T.C.
DÜZCE ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ADAPTİF KISITLAMA FAKTÖRLÜ PARÇACIK SÜRÜ OPTİMİZASYONU

ERDİ YALÇIN

YÜKSEK LİSANS TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

DANIŞMAN
DOÇ. DR. PAKİZE ERDOĞMUŞ

DÜZCE, 2016

T.C.
DÜZCE ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ADAPTİF KISITLAMA FAKTÖRLÜ PARÇACIK SÜRÜ OPTİMİZASYONU

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

Tez Danışmanı

Doç. Dr. Pakize ERDOĞMUŞ

Düzce Üniversitesi

Jüri Üyeleri

Doç. Dr. Pakize ERDOĞMUŞ
Düzce Üniversitesi ______________________

Doç. Dr. Resul KARA
Düzce Üniversitesi ______________________

Yrd. Doç. Dr. Okan ERKAYMAZ
Bülent Ecevit Üniversitesi ______________________

Tez Savunma Tarihi: 19/12/2016

BEYAN

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

19 Aralık 2016

Erdi YALÇIN

TEŞEKKÜR

Yüksek lisans öğrenimimde ve bu tezin hazırlanmasında gösterdiği her türlü destek ve yardımlarından dolayı çok değerli hocam Doç. Dr. Pakize ERDOĞMUŞ'a en içten dileklerimle teşekkür ederim.

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

19 Aralık 2016 **Erdi YALÇIN**

İÇİNDEKİLER

ŞEKİL LİSTESİ VI
ÇİZELGE LİSTESİ VII
KISALTMALAR VIII
SİMGELER IX
ÖZET X
ABSTRACT XI
1. GİRİŞ 1
2. GENEL KAVRAMLAR 3
2.1 OPTIMIZASYON 3
2.2 SEZGISEL OPTIMIZASYON 5
2.3 METASEZGISEL ALGORITMALAR 6
3. PARÇACIK SÜRÜ OPTİMİZASYONU 8
3.1 GELENEKSEL PARÇACIK SÜRÜ OPTIMIZASYONU 8
3.2 ATALET DEĞERLI PSO 11
3.3 ADAPTIF KISITLAMALI PSO 12
4. ÖNERİLEN FORMÜLASYONLAR VE UYGULANMASI 13
4.1 TAVLAMA BENZETIMI 13
4.1.1 Aritmetik Azalma 14
4.1.2 Geometrik Azalma 14
4.1.3 Ters Fonksiyonel Azalma 15
4.2 DENEYSEL ÇALIŞMALAR 15
5. SONUÇLAR VE ÖNERİLER 28
KAYNAKÇA 29
ÖZGEÇMİŞ 31

ŞEKİL LİSTESİ

Sayfa No

Şekil 2.1 - Bir Tasarımın Optimizasyon Şeması 3
Şekil 2.2 - Optimizasyon Problemlerinin Sınıflandırılması [27, Kaynak 56] 4
Şekil 2.3 - Optimizasyon Algoritmalarının Sınıflandırılması [27, Kaynak 56] 5
Şekil 2.4 - Metasezgisel Algoritmaların Cinslerine Göre Sınıflandırılması [15] 7
Şekil 3.1 - Parçacık Sürü Optimizasyonunun Akış Diyagramı 10
Şekil 4.1 - Aritmetik Azalma Grafiği 14
Şekil 4.2 – Geometrik Azalma Grafiği 14
Şekil 4.3 – Ters Fonksiyonel Azalma Grafiği 15
Şekil 4.4 - Ackley Fonksiyonunda w, Atalet Ağırlığının ve K, Adaptif Kısıtlama Faktörünün İterasyon Sayısıyla Azalmasının Sonuca Etkisi 18
Şekil 4.5 - Beale Fonksiyonunda w, Atalet Ağırlığının ve K, Adaptif Kısıtlama Faktörünün İterasyon Sayısıyla Azalmasının Sonuca Etkisi 18
Şekil 4.6 - Booth Fonksiyonunda w, Atalet Ağırlığının ve K, Adaptif Kısıtlama Faktörünün İterasyon Sayısıyla Azalmasının Sonuca Etkisi 19
Şekil 4.7 - Colville Fonksiyonunda w, Atalet Ağırlığının ve K, Adaptif Kısıtlama Faktörünün İterasyon Sayısıyla Azalmasının Sonuca Etkisi 19
Şekil 4.8 – Dixon & Price Fonksiyonunda w, Atalet Ağırlığının ve K, Adaptif Kısıtlama Faktörünün İterasyon Sayısıyla Azalmasının Sonuca Etkisi 20
Şekil 4.9 - Griewank Fonksiyonunda w, Atalet Ağırlığının ve K, Adaptif Kısıtlama Faktörünün İterasyon Sayısıyla Azalmasının Sonuca Etkisi 20
Şekil 4.10 - Hump Fonksiyonunda w, Atalet Ağırlığının ve K, Adaptif Kısıtlama Faktörünün İterasyon Sayısıyla Azalmasının Sonuca Etkisi 21
Şekil 4.11 - Levy Fonksiyonunda w, Atalet Ağırlığının ve K, Adaptif Kısıtlama Faktörünün İterasyon Sayısıyla Azalmasının Sonuca Etkisi 21
Şekil 4.12 - Matyas Fonksiyonunda w, Atalet Ağırlığının ve K, Adaptif Kısıtlama Faktörünün İterasyon Sayısıyla Azalmasının Sonuca Etkisi 22
Şekil 4.13 - Perm Fonksiyonunda w, Atalet Ağırlığının ve K, Adaptif Kısıtlama Faktörünün İterasyon Sayısıyla Azalmasının Sonuca Etkisi 22
Şekil 4.14 - Rastrigin Fonksiyonunda w, Atalet Ağırlığının ve K, Adaptif Kısıtlama Faktörünün İterasyon Sayısıyla Azalmasının Sonuca Etkisi 23
Şekil 4.15 - Rosenbrock Fonksiyonunda w, Atalet Ağırlığının ve K, Adaptif Kısıtlama Faktörünün İterasyon Sayısıyla Azalmasının Sonuca Etkisi 23
Şekil 4.16 - Schwefel Fonksiyonunda w, Atalet Ağırlığının ve K, Adaptif Kısıtlama Faktörünün İterasyon Sayısıyla Azalmasının Sonuca Etkisi 24
Şekil 4.17 - Sphere Fonksiyonunda w, Atalet Ağırlığının ve K, Adaptif Kısıtlama Faktörünün İterasyon Sayısıyla Azalmasının Sonuca Etkisi 24
Şekil 4.18 - Sum Squares Fonksiyonunda w, Atalet Ağırlığının ve K, Adaptif Kısıtlama Faktörünün İterasyon Sayısıyla Azalmasının Sonuca Etkisi 25
Şekil 4.19 - Zakharov Fonksiyonunda w, Atalet Ağırlığının ve K, Adaptif Kısıtlama Faktörünün İterasyon Sayısıyla Azalmasının Sonuca Etkisi 25

ÇİZELGE LİSTESİ

Sayfa No

Çizelge 4.1 - Uygulamada kullanılan karşılaştırma fonksiyonları 16
Çizelge 4.2 – Adaptif Kısıt Parametreli (K) PSO'nun Fonksiyonlar Üzerindeki Test Sonuçları 26
Çizelge 4.3 - Atalet Ağırlıklı (w) PSO'nun Fonksiyonlar Üzerindeki Test Sonuçları 27

KISALTMALAR

ABC	Artificial Bee Colony
BA	Bat Algorithm
BT	Benzetim Tavlama
COA	Cuckoo Optimization Algorithm
FO	Firefly Optimization
GA	Genetik Algoritma
IWPSO	Inertial Weight PSO
KKO	Karınca Kolonisi Optimizasyonu
PSO	Parçacık Sürü Optimizasyonu
TB	Tavlama Benzetimi
YAK	Yapay Arı Kolonisi

SİMGELER

c_1, c_2	Sosyal ve Bilişsel Katsayılar
$gbest, p_{gd}$	Global Optimum Değer
j_{max}	Toplam İterasyon Sayısı
k	İterasyon Sayısı
K	Kısıtlama Faktörü
$pbest, p_{id}$	Lokal Optimum Değer
v_i^{k+1}	Bir Sonraki İterasyon Hızı
v_i^k	i. Parçacığın k. İterasyondaki Hızı
w	Atalet Ağırlığı
x_i^k	i. Parçacığın k. İterasyondaki Konumu
x_i^{k+1}	Bir Sonraki İterasyon Konumu

ÖZET

ADAPTİF KISITLAMA FAKTÖRLÜ PARÇACIK SÜRÜ OPTİMİZASYONU

Erdi YALÇIN

Düzce Üniversitesi

Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Anabilim Dalı

Yüksek Lisans Tezi

Danışman: Doç. Dr. Pakize ERDOĞMUŞ

Aralık 2016, 42 sayfa

Optimizasyon son yıllarda çok çeşitli alanlarda kullanılan bir teknik olmuştur. Bilgisayar ağlarında, ekonomide, görüntü işlemede, robotikte ve daha birçok alanda kullanılmıştır. Kuş ve balık sürülerinden esinlenilerek oluşturulmuş olan Parçacık Sürü Optimizasyonu (PSO) yöntemi hızlı yakınsayan bir algoritmadır.

Bu çalışmada, PSO algoritmalarında genellikle statik olarak yer alan atalet ağırlığı (w) ve adaptif kısıt parametresi (K), benzetim tavlamadaki sıcaklık parametresinden esinlenilerek fonksiyonel şekilde (aritmetik, geometrik ve ters fonksiyon) azaltılarak karşılaştırma (benchmark) fonksiyonları üzerinde algoritmanın performansı incelenmiştir. Sabit parametrelerle elde edilen sonuçlar, uyguladığımız azalan yöntemlerle karşılaştırılmış ve kısıtsız optimizasyon test problemleri için ters fonksiyonun sabit parametreye göre daha başarılı sonuçlar verdiği görülmüştür.

Sonuç grafiklerinde de görüleceği üzere, atalet ağırlığı (w) ve adaptif kısıt parametresini (K) kullanan PSO'larda azalan fonksiyonlar ile yapılan testler sabit değer ile yapılan testlere göre daha iyi sonuçlar verdiği gözlemlenmiştir. Gerçek zamanlı uygulamalar için geliştirilen PSO algoritmalarında, parametrelerin sabit kullanımı yerine parametrelerin fonksiyonel şekilde azaltılarak kullanılabileceği önerilmektedir.

Anahtar sözcükler: Parçacık Sürü Optimizasyonu, Sezgisel Araştırma, Kısıtsız Optimizasyon, Kısıtlama Faktörü, Atalet Ağırlığı

ABSTRACT

PARTICLE SWARM OPTIMIZATION WITH ADAPTIVE RESTRICTION FACTOR

Erdi YALÇIN
Düzce University
Graduate School of Natural and Applied Sciences, Department of Computer Engineering
Master Thesis
Supervisor: Ass. Prof. Dr. Pakize ERDOĞMUŞ
December 2016, 42 pages

Optimization has been a technique that has been used in a wide variety of areas in recent years. Computer networks, economics, image processing, robotics and many other areas are used. The Particle Swarm Optimization (PSO) method which is developed from standard PSO, inspired by bird and fish swarms, is a fast converging algorithm. In this study, K and w ,generally used as static, are decreased with as aritmethic, geometric and inversely inspired by temperature parameter in Simulated Annealing and the performances of the algorithms are observed on benchmark function.

The results that obtained with constant parameters, compared with reducing methods that applied and it has been shown that the inverse function gives more successful results than the fixed parameter for unconstrained optimization test problems.

As can be seen in the result graphs, tests made with functions decreasing in PSOs using inertia weight (w) and adaptive constraint parameter (K) yielded better results than tests with constant values. In PSO algorithms developed for real-time applications, it is suggested that parameters can be used reducing functionally instead of constant use of parameters.

