

**SEZGİSEL ALGORİTMALARIN DENEKTAŞI İŞLEVLER ÜZERİNDE
BAŞARIM METRİKLERİYLE KARŞILAŞTIRILMASI**

Ayşe BAŞTUĞ

YÜKSEK LİSANS TEZİ

**Bilgisayar Mühendisliği Anabilim Dalı
Danışman: Doç. Dr. Cihan KARAKUZU**

**Eskişehir
Anadolu Üniversitesi
Fen Bilimleri Enstitüsü
Mayıs, 2017**

JÜRİ VE ENSTİTÜ ONAYI

Ayşe BAŞTUĞ'un "Sezgisel Algoritmaların Denektaşı İşlevler Üzerinde Başarım Metrikleriyle Karşılaştırılması" başlıklı tezi 22/05/2017 tarihinde aşağıdaki jüri tarafından değerlendirilerek "Anadolu Üniversitesi Lisansüstü Eğitim-Öğretim ve Sınav Yönetmeliği"nin ilgili maddeleri uyarınca, Bilgisayar Mühendisliği Anabilim Dalı'nda Yüksek Lisans tezi olarak kabul edilmiştir.

	<u>Unvanı-Adı Soyadı</u>	<u>İmza</u>
Üye (Tez Danışmanı)	: Doç. Dr. Cihan KARAKUZU
Üye	: Prof. Dr. Yaşar HOŞCAN
Üye	: Doç. Dr. Uğur YÜZGEÇ

Prof. Dr. Nedim DEĞİRMENCİ
Enstitü Müdürü

ÖZET

SEZGİSEL ALGORİTMALARIN DENEKTAŞI İŞLEVLER ÜZERİNDE BAŞARIM METRİKLERİYLE KARŞILAŞTIRILMASI

Ayşe BAŞTUĞ

Bilgisayar Mühendisliği Anabilim Dalı

Anadolu Üniversitesi, Fen Bilimleri Enstitüsü, Mayıs, 2017

Danışman: Doç. Dr. Cihan KARAKUZU

Bilgi çağı olan günümüzde bilgi ve iletişim teknolojileri kullanımının artmasıyla karmaşık problemler çok daha kısa sürede çözülebilmektedir. Teknolojik ilerlemelerle tıpkı bir insan gibi düşünüp, kendini geliştirebilen yapay zeka sistemleri ve en iyisini bulmaya çalışan optimizasyon teknikleri geliştirilmektedir. Optimizasyon işleminde kullanılan tekniklerden sezgisel algoritmalar, canlıların doğal yaşamından esinlenilerek geliştirilmiş olup, en iyi çözüme en yakın olan çözüme kolay ve hızlı bir şekilde ulaşmaktadır. Sezgisel algoritmaların çok sayıda türleri bulunmakta ve gün geçtikçe de yenileri türetilmektedir. Bu çalışmada sezgisel algoritmalar içerisinde bilim dünyasında yaygın olarak kullanılması, açık kaynak kodlu olması ve başarımlarının da diğerlerine görece daha iyi olması gibi sebeplerle sürü tabanlı yapay arı kolonisi(ABC), biyocoğrafya tabanlı optimizasyon(BBO), guguk kuşu arama algoritması(CSO), farksal gelişim algoritması(DE), yayılcı rekabetçi algoritma(ICA) ve parçacık sürü algoritması(PSO) kullanılmıştır. Seçilen bu algoritmaların her biri literatürde sıklıkla kullanılan 8 denektaşı işlev üzerinde küresel minimum noktasını bulmak için aynı başlangıç pozisyonları ve şartları ile 2, 5 ve 10 boyutlu arama uzaylarında 30'ar kez koşturulmuştur. Koşmalar sonucunda algoritmaların başarımları en iyi ölçüt, en kötü ölçüt, doğruluk, kararlılık, zaman ve standart sapma başarımları sonuçlarına bakılarak değerlendirilmiştir. Algoritmaların kümülatif ortalama başarımları değerlerine göre karşılaştırılmasında en iyi performansı DE'nin, takiben de PSO'nun