,737676675
9	0,205893943	3,510131095	9,003664015	0,207060488	1,737998873
10	0,20559895	3,519586293	9,002546553	0,206317192	1,73820087

Deney 25:

Sürü büyüklüğü 50'ye eşit ve yineleme sayısı 500'e eşit olduğu 10 deneme sonuçları aşağıdaki Tablo EK-1.25'te sunulmuştur.

Tablo EK-1.25. Yirmi beşinci deneyin sonuçları

Deneme	X_1	X_2	X_3	X_4	Maliyet
1.	0,204987614	3,479413387	9,021729763	0,206037127	1,724685269
2.	0,206102004	3,504106829	9,011149859	0,207374507	1,737159204
3.	0,205975856	3,500417116	9,021853723	0,206134848	1,736588331
4.	0,205589183	3,519442874	9,026975869	0,207203959	1,737250748
5.	0,20640298	3,515783012	9,016090244	0,206372761	1,737940813
6.	0,206367957	3,516038883	9,011924722	0,206680263	1,736126796
7.	0,206343668	3,505898991	9,024374533	0,20743182	1,73664772
8.	0,206750489	3,524597915	9,018729465	0,206573653	1,738961836
9.	0,206547233	3,524420186	9,00574755	0,206358777	1,737732934
10.	0,206343661	3,506477598	9,032623833	0,206641183	1,738366035

ÖZGEÇMİŞ

Adı Soyadı : Shahad Odah Feneaker FENEAKER
Doğum Yeri ve Yılı : Irak / Baghdad – 13.02.1988
Medeni Hali : Evli
Yabancı Dili : Arapça ve İngilizce
E-posta : shahadsw88@gmail.com
Mesleği : Bağdat Diyanet İşleri Müdürlüğü – Bilgisayar Mühendisi



Eğitim Durumu

Lise : Alkhadraa Kız Lisesi, 2007
Lisans : Alrafidain Kolej – Bilgisayar Mühendisliği Bölümü, 2011