Keywords: Particle Swarm Optimization, Heuristic Search, Unconstrained Optimization, Adaptive Restriction Parameter, Inertial Weight

1. GİRİŞ

Optimizasyon günümüzde hemen hemen her alanda kullanılmış ve kullanılmaya devam edilen bir yöntemdir. Kısaca tanımlanacak olursa, değişkenlere göre değişen bir amaç fonksiyonunun verilen kısıtlayıcı şart fonksiyonları altında en uygun değerlerinin bulunması işlemidir. Tasarım mühendisliğinden, elektrik mühendisliğine, ekonomiden, bilgisayar ağlarına, robotikten, görüntü işlemeye kadar birçok kullanım alanı mevcuttur. Bir makine tasarımında en uygun makine boyutları bulunması, bir firmanın kâr gelirinin arttırılması veya işletmede üretilen bir ürünün imkanlar dahilinde en az maliyetle üretimi, sensör yerleşiminin en geniş çekim alanı oluşturacak şekilde yapılması, robotun engellere takılmadan en kısa yolu takip ederek hedefe ulaşması optimizasyon problemlerindendir. Geleneksel yöntemler bu problemleri çözmede yeterli bulunmamış ve yeni çözüm yöntemleri aranmaya başlanmıştır. 1970'li yılların ortasına doğru, sezgisel yöntemler üzerine çalışmalar yapılmıştır.

Zor ve karmaşık optimizasyon problemini çözmek için geliştirilen sezgisel algoritmalar giderek artmaktadır. Klasik yöntemlerin gerçek hayat problemlerini çözmekteki zorluğu hatta doğru bir şekilde çözüm üretememeleri sezgisel algoritmalara olan ilgiyi arttırmıştır[1]. Sezgisel yöntemler tabiattan topluma, kültürden politikaya ve insana kadar çevremizde gördüğümüz hemen her şeyin modellenmesi ile oluşturulmuştur[2]. Günümüzde biyolojik sistemlerden esinlenilerek geliştirilmiş bir çok sezgisel yöntem, optimizasyon problemlerinin çözümünde kullanılmaktadır. Sezgisel yöntemlerden ilki, 1976'ta Holland[3] tarafından geliştirlen Genetik Algoritmadır(GA). Evrim teorisinden ortaya çıkarılan GA, sezgisel algoritmalardan üzerinde çalışılanıdır. GA, uzay mühendisliği, astronomi ve astrofizik, jeofizik, finans, malzeme mühendisliği gibi daha pek çok alanda uygulanmış bir algoritmadır[4]. Biyolojik sistemlerde, bireyin çevreyle ve diğer bireylerle olan etkileşiminin ve ortak davranışlarının incelenerek modellenmesi ile sürü zekâsına dayalı birçok algoritma geliştirilmiştir. 1990'lı yılların başında M. Dorigo ve arkadaşları tarafından geliştirilen Karınca Kolonisi Optimizasyonu (KKO) (Ant Colony Optimization (ACO)) [5] karıncaların sürü zekâsını, özellikle ayrık optimizasyon problemlerinin çözümüne uyarlamış bir algoritmadır.

PSO (Particle Swarm Optimization) kuş ve balık sürülerinin davranışlarından esinlenilerek ortaya çıkarılan bir optimizasyon yöntemidir. 1995 yılında J. Kennedy ve R. C. Eberhart tarafından geliştirilmiştir [6]. PSO, fonksiyon optimizasyonu, yapay sinir ağları eğitimi, otomatik kontrolde parametrelerin ayarlanması gibi birçok optimizasyon problemini başarı ile çözmüştür[7]. Sürü zekâsına dayalı olarak geliştirilen bir başka algoritma ise Yapay Arı Kolonisi (YAK) (Artificial Bee Colony (ABC)) optimizasyon algoritmasıdır. 2005 yılında Karaboğa tarafından tanıtılmıştır[8]. Guguk kuşu algoritması (Cuckoo Optimization Algorithm (COA)) guguk kuşlarının kuluçka parazit türlerini temel alarak 2011 yılında geliştirilmiş bir global optimizasyon algoritmasıdır[9]. Ateş böceği optimizasyonu (Firefly Optimization (FO) ve Yarasa Algoritması (Bat Algorithm (BA)) da sırası ile 2009 ve 2010 yıllarında geliştirilen sezgisel optimizasyon algoritmalarına örnektir [28].

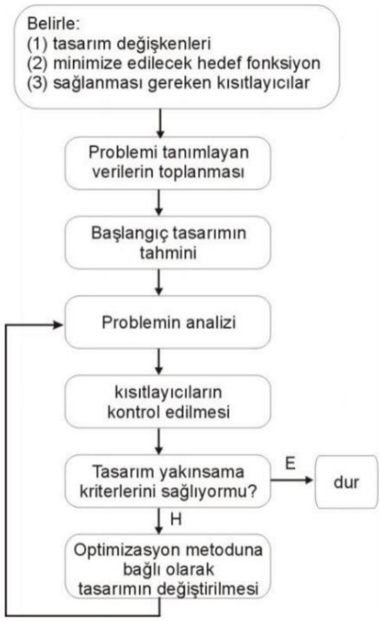
Bu tez beş ana bölümden oluşmaktadır.

Bu çalışmanın birinci bölümünde konuya giriş yapılmıştır. Kısaca neler yapılacağı ve amaçlar açıklanmıştır. İkinci bölümde; tez çalışmasında kullanılan temel kavramlar (optimizasyon, optimizasyon algoritmaları, sezgisel optimizasyon, metasezgisel algoritmalar) hakkında bilgi verilmiştir. Üçüncü bölümde; PSO hakkında bilgiler verilmiştir. Dördüncü bölümde; adaptif kısıtlama faktöründeki ve atalet ağırlığındaki değişimin, PSO'ya etkisini ölçmek için geliştirilen yöntemler açıklanmış ve deneysel sonuçlarına yer verilmiştir. Beşinci bölümde; sonuçlar ve öneriler ele alınarak, tez çalışması hakkında genel değerlendirme yapılmıştır. Ayrıca gelecekte optimizasyon alanında yapılacak yeni çalışmalar için öneriler de sunulmuştur.

2. GENEL KAVRAMLAR

2.1 Optimizasyon

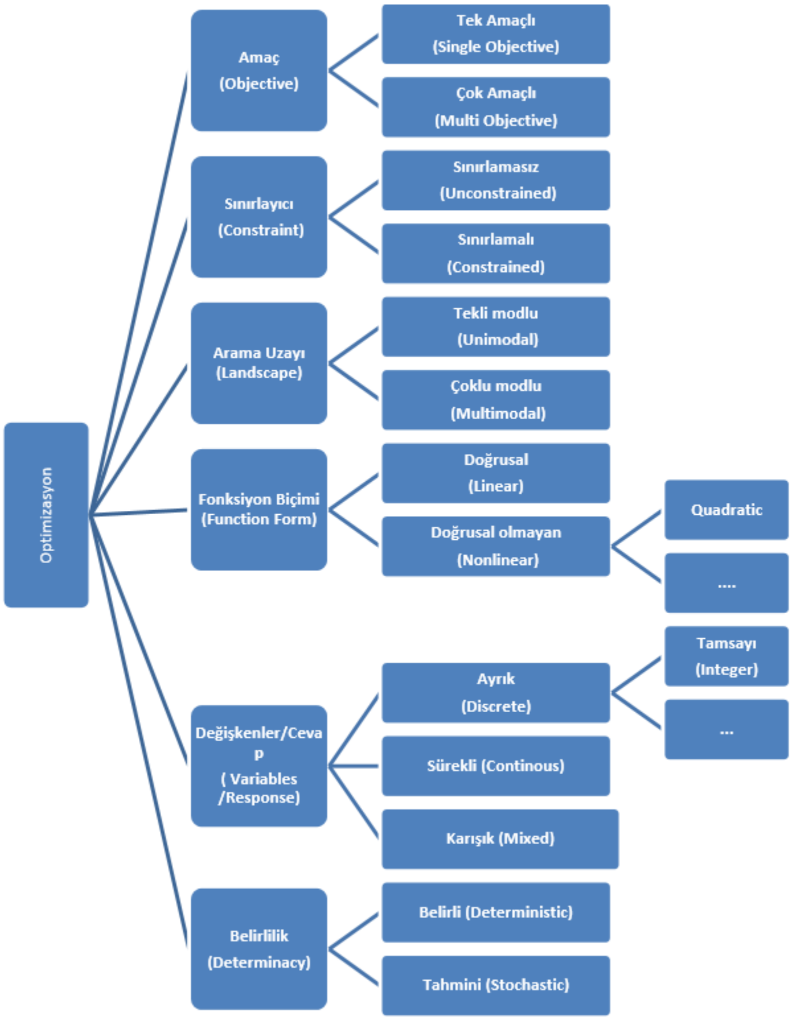
Optimizasyon, bir sistemin tasarlanmasında olası tüm çözümlerin arasından en iyisinin bulunması olarak ifade edilebilir. Belirli kısıtları olan, bir problemin sonucunu etkileyen parametre değerlerinin bulunarak en kârlı sonucun minimum maliyetlerle belirlenmesini hedeflemek, problemin optimize edilmesi anlamını taşır. Her bir gerçek dünya probleminde gerekli çaba, sermaye, malzeme ve işçiliğin minimum seviyede belirlenmesi ve kazancın maksimum düzeyde olması en gerçekçi amaç olmuştur. Optimizasyon işleminde problemin çözümünü belirleyen karar değişkenlerinin belirlenmesi, sonrasında ise bu karar verici parametreler ışığında minimize edilecek maliyet fonksiyonu ya da maksimize edilecek kâr fonksiyonları tanımlanmalıdır (amaç fonksiyonu). Bunların tanımlanmasında problemi sınırlayan, karar değişkenlerinin alabileceği değer ya da değer aralıklarını ifade eden sınırlamaların belirtilmesi gerekmektedir. Probleme göre bazı kısıtlamalar eşitsizlik, bazıları ise eşitlikler şeklinde olabilmektedir (kısıtlayıcılar). Şekil 2.1'de bir sistem tasarlanırken optimizasyona ait yukarıda açıklanan adımlar şematize edilmiştir. [27]



Şekil 2.1 - Bir Tasarımın Optimizasyon Şeması

Optimizasyon, geniş bir yelpazede ele alınacak problemlerin kesin optimum sonuçlarının araştırılması amaçlarını içermektedir. Bu yüzden optimizasyon problemlerinin çok farklı isimlendirmeleri ve sınıflandırmaları mevcuttur. Optimizasyon teknikleri problemden probleme önemli ölçüde değişebilmektedir. Her bir optimizasyon probleminin çözümü için tek bir yaklaşım söz konusu olamamaktadır. Her bir problemin karmaşıklığı kısıtlayıcılara ve amaç fonksiyonlarına önemli oranda bağlı olduğundan çok farklılık gösterebilmektedir.

Yang, optimizasyon problemlerini Şekil 2.2' de olduğu gibi sınıflandırmıştır.



Şekil 2.2 - Optimizasyon Problemlerinin Sınıflandırılması [27, Kaynak 56]

Optimizasyon problemlerinin çözümünde kullanılan algoritmalar iki kategoride değerlendirilebilir: Bunlardan ilki belirli bir prosedürü takip eden, takip edilen yolun, tasarım değişkenleri ve fonksiyon değerlerinin tekrarlanabildiği deterministik (rastgele olmayan, belirli) algoritmalardır. Yani algoritma ne zaman çalıştırılırsa çalıştırılsın aynı girdiler için aynı sonuçları her zaman üretebilen algoritmalar deterministik yaklaşımla açıklanmaktadır. Diğeri ise rastsallığı içinde barındıran stokastik (olasılıksal) algoritmalardır. Ayrıca, deterministik ve stokastik algoritmaların bir arada kullanıldığı hibrit yaklaşımlar da üçüncü bir tür optimizasyon algoritması olarak ele alınmaktadır.

Optimizasyon problemlerinin çözümünde genellikle melez yaklaşımlar tercih edilmektedir. Şekil 2.3'te optimizasyon problemlerinin çözümünde kullanılan algoritmaların sınıflandırılması gösterilmektedir.

Şekil 2.3 - Optimizasyon Algoritmalarının Sınıflandırılması [27, Kaynak 56]

2.2 Sezgisel Optimizasyon

Sezgisel optimizasyon, herhangi bir problemin optimizasyonunda alternatif çözüm yollarından en optimum sonuçlara ulaşabilmek için doğal fenomenlerden ilham alan optimizasyon teknikleridir. Sezgisel optimizasyon çözümünde kullanılan algoritmalar, yakınsama özelliğine sahip olup, her zaman optimum çözümü garanti edememektedirler. Yakınsama özelliğinden dolayı optimum sonuca yakın sonuçlar üretebilmektedirler.

Sezgisel algoritmaların değerlendirilmesinde kullanılan kriterler aşağıdaki gibi olmalıdır [13]:

Esneklik: Algoritmalar modelde, sınırlamalarda ve amaç fonksiyonlarında yapılacak değişiklikleri kolayca karşılayabilmelidir.

Algoritma Basitliği ve Gerçeklenebilirlik: Algoritma prensipleri basit olmalı ve genel olarak uygulanabilir olmalıdır. Bu durum problem yapısı ile ilgili başlangıçta çok az bilgiye sahip olunması halinde bile algoritmanın yeni alanlara kolaylıkla uygulanabilmesini sağlar.

Çözüm Kalitesi ve Hesaplama Zamanı: Çözüm kalitesi ve hesaplama zamanı bir algoritmanın etkinliğinin değerlendirilmesi için önemli kriterlerdendir. Bundan dolayı bir algoritma ayarlanabilir parametreler setine sahip olmalı ve bu parametreler kullanıcıya etkinlik açısında hesaplama maliyeti ile çözüm kalitesi arasında bir vurgulamanın yapılabilmesine imkân vermelidir. Diğer bir deyişle çözüm kalitesi ile hesap zamanı arasındaki ilişki kontrol edilebilmelidir.

Etkileşimli Hesaplama ve Teknoloji Değişimleri: Algoritma içinde insan-makine etkileşimini kullanma fikri çoğu sistemde yaygın olarak gerçekleştirilmektedir. Herkesçe bilindiği gibi iyi bir kullanıcı arayüzü herhangi bir bilgisayar sistemini veya algoritmayı daha çekici yapmaktadır. Bunun en önemli avantajı çözümlerin grafiksel olarak sergilenebilmesidir.

Dinçlik: Yöntem başlangıç çözümünün seçimine sahip olmaksızın her zaman yüksek kaliteli, kabul edilebilir çözümleri üretebilme kabiliyetine sahip olmalıdır.

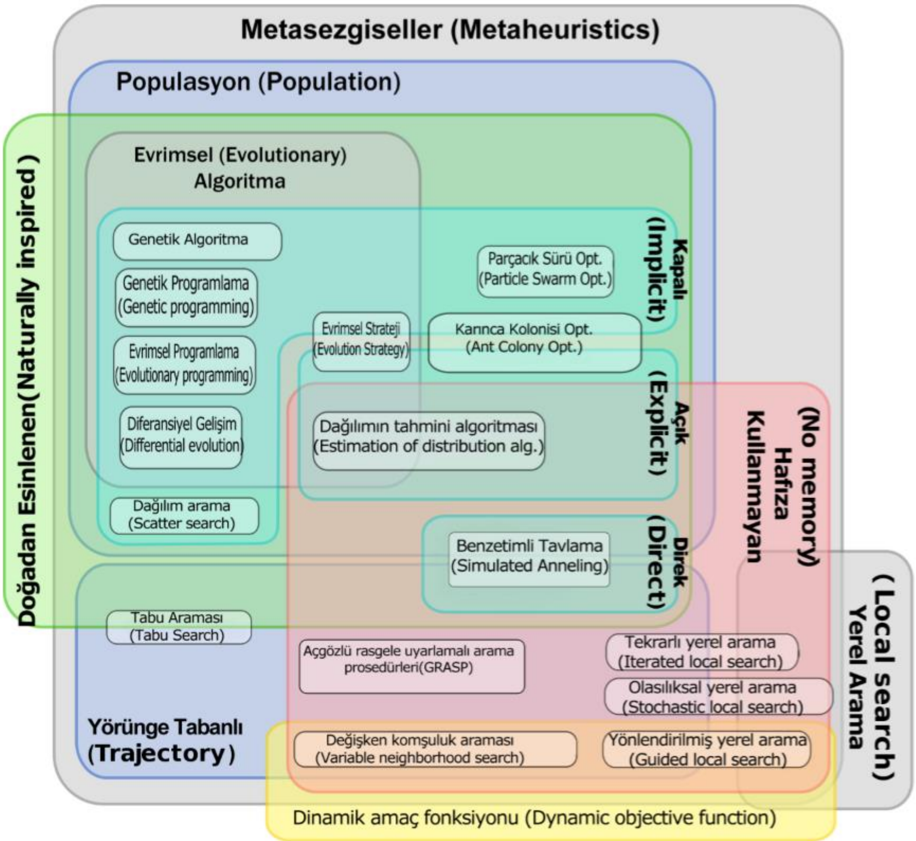
Basitlik ve Analiz Edilebilirlik: Karmaşık algoritmalar, esneklik ve çözüm kalitesi açısından basit algoritmalardan daha zor analiz edilmektedir. Algoritma kolayca analiz edilebilir olmalıdır.

2.3 Metasezgisel Algoritmalar

Metasezgisel algoritmalar, sezgisel optimizasyon algoritmalarının gelişmişi olarak ifade edilir. Temel sezgisel yöntemlerin bir arada kullanılmasıyla ortaya çıkmışlardır. Yüksek seviye anlamında kullanılan “meta” ifadesi ile bu algoritmalar basit sezgisel algoritmalardan daha iyi performans sergilemektedirler. Ayrıca tüm metasezgisel algoritmalar, rastgelelik ve yerel aramayı değişimli olarak kullanmaktadırlar.

Rastgelelik, optimizasyon algoritmalarının çözümlerini yerel aramalardan kurtararak küresel ölçekli aramalar yaparak iyileştirmeyi sağlamaktadır [12]. Şekil 2.4’ te metasezgisel algoritmaların sınıflandırılması gösterilmektedir.

Pek çok sezgisel algoritma probleme bağımlıdır. Bir problem için en iyi değeri gösterirken başka bir problem için aynı şekilde başarılı olmayabilir. Metasezgisellerin tüm problemleri kapsayan yeni yöntemler geliştirmek için çalışmalar yapılmaktadır[14].



Şekil 2.4 - Metasezgisel Algoritmaların Cinslerine Göre Sınıflandırılması [15]

Metasezgisel algoritmaların karakteristik özellikleri aşağıdaki gibi sıralanabilir [16]:

- Metasezgisel algoritmalar arama süreçlerine rehberlik eden stratejilerdir.
- Arama uzayındaki keşiflerle en iyi ya da buna en yakın çözümlerin bulunması amacını taşır.
- Metasezgisel algoritmaları oluşturan teknikler, basit yerel arama prosedüründen karmaşık öğrenim süreçlerine kadar olan aralıkta bir değişim gösterir.
- Metasezgisel algoritmalar tahmine dayalı rastgelelik içerirler.
- Bulunan bir çözüme takılıp kalmayı önleyici mekanizmalar barındırırlar.
- Metasezgisel algoritmalar probleme özgü değildir.
- Farklı yöntemlerin kullanılmasıyla arama uzayını keşfeden yüksek seviyeli stratejilerdir.

Bu tez çalışması kapsamında ele alınan PSO optimizasyon algoritması Şekil 2.4'te de görüldüğü gibi, doğadan esinlenilen - popülasyon tabanlı metasezgisel algoritmalar sınıfında değerlendirilmektedir.

3. PARÇACIK SÜRÜ OPTİMİZASYONU

3.1 Geleneksel Parçacık Sürü Optimizasyonu

PSO, kuş sürüleri, balıklar ve pek çok sosyal hayvanın davranışları gözlemlenerek esinlenilen, optimizasyon problemleri için öne sürülmüş bir algoritmadır. Bu sosyal hayvanlar grubu gözlemlendiğinde bu hayvanların yiyecek ararken etkileşim içerisindedirler. Sürü içerisindeki bir hayvanın yiyecek bulması ile diğer hayvanların da sürüden kopmadan konumlarını yiyeceğin olduğu yöne doğru yöneldikleri gözlemlenmiştir. Ayrıca hızlarını da bu etkileşime göre güncelledikleri gözlemlenmiştir. Hayvanlar arasındaki bu sosyal etkileşim araştırmacılar tarafından PSO ile modellenmiştir [28].

PSO, Reeves'in 1993 yılında Lucas film'de çalışması sırasında yaptığı bir çalışma ile ortaya çıkmıştır. Çalışma, bulut veya patlama gibi bulanık bir objenin oluşturulmasını gerçekleştirmek için birlikte çalışan çok sayıda bireyi kullanan bir parçacık sistemidir [17].

Reynolds [18], Reeves'in tanımlamış olduğu parçacık sistemine iç-cisim iletişimi ve oryantasyonu ilave etmiştir. Bu şekilde parçacıklar temel sürü kurallarını uygulayabilmiştir. PSO sezgisel bir yöntemdir. Doğadaki çoğu hayvan sürü davranışlarının eklenmesiyle gelişmiş bir algoritma olmuştur.

PSO, Heppener ve Grenarde [19]'ın 1990 yılında, kuş sürülerinin mısır arayışların benzeşmesini içeren bir çalışma yapmıştır. Bu çalışmadan etkilenen Kennedy ve Eberhart'ın [20] geliştirdiği geleneksel PSO olarak adlandırdığımız güçlü bir optimizasyon olarak ortaya çıkmıştır [6].

PSO, Genetik Algoritma gibi evrimsel bir algoritmadır. Genetik Algoritma'da çözüm uzayındaki o anki çözümlere popülasyon adı verilirken, PSO'da sürü (swarm) adı verilmektedir. PSO, mutasyon ve çaprazlama gibi operatörleri olmadığından GA'ya göre daha basittir ve bu şekilde en iyi çözüme daha hızlı bir şekilde yakınsar. Sürüyü oluşturan her bir kuş bir particle (parçacık) şeklinde tanımlanmaktadır. PSO'da her bir parçacık bir çözüm adayıdır.

Örneğin, D (dimension) değişkenli bir problemin uygunluk fonksiyonu hesabı için;

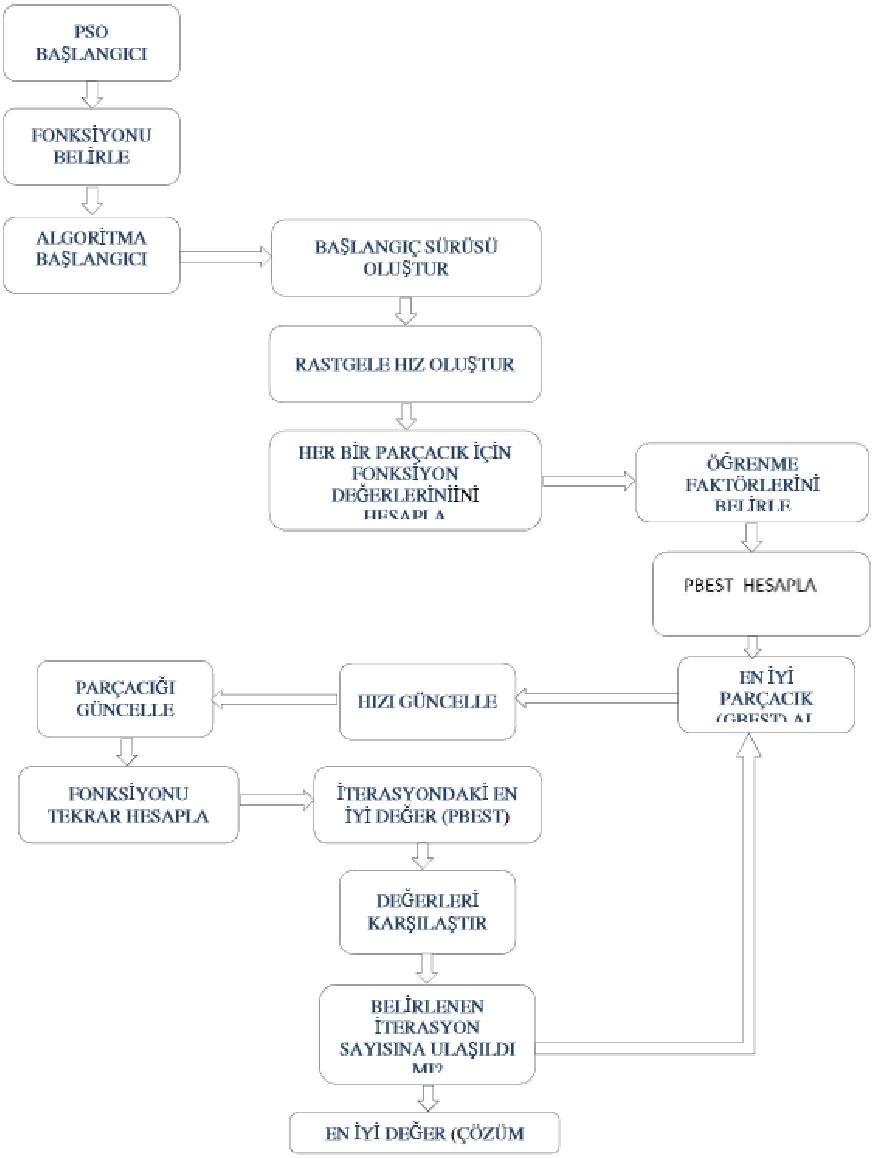
n adet parçacık (particle) ile bir başlangıç çözümü denklem (3.1)'de verilmiştir [28].

$$\begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1D} \\ x_{21} & x_{22} & \cdots & x_{2D} \\ \cdots & \cdots & \cdots & \cdots \\ x_{n1} & x_{n2} & \cdots & x_{nD} \end{bmatrix} \tag{3.1}$$

i. parçacık için uygunluk fonksiyonu aşağıda denklem (3.2)'de verilmiştir.

$$f_{fitness} = f(x_{i1}, x_{i2}, x_{i3}, \dots, x_{iD}) \tag{3.2}$$

Yukarıdaki matriste (3.1), parçacık(particle) her bir satır için bir çözümü ifade etmektedir. ***n*** parçacıklı bir problemde n adet çözüm elde edilmektedir. Parçacıkların hareketi iki önemli parametreye bağlıdır. Her bir iterasyonda önce ***pbest*** sonra ***gbest*** bulunur. pbest bireyin kendisinin o ana kadar bulduğu en iyi çözümüdür. gbest sürünün o ana kadar bulduğu en iyi çözümüdür. Bu şekilde çalışan bir PSO'da sadece bir adet gbest varken n adet pbest bulunur. Algoritmanın başlangıcında her bir parçacık/çözüm için verilen rastgele pbest değerleri aynı zamanda her parçacığın/çözümün başlangıç değeridir. İterasyon başladıktan sonra parçacıkların konumu diğer parçacıkların durumuna göre güncellenir. Güncellendikten sonra kendisinin en iyi çözümü olan pbest ile karşılaştırılır. Yeni çözüm pbest'ten daha iyi bir çözüm ise yeni çözüm pbest olarak atanır [28].



Şekil 3.1 - Parçacık Sürü Optimizasyonunun Akış Diyagramı

Algoritma her bir parçacık için tanımlı aralık içinde hız ve konum değerinin atanmasıyla başlar. Her bir parçacığın bir sonraki hızı hem kendisinin pbest değerine hem de sürüsüne bağlı olarak yani gbest değerine bağlı olarak oluşturulur. Her bir iterasyonda parçacıklar en uygun çözüme yaklaşırlar. Geleneksel PSO'da parçacığa ait en basit haliyle hız ve konum formülleri denklem (3.3) ve (3.4)'te verilmiştir. [21,22].

$$v_i^{k+1} = v_i^k + c_1 rand()\left(p_{best\,i}^{\;k} - x_i^k\right) + c_2 rand()\left(g_{best}^{\;k} - x_i^k\right) \tag{3.3}$$

Bu denklemde **k,** iterasyon sayısıdır. v_i^k, i. parçacığın k. iterasyondaki hızıdır. v_i^{k+1} ise bir sonraki iterasyon için hızıdır. c_1 ve c_2 sosyal (social) ve bilişsel (cognitive) parametrelerdir. Yukarıdaki formüle göre hareket eden kuş, hem kendi en iyi çözümü hem de küresel en iyi çözümün konumunu kullanarak arama uzayının ilgili boyutunda hareket eder. Hangi en iyi çözümü kullanacağı c_1 ve c_2 katsayılarına bağlıdır. Bu katsayılardan c_1 katsayısı yüksek seçilirse kendi en iyi bulduğu çözüm etrafında arama yapar buna bölgesel arama denir. Eğer c_2 katsayısı yüksek seçilirse küresel en iyi çözüm etrafında arama yapar buna ise küresel arama denir. c_1 ve c_2 katsayıları literatürde genellikle birbirine eşit sayılar seçilmiştir. Eberhart ve Shi yaptıkları bir çalışmada bu katsayılar için $c_1 = c_2 = 1.4941$ olarak alınabileceğini göstermişlerdir [23]. Rand(), düzgün rastgele dağılımdan rastgele çekilen 0 ve 1 arasında bir sayıdır. Hız hesaplandıktan sonra konum aşağıdaki formülle hesaplanır.

$$x_i^{k+1} = x_i^k + v_i^{k+1} \tag{3.4}$$

(3.4) numaralı denklemde x_i^k mevcut (eski) parçacığın konum/çözüm değerini göstermektedir. x_i^{k+1} mevcut parçacığın yeni konumu/çözümü değerini göstermektedir.

(3.3) numaralı denklemde sadece v_i^k hızının değişmesi durumunda, parçacığın/çözümün sabit hız ile ilk konumundan/çözümünden, hızını ve yönünü değiştirmeden hareket edeceği gözlemlenebilir. Sadece kalan kısmın uygulanması durumunda ise, parçacıkların/çözümlerin o an ki uygun çözüme en yakın parçacık/çözüm yönünde hareket ettiği, rastgele yeni bir parçacık daha uygun olur ise bir ilerleme olacağı, aksi takdirde giderek daralan bir alanda arama yapılacağı görülmektedir. Bu davranışlar, farklı PSO yöntemlerinin ortaya çıkmasında etkili olmuşlardır.

3.2 Atalet Değerli PSO

Bu PSO algoritması hakkında, sürünün taradığı alan genişliği hakkında aşağıdaki yorumlara ulaşılmaktadır [24-27]:

- Eski hız değeri, sürünün geniş bir alanı taramasını sağlar.
- Eşitliğin ikinci kısmı ise sürünün daha dar bir alanı taramasını sağlar.

Elde edilen bu yorumlardan başka, algoritmanın davranışı ile ilgili aşağıdaki istekler de bulunmalıdır ve gerçekleştirilebilir olmalıdır:

- Algoritma mümkün olduğu kadar geniş bir alanı taramalıdır.
- Algoritma optimum değere ulaşmalıdır.

Elde edilen bu sonuçlara göre sürünün davranışı matematiksel olarak ifade edilmeden önce sözlü veya dilsel olarak ifade edilebilir. Sürünün başlangıçta geniş bir alanı taraması istenmeli ve sürü optimum değere yaklaştıkça taranması istenilen alan küçültülmeli ve bu sayede optimum değer elde edilebiliyor olmalıdır. Bu işlemin gerçekleştirilebilmesi için eski hız değerinin eşitlik içinde kontrol edilmesi gerekmektedir. Bu durum geleneksel PSO algoritmasında, eski hız değeri bir katsayı ile çarpılarak IWPSO yöntemi olarak adlandırılan eşitliklerle sağlanır. Bu katsayının hesaplanmasıyla ilgili formül Denklem 3.5'te verilmiştir.

$$v_{id} = wv_{id} + c_1 rand()\,(p_{id} - x_{id}) + c_2 rand()\,(p_{gd} - x_{id}) \tag{3.4}$$

$$w = w_{max} - \frac{w_{max} - w_{min}}{j_{max}} j \tag{3.5}$$

w_{max} ve w_{min} değerleri sırası ile 0.9 ve 0.4 değerlerine sahiptir. Değişkenlerin aldığı bu değerler daha önce yapılan deneysel çalışmalar sonucunda elde edilmiştir [24-25]. j değişkeni mevcut iterasyonun indeksini, j_{max} ise maksimum iterasyon sayısını ifade etmektedir.

3.3 Adaptif Kısıtlamalı PSO

$$v_i^{k+1} = K\left(v_i^k + c_1 rand()\left(p_{best\,i}^{\;k} - x_i^k\right) + c_2 rand()\left(g_{best} - x_i^k\right)\right) \tag{3.6}$$

Bu denklemde **k** iterasyon sayısıdır. v_i^k i. parçacığın k. iterasyondaki hızıdır. v_i^{k+1} ise bir sonraki iterasyon için hızıdır. K, c_1 ve c_2 birbirine bağlı katsayılardır. K kısıtlama faktörü, parçacıkların hareketlerindeki osilasyonu sönümleme etkisine sahiptir. Bu sebeple parçacıklar zamanla sonuca yakınsarlar. c_1 ve c_2 sosyal (social) ve bilişsel (cognitive) parametrelerdir. Rand() düzgün rastgele dağılımdan çekilen 0 ve 1 arasında bir sayıdır. Hız hesaplandıktan sonra konum aşağıdaki formülle hesaplanır.

$$x_i^{k+1} = x_i^k + v_i^{k+1} \tag{3.7}$$

K, c_1 ve c_2 'ye bağlı bir sabittir. Denklem (3.8)'de K sabitinin formülü verilmiştir.

$$K = \frac{2}{\left|2 - c - \sqrt{c^2 - 4c}\right|} \quad , \quad c = c_1 + c_2 \quad , \quad c > 4 \tag{3.8}$$

4. ÖNERİLEN FORMÜLASYONLAR VE UYGULANMASI

PSO yöntemi, tüm sürünün en iyisi bulunurken sergilediği davranış aracılığı ile elde edilmektedir. Şimdiye dek yapılmış ve yayımlanmış pek çok araştırmada, parçacıkların her birinin davranışları ayrı ayrı esas alınmayıp yerine sürünün genel davranışı temel alınmıştır. Bu çalışmanın amacı ise, geleneksel PSO'dan geliştirilen atalet ağırlığı (w) ve adaptif kısıt parametresini (K) ayrı ayrı fonksiyonel şekilde azaltarak karşılaştırma fonksiyonları üzerinde algoritmanın performansını incelemektir.

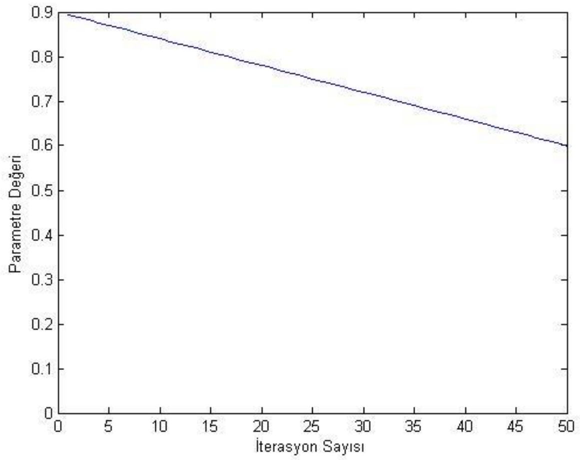
4.1 Tavlama Benzetimi

Tavlama Benzetimi (TB) orijinini doğal tavlama işleminden almaktadır. Bu algoritmanın optimizasyon problemlerine uygulanması ile ilgili çalışmalar 1983 yılında Kirkpatrick ve arkadaşları tarafından yapılan bir çalışma ile başlamıştır [29]. Algoritma metallerin tavlama işlemi ile bir optimizasyon problemine çözüm araştırma olayları arasındaki benzerlikten ilham alınarak ortaya konulmuştur. Burada metaldeki atomların durumları optimizasyon probleminin muhtemel çözümlerine ve durumların enerjileri de çözümlere ait amaç fonksiyon değerlerine karşılık gelmektedir. Hızlı soğutma işlemi ise bölgesel optimizasyon işlemine tekabül etmektedir. Dış sıcaklık sıfır olduğunda daha yüksek enerjili bir duruma geçiş mümkün olmaz. Böylece bölgesel optimizasyondaki gibi yukarı doğru hareketler yasaklanır ve araştırma bir bölgesel minimale takılı kalır. Bu işlemde sıcaklık (T) çeşitli seviyeler boyunca yavaşça düşürülür. Enerji seviyesinden uzaklaşmamayı sağlamak için mevcut sıcaklığı belirli bir süre muhafaza etmek suretiyle ve sıfır dereceye yaklaşıncaya kadar bu işlemlerin tekrarlanması bölgesel optimallikten kaçışı sağlayabilmektedir [13].

Bu çalışmada TB algoritmasının sıcaklık azaltma fonksiyonlarından esinlenilmiştir.

4.1.1 Aritmetik Azalma

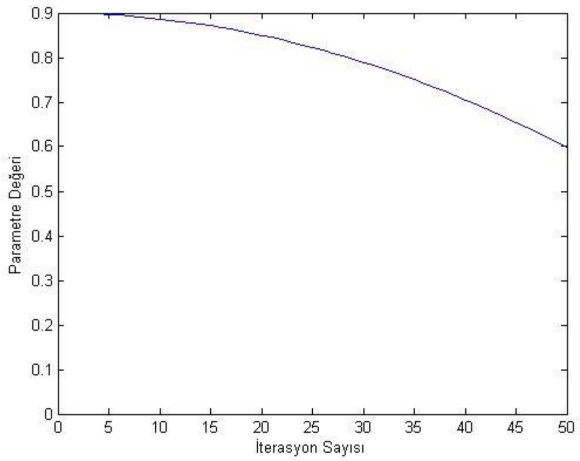
Aritmetik azalma Şekil 4.1'de görüldüğü gibi her bir iterasyonda sabit bir oran ile azalmaktadır.



Şekil 4.1 - Aritmetik Azalma Grafiği

4.1.2 Geometrik Azalma

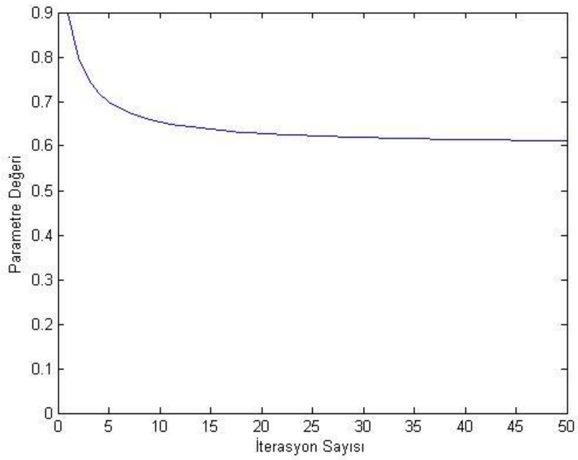
Geometrik azalma Şekil 4.2'de görüldüğü gibi ilk iterasyonlarda azalma miktarı daha az iken ilerleyen iterasyonlarda ise azalma miktarı artmaktadır.



Şekil 4.2 – Geometrik Azalma Grafiği

4.1.3 Ters Fonksiyonel Azalma

Ters Fonksiyonel azalma Şekil 4.3'de görüldüğü gibi geometrik azalmanın tersi olarak düşünülebilir. İlk iterasyonlarda azalma miktarı çok iken ilerleyen iterasyonlarda ise azalma miktarı azalmaktadır.



Şekil 4.3 – Ters Fonksiyonel Azalma Grafiği

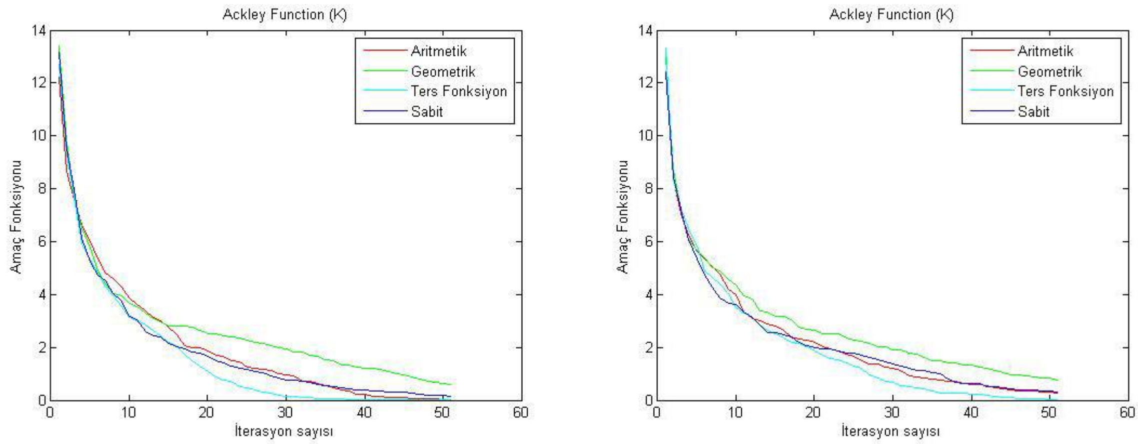
4.2 Deneysel Çalışmalar

Bu çalışmada, yeni üretilen formülasyonlar "karşılaştırma fonksiyonu" ile isimlendirilen ve optimizasyon algoritmalarının test edilmesinde kullanılan fonksiyonlar kullanılarak test edilmektedir. Halihazırda pek çok karşılaştırma fonksiyonu mevcuttur. Bu karşılaştırma fonksiyonları, sıklıklarına ve yerel optimum değerlerine göre sınıflandırılmaktadır. Yapılan çalışmada ise sıklıkla tercih edilen on altı fonksiyon seçilmiştir. Ortalama en optimum hata değerlerine göre test sonuçları karşılaştırılıp yorumlanmaktadır. Karşılaştırma işlemi K, adaptif kısıtlama ve w, atalet ağırlığı parametreleri için ayrı ayrı azalma fonksiyonları üzerinde uygulanmaktadır. K faktörü sabit olarak alındığında 0.7298 olarak, azalan fonksiyonlar kullanıldığında ise maksimum değer olarak 0.9, minimum değer olarak 0.6 değerleri alınmıştır. c1 ve c2 parametrelerine ise sırasıyla 2 ve 2.1 değerleri verilmiştir. w faktöründe ise sabit olarak alındığında 0.79 olarak, azalan fonksiyonlar kullanıldığında maksimum değer olarak 0.9, minimum değer olarak 0.6 değerleri alınmıştır. c1 ve c2 parametrelerine ise 1.49 değeri verilmiştir. Geliştirilen algoritma her bir fonksiyon için 50 iterasyon ile çalışmaktadır. Parçacık sayısı olarak 30 alınmıştır. Sonuçlar her test için 30 defa çalıştırıp ortalaması alınarak grafiğe dökülmüştür.

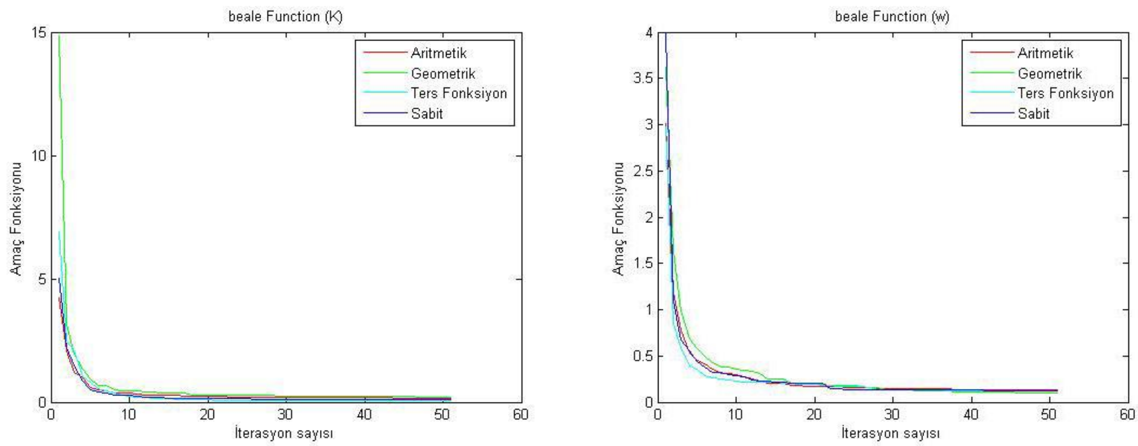
Çizelge 4.1 - Uygulamada kullanılan karşılaştırma fonksiyonları

Fonksiyon Adı	Fonksiyon	Değer Aralığı
Ackley	$20 + e - 20e^{\frac{-1\sqrt{\frac{1\sum_{i=1}^{n} x_i^2}{n}}}{5}} - e^{-\frac{1}{n\sum_{i=1}^{n}\cos(2\pi x_i)}}$	$-15 \le x_i \le 30$
Beale	$(1.5 - x_1 + x_1x_2)^2 + (2.25 - x_1 + x_1x_2^2)^2 + (2.625 - x_1 + x_1x_2^3)^2$	$-4.5 \le x_i \le 4.5$
Booth	$(x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2$	$-10 \le x_i \le 10$
Colville	$100(x_1^2 - x_2)^2 + (x_1 - 1)^2 + (x_3 - 1)^2 + 90(x_3^2 - x_4)^2 + 10.1((x_2 - 1)^2 + (x_4 - 1)^2) + 19.8(x_2 - 1)(x_4 - 1)$	$-10 \le x_i \le 10$
Dixon&Price	$(x_1 - 1)^2 + \sum_{i=2}^{n} i\,(2x_i^2 - x_{i-1})^2$	$-10 \le x_i \le 10$
Griewank	$\sum_{i=1}^{n} \frac{x_i^2}{4000} - \prod_{i=1}^{n} \cos(x_i/\sqrt{i}) + 1$	$-600 \le x_i \le 600$
Hump	$f(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	$-5 \le x_i \le 5$
Levy	$f(x) = sin^2(\pi y_1) + \sum_{i=1}^{n-1}[(y_i - 1)^2(1 + 10sin^2(\pi y_i + 1))] + (y_n - 1)^2(1 + 10sin^2(2\pi y_n))$	$-10 \le x_i \le 10$
Matyas	$f(x) = 0.26(x_1^2 + x_2^2) - 0.48x_1x_2$	$-10 \le x_i \le 10$
Perm	$\sum_{k=1}^{n}[\sum_{i=1}^{n}(i^k + \beta)(\frac{x_i}{i} - 1)^k]^2$	$-n \le x_i \le n$
Rastrigin	$10n + \sum_{i=1}^{n}(x_i^2 - 10\cos(2\pi x_i))$	$-5.12 \le x_i \le 5.12$

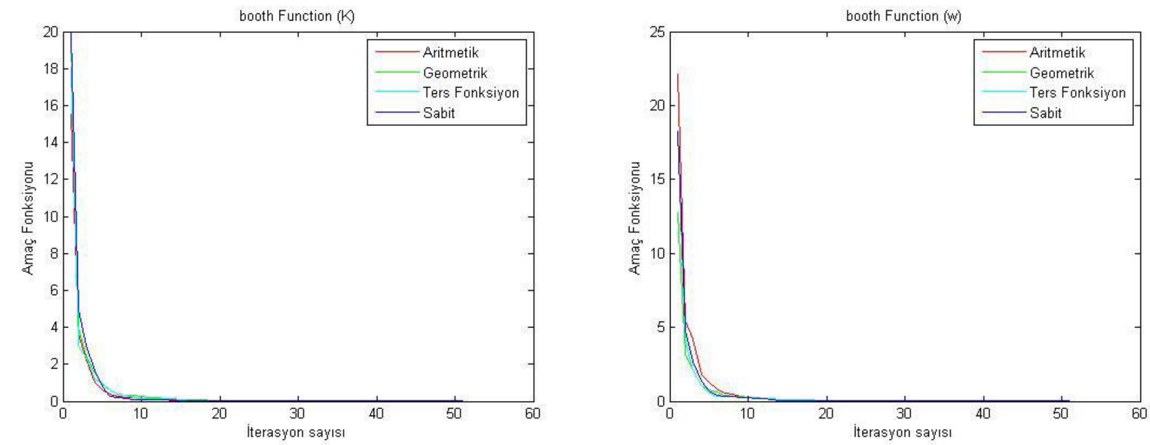
Rosenbrock	$\sum_{i=1}^{n-1}[100\left(x_i^2+x_i\right)^2+(x_i-1)^2]$	$-5 \leq x_i \leq 10$
Schwefel	$418.9829\mathrm{n} - \sum_{i=1}^{n} x,\ sin\sqrt{\lvert x_i \rvert}$	$-500 \leq x_i \leq 500$
Sphere	$f(x) = \sum_{i=1}^{n} x_i^2$	$-5.12 \leq x_i \leq 5.12$
Sum Squares	$f(x) = \sum_{i=1}^{n} i\, x_i^2$	$-10 \leq x_i \leq 10$
Zakharov	$z_n(x) = \sum_{i=1}^{n} x_i^2 + \left(\sum_{i=1}^{n} 0{,}5i\ x_i\right)^2 + \left(\sum_{i=1}^{n} 0{,}5i\ x_i\right)^4$	$-5 \leq x_i \leq 10$



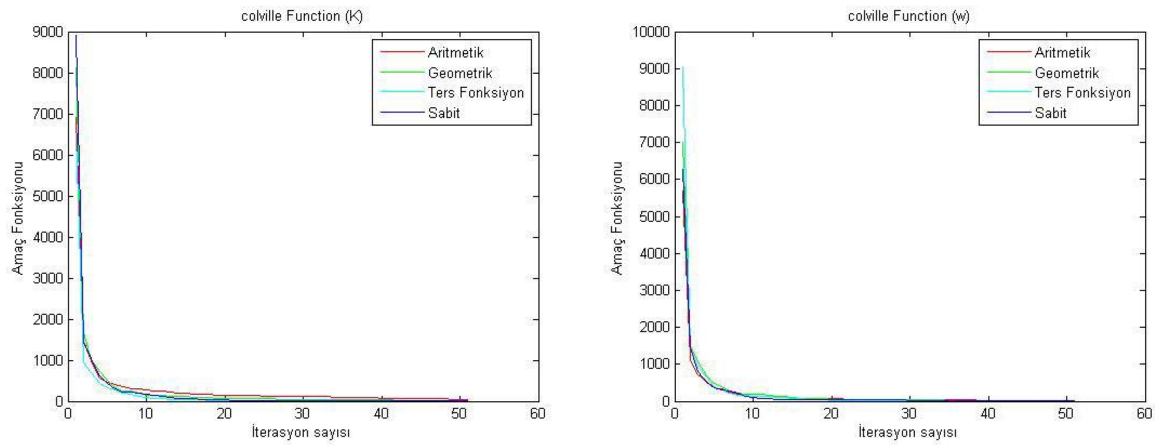
Şekil 4.4 - Ackley Fonksiyonunda w, Atalet Ağırlığının ve K, Adaptif Kısıtlama Faktörünün İterasyon Sayısıyla Azalmasının Sonuca Etkisi



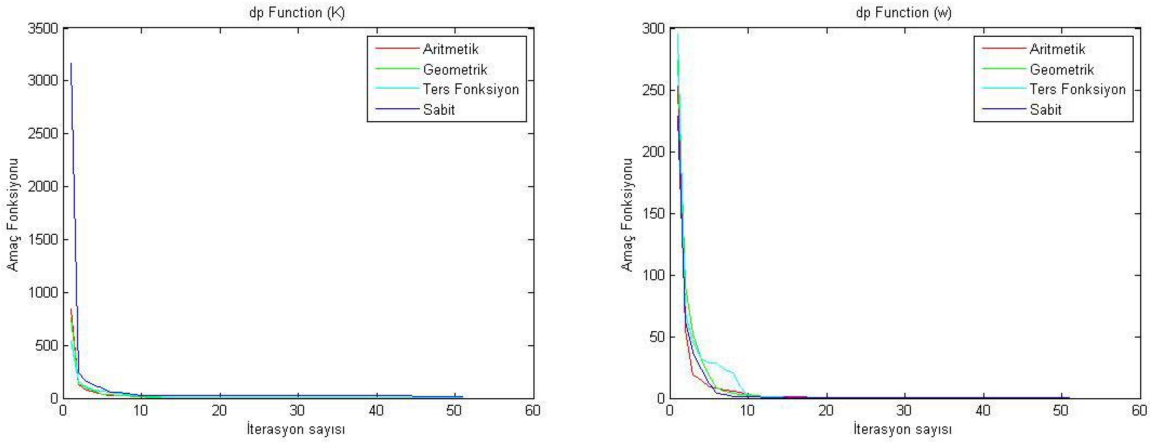
Şekil 4.5 - Beale Fonksiyonunda w, Atalet Ağırlığının ve K, Adaptif Kısıtlama Faktörünün İterasyon Sayısıyla Azalmasının Sonuca Etkisi



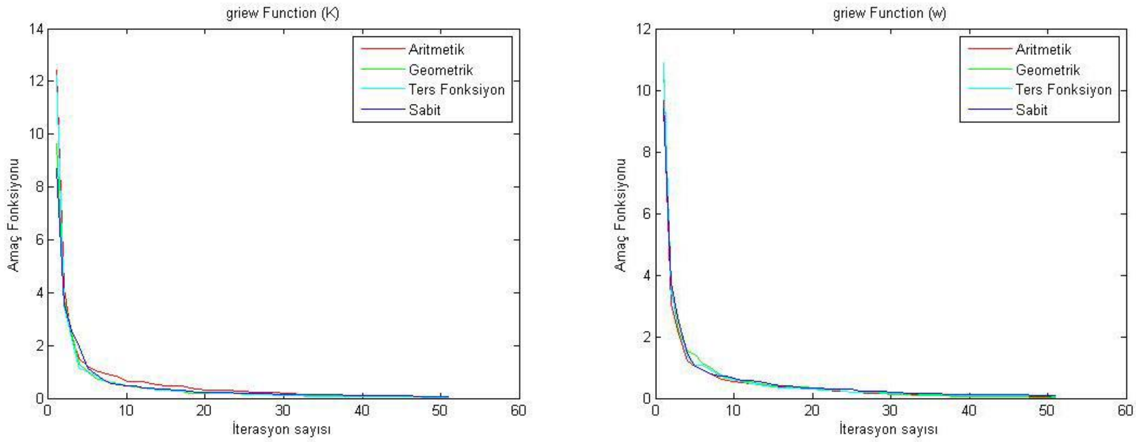
Şekil 4.6 - Booth Fonksiyonunda w, Atalet Ağırlığının ve K, Adaptif Kısıtlama Faktörünün İterasyon Sayısıyla Azalmasının Sonuca Etkisi



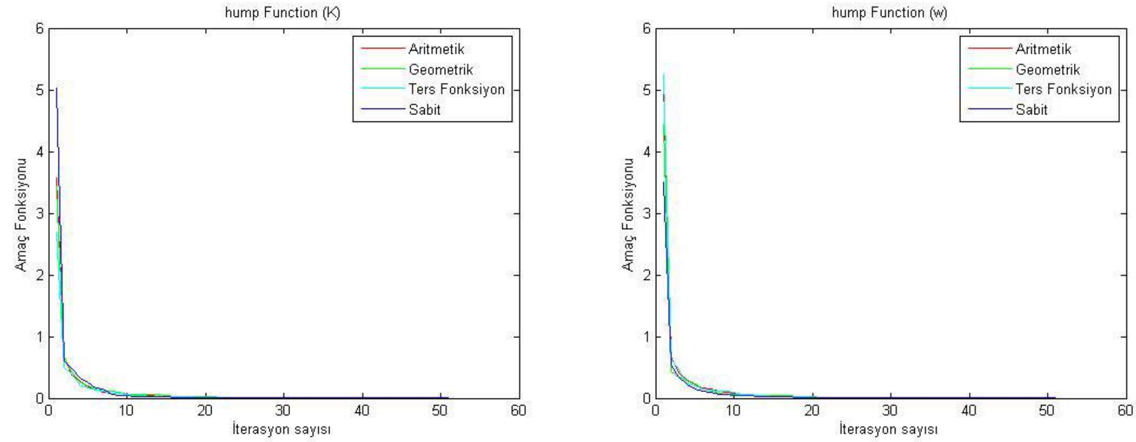
Şekil 4.7 - Colville Fonksiyonunda w, Atalet Ağırlığının ve K, Adaptif Kısıtlama Faktörünün İterasyon Sayısıyla Azalmasının Sonuca Etkisi



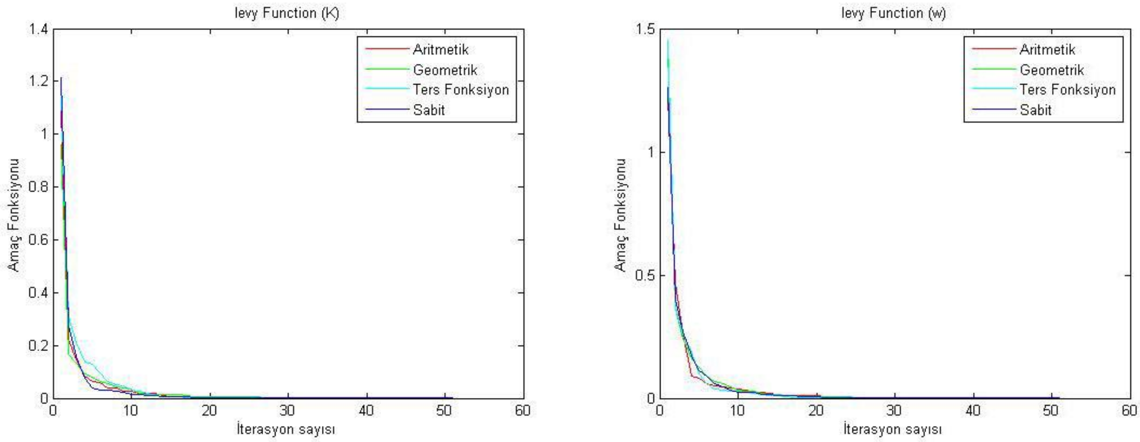
Şekil 4.8 – Dixon & Price Fonksiyonunda w, Atalet Ağırlığının ve K, Adaptif Kısıtlama Faktörünün İterasyon Sayısıyla Azalmasının Sonuca Etkisi



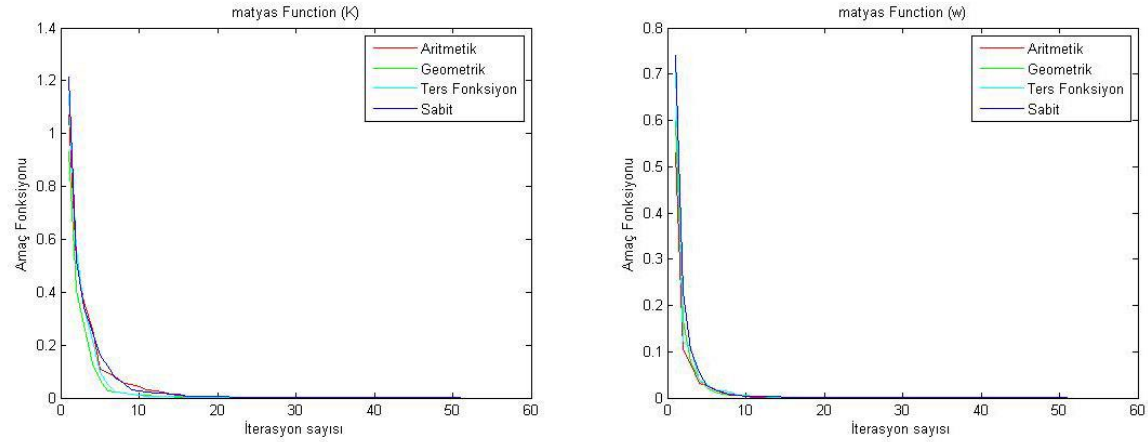
Şekil 4.9 - Griewank Fonksiyonunda w, Atalet Ağırlığının ve K, Adaptif Kısıtlama Faktörünün İterasyon Sayısıyla Azalmasının Sonuca Etkisi



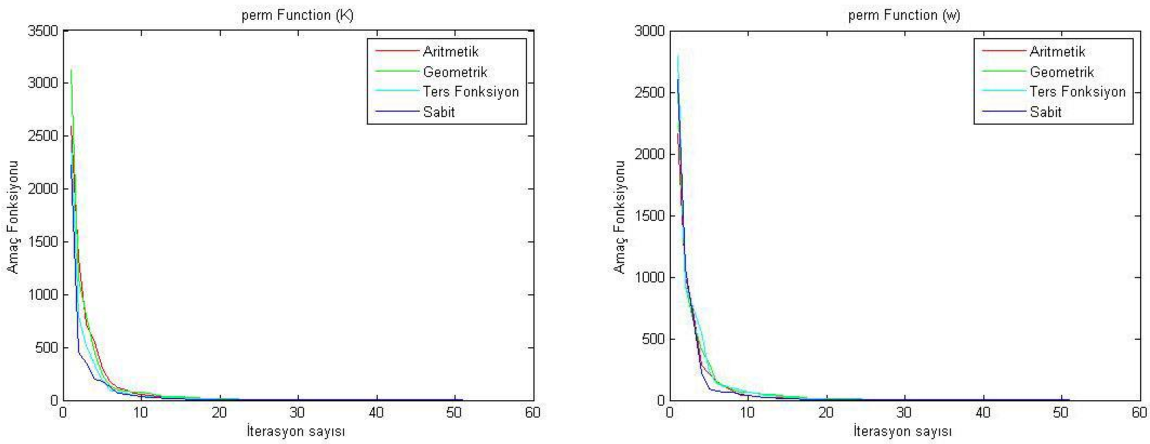
Şekil 4.10 - Hump Fonksiyonunda w, Atalet Ağırlığının ve K, Adaptif Kısıtlama Faktörünün İterasyon Sayısıyla Azalmasının Sonuca Etkisi



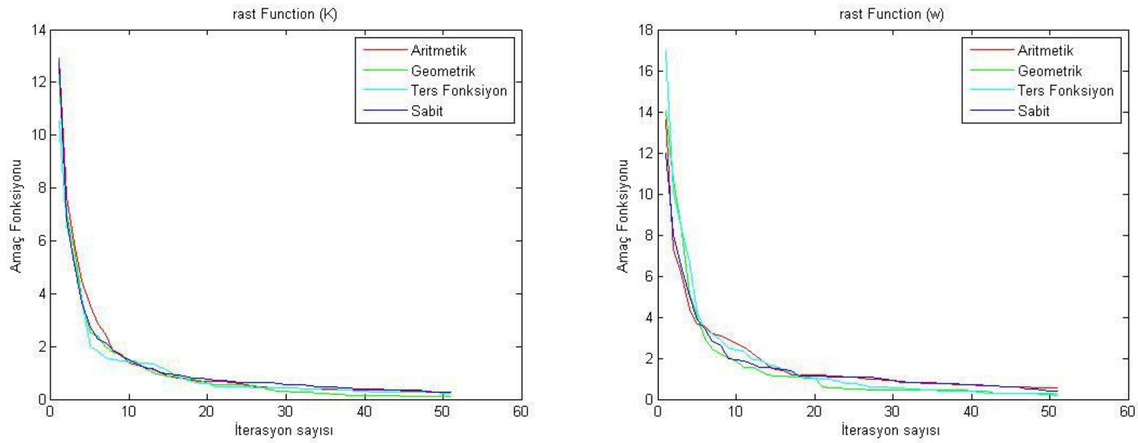
Şekil 4.11 - Levy Fonksiyonunda w, Atalet Ağırlığının ve K, Adaptif Kısıtlama Faktörünün İterasyon Sayısıyla Azalmasının Sonuca Etkisi



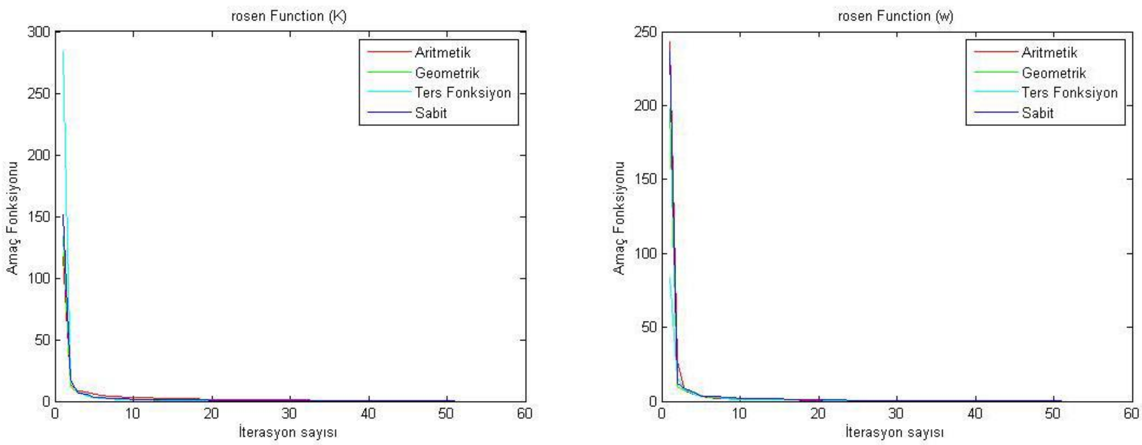
Şekil 4.12 - Matyas Fonksiyonunda w, Atalet Ağırlığının ve K, Adaptif Kısıtlama Faktörünün İterasyon Sayısıyla Azalmasının Sonuca Etkisi



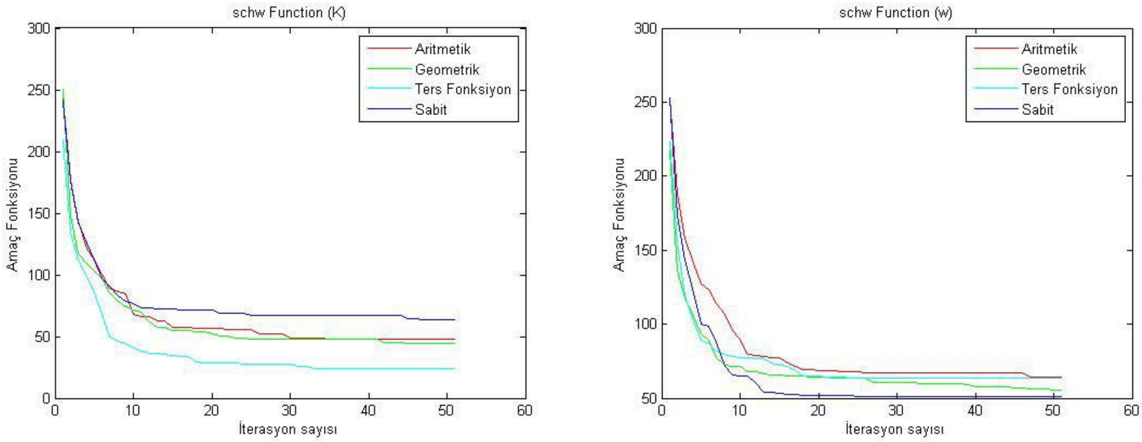
Şekil 4.13 - Perm Fonksiyonunda w, Atalet Ağırlığının ve K, Adaptif Kısıtlama Faktörünün İterasyon Sayısıyla Azalmasının Sonuca Etkisi



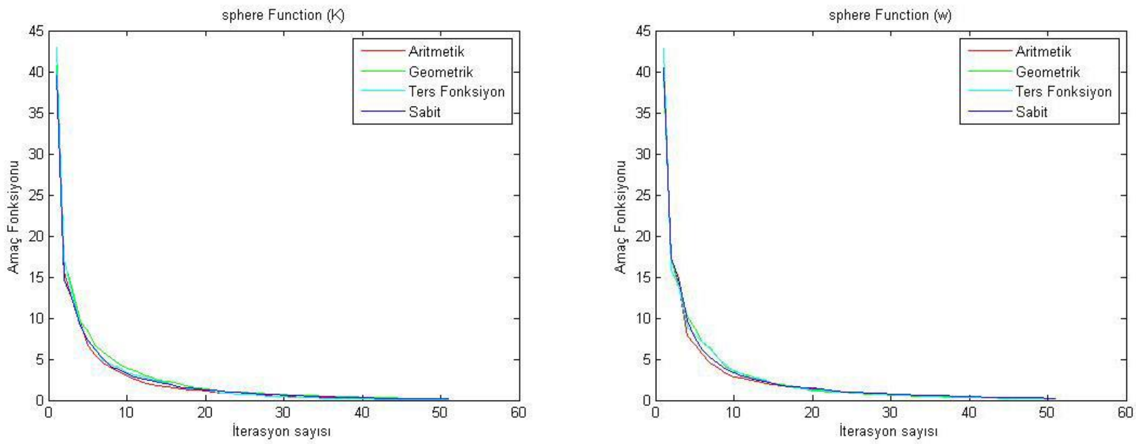
Şekil 4.14 - Rastrigin Fonksiyonunda w, Atalet Ağırlığının ve K, Adaptif Kısıtlama Faktörünün İterasyon Sayısıyla Azalmasının Sonuca Etkisi



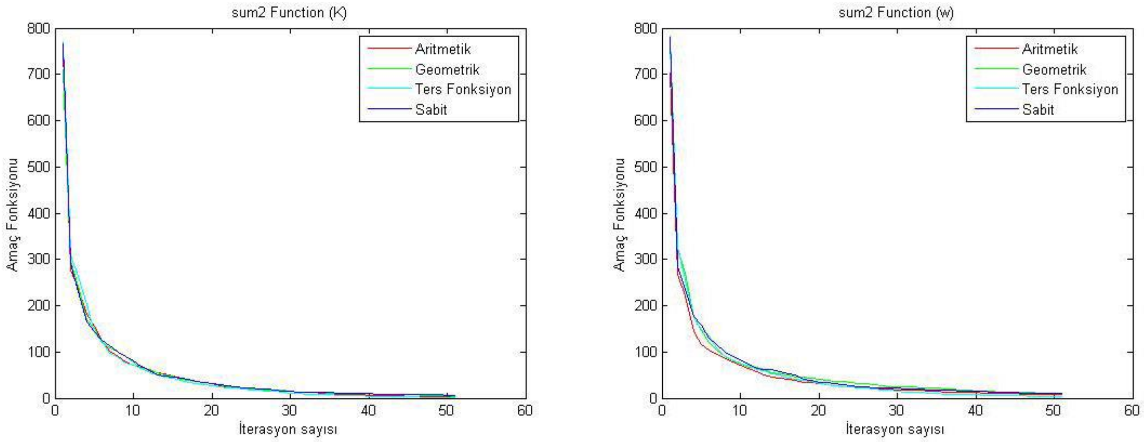
Şekil 4.15 - Rosenbrock Fonksiyonunda w, Atalet Ağırlığının ve K, Adaptif Kısıtlama Faktörünün İterasyon Sayısıyla Azalmasının Sonuca Etkisi



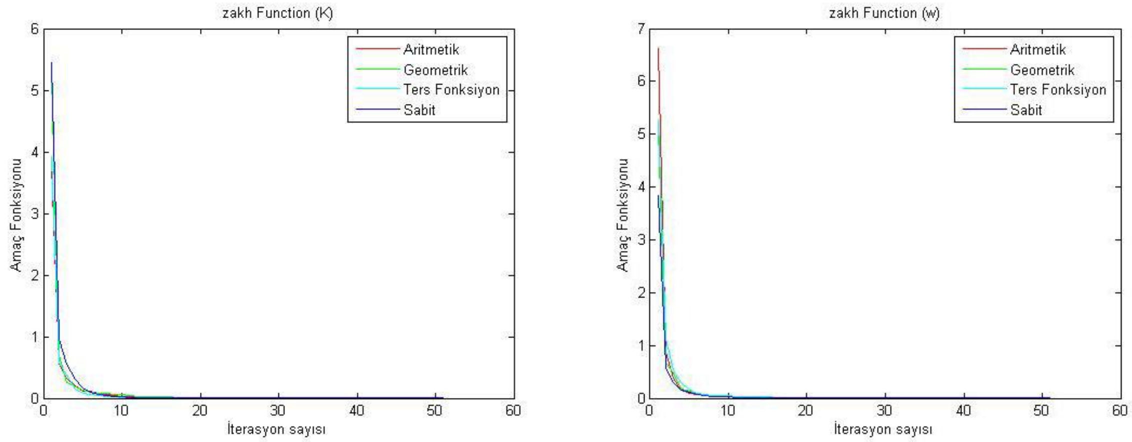
Şekil 4.16 - Schwefel Fonksiyonunda w, Atalet Ağırlığının ve K, Adaptif Kısıtlama Faktörünün İterasyon Sayısıyla Azalmasının Sonuca Etkisi



Şekil 4.17 - Sphere Fonksiyonunda w, Atalet Ağırlığının ve K, Adaptif Kısıtlama Faktörünün İterasyon Sayısıyla Azalmasının Sonuca Etkisi



Şekil 4.18 - Sum Squares Fonksiyonunda w, Atalet Ağırlığının ve K, Adaptif Kısıtlama Faktörünün İterasyon Sayısıyla Azalmasının Sonuca Etkisi



Şekil 4.19 - Zakharov Fonksiyonunda w, Atalet Ağırlığının ve K, Adaptif Kısıtlama Faktörünün İterasyon Sayısıyla Azalmasının Sonuca Etkisi

Çizelge 4.2 – Adaptif Kısıt Parametreli (K) PSO ile bulunan optimum değerler

Fonksiyon Adı	Aritmetik	Geometrik	Ters Fonksiyon	Sabit
Ackley	0,0308000	0,6085000	0,0016000	0,1572000
Beale	0,1720000	0,2051000	0,0512000	0,1062000
Booth	0,0000020	0,0010000	0,0000002	0,0000080
Colville	45,6293000	7,2770000	0,0197000	0,2570000
Dixon & Price	0,0047000	0,0437000	0,0000010	11,5731000
Griewank	0,0661000	0,0519000	0,0382000	0,0696000
Hump	0,0000013	0,0000680	0,0000001	0,0000070
Levy	0,0000001	0,0000142	0,0000000	0,0000024
Matyas	0,0000002	0,0000076	0,0000000	0,0000053
Perm	0,6300000	1,6206000	0,4596000	1,0423000
Rastrigin	0,2699000	0,0974000	0,2370000	0,2716000
Rosenbrock	0,3719000	0,0238000	0,0000670	0,4487000
Schwefel	47,3773000	44,1185000	23,6945000	63,1872000
Sphere	0,0564000	0,1636000	0,0550000	0,1646000
Sum Square	2,5387000	4,6574000	1,5271000	6,8614000
Zakharov	0,0000001	0,0000094	0,0000001	0,0000014

Adaptif kısıt parametresi (K), fonksiyonel şekilde (aritmetik, geometrik ve ters fonksiyon) azaltılarak karşılaştırma (benchmark) fonksiyonları üzerinde algoritma performansının sonuçları verilmiştir. Sonuçlara baktığımızda ters fonksiyonun diğer azaltma fonksiyonları arasında en iyi sonuçları verdiği gözlenmiştir.

Çizelge 4.3 - Atalet Ağırlıklı (w) PSO ile bulunan optimum değerler

Fonksiyon Adı	Aritmetik	Geometrik	Ters Fonksiyon	Sabit
Ackley	0,2790000	0,7518000	0,0308000	0,3163000
Beale	0,1292000	0,1043000	0,1271000	0,1282000
Booth	0,0000400	0,0038000	0,0000080	0,0000900
Colville	7,0332000	4,5522000	0,9159000	8,2648000
Dixon & Price	0,0052000	0,0327000	0,0014000	0,0095000
Griewank	0,0509000	0,0294000	0,0653000	0,0970000
Hump	0,0000380	0,0025000	0,0000210	0,0012000
Levy	0,0000049	0,0000514	0,0000010	0,0000126
Matyas	0,0000002	0,0000005	0,0000002	0,0000022
Perm	0,4503000	1,3568000	0,6864000	1,2465000
Rastrigin	0,5435000	0,1942000	0,2671000	0,4156000
Rosenbrock	0,1543000	0,0830000	0,0142000	0,0403000
Schwefel	64,3034000	55,4425000	63,16800s00	51,3742000
Sphere	0,1481000	0,1562000	0,1365000	0,2584000
Sum Square	7,3370000	9,9821000	3,5485000	9,5568000
Zakharov	0,0000030	0,0000579	0,0000026	0,0000086

Atalet ağırlığı (w), fonksiyonel şekilde (aritmetik, geometrik ve ters fonksiyon) azaltılarak karşılaştırma (benchmark) fonksiyonları üzerinde algoritma performansının sonuçları verilmiştir. Sonuçlara baktığımızda ters fonksiyonun diğer azaltma fonksiyonlar arasında en iyi sonuçları verdiği gözlenmiştir. Ayrıca bazı test problemlerinde geometrik azalma fonksiyonunun da iyi sonuçlar verdiği gözlenmiştir.

5. SONUÇLAR VE ÖNERİLER

Bu çalışmada, son dönemlerde yaygın olarak çalışılan bir algoritma olan PSO algoritmasının çözüm üzerindeki etkisinin yükseltilmesi hedeflenmiştir. Optimizasyon algoritmalarının başarısını etkileyen faktörlerden en önemlisi, parametrelerinin optimizasyonudur.

PSO algoritmasının başarısı az sayıda parametre kullanması ve hızlı yakınsamasıdır. Geliştirilen PSO algoritmalarında kullanılan parametrelerden en önemlilerinden K kısıtlama faktörü ve w atalet ağırlığı parametrelerinin çözüm başarısını arttıracak şekilde dinamik olarak kullanılması düşünülmüştür. BT'deki sıcaklık parametresinden esinlenilerek genellikle sabit olarak kullanılan bu parametreler aritmetik, geometrik ve ters fonksiyon ile azaltılarak çözüm başarısı incelenmiştir.

Algoritmaların performansını test etmek için kısıtsız optimizasyon problemlerinin 16 tanesi çözdürülmüştür. Algoritmalar Matlab© programlamada, Intel i7-4700 HQ 2.4GHz işlemci ve 8GB RAM'e sahip donanımlı bir bilgisayarda simüle edilmiştir.

Her bir karşılaştırma fonksiyonu için algoritmalar 30 defa çalıştırılıp ortalamaları grafikler üzerinde ve tablolarda verilmiştir. Sonuçlardan görüleceği üzere, atalet ağırlığı (w) ve adaptif kısıt parametresini (K) kullanan PSO'larda azalan fonksiyonlar ile yapılan testler sabit değer ile yapılan testlere göre daha iyi sonuçlar verdiği gözlemlenmiştir. Gerçek zamanlı uygulamalar için geliştirilen PSO algoritmalarında, parametrelerin sabit kullanımı yerine parametrelerin fonksiyonel şekilde azaltılarak kullanılabileceği önerilmektedir.

KAYNAKÇA

[1] H. Haklı ve H. Uğuz, «A novel particle swarm optimization algorithm with Levy flight,» *Applied Soft Computing,* no. 23, pp. 333-345, 2014.

[2] A. H. Kashan, «League Championship Algorithm (LCA): An algorithm for gloal optimization inspired by sport championships,» *Applied Soft Computing,* no. 16, pp. 171-200, 2014.

[3] D. E. Goldberg, Genetic Algorithms in Search Optimization and Machine Learning, Addison Wesley, 1989.

[4] T. Y. Lim, «Structured population genetic algorithms: a literature survey,» *Artificial Intelligence Review,* no. 41, pp. 385-399, 2014.

[5] M. Dorigo ve C. Blum, «Ant colony optimization theory: A survey,» *Theoretical Computer Science,* pp. 344, 243-278, 2005.

[6] J. Kennedy ve R. Eberhart, «IEEE International Conference,» %1 içinde *Particle swarm optimization, Neural Networks, Proceedings*, 1995.

[7] P. Erdoğmuş, «Particle Swarm Optimization Performance on Special Linear Programming Problems,» *Scientific Research and Essays,* no. 5(12), pp. 1506-1518, 2010.

[8] D. Karaboga ve B. Gorkemli, «A quick artificial bee colony (qABC) algorithm and its performance on optimization problems,» *Applied Soft Computing,* no. 23, pp. 227-238, 2014.

[9] R. Rajabioun, «Cuckoo Optimization Algorithm,» *Applied Soft Computing,* no. 11(8), pp. 5508-5518, 2011.

[10] S. Mishra, K. Shaw ve D. Mishra, «A New Meta-heuristic Bat Inspired Classification Approach for Microarray Data,» *Procedia Technology,* no. 4, pp. 802-806, 2012.

[11] X. S. Yang, S. S. S. Hosseini ve A. H. Gandomi, «Firefly Algorithm for solving nonconvex economic dispatch problems with valve loading effect,» *Applied Soft Computing,* no. 12(3), pp. 1180-1186, 2012.

[12] X.-S. Yang, Nature-Inspired Metaheuristic Algorithms Second Edition, Luniver Press, 2010.

[13] D. Karaboğa, Yapay Zekâ Optimizasyon Algoritmaları, Nobel Yayın Dağıtım, 2011.

[14] Metaheuristic [online], «Wikipedia,» 2016. [Çevrimiçi]. Available: http://en.wikipedia.org/wiki/Metaheuristic. [Erişildi: 19 Aralık 2016].

[15] C. Blum ve A. Roli, «Metaheuristics in combinatorial optimization: Overview and conceptual comparison,» *ACM Computing Surveys,* no. 35, pp. 268-308, 2003.

[16] K. Parsopoulos ve M. Vrahatis, «Particle Swarm Optimization and intelligence: Advances and application.,» *Information Science Reference,* 2010.

[17] W. Reeves, «Particle Systems - a Technique for modeling a class of fuzzy objects,» *Association for Computing Machinery Trans Graphics,* pp. 91-108, 1983.

[18] C. Reynolds, «Flocks, herds and schools: a distributed behavioral model,» *Computer Graphics,* no. 21(4), pp. 25-34, 1987.

[19] F. Hembecker, H. Lopes ve W. Godoy, «Particle Swarm Optimization for the Multidimensional Knapsack Problem,» %1 içinde *International Conference on Adaptive and Natural Computing Algorithms*, Berlin Heidelberg, 2007.

[20] M. Clerc, «The swarm and the queen: towards a deterministic and adaptive particle swarm optimization,» *Evolutionary Computation,* no. 3, pp. 1951-1957, 1999.

[21] D. Parrott ve L. Xiaodong, «Locating and tracking multiple dynamic optima by a particle swarm model using speciation,» *Evolutionary Computation,* no. 10(4), pp. 440,458, 2006.

[22] R. Eberhart ve Y. SHI, «Comparing inertia weights and constriction factors in particle swarm optimization,» *IEEE congress evolutionary computation,* p. 84–88, 2000.

[23] Y. Shi ve R. Eberhart, «A Modified Particle Swarm Optimizer,» *IEEE Congress on Evolutionary Computation,* pp. 69-73, 1998.

[24] Y. Shi ve R. Eberhart, «Empirical Study of Particle Swarm Optimization,» *IEEE Congress on Evolutionary Computation,* pp. 1945-1950, 1999.

[25] M. Clerc, «The Swarm and The Queen: Towards a Deterministic and Adaptive PSO,» pp. 1951-1957, 1999.

[26] M. Clerc ve J. Kennedy, «The Particle Swarm – Explosion, Stability, and Convergence in a Multidimensional Complex Space,» *IEEE Transactions on Evolutionary Computation,* cilt 1, no. 6, pp. 58-73, 2002.

[27] H. Eldem, *Karınca Koloni Optimzasyonu (KKO) ve Parçacık Sürü Optimizasyonu (PSO) Algoritmaları Temelli Bir Hiyerarşik Yaklaşım Geliştirilmesi,* Konya, 2014, pp. 10,13,14.

[28] P. Erdoğmuş ve E. Yalçın, «Parçacık Sürü Optimizasyonu İle Kısıtsız Optimizasyon Test Problemlerinin Çözümü,» *İleri Teknoloji Bilimleri Dergisi,* cilt 4, no. 1, 2015.

[29] S. KirkPatrick, C. D. Gelatt ve M. P. Vecchi, «Optimization by Simulated Annealing,» *Science,* cilt 220, no. 4598, 13 May 1983.

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı	:Erdi YALÇIN
Doğum Tarihi ve Yeri	:20.12.1988 Tarsus
Yabancı Dili	:İngilizce
E-posta	:erdi.yalcin@gmail.com

ÖĞRENİM DURUMU

Derece	**Alan**	**Okul/Üniversite**	**Mezuniyet Yılı**
Lisans	Bilgisayar Öğretmenliği	Düzce Üniversitesi	2011
Lise	Fen Bilimleri	Tarsus Cumhuriyet Lisesi	2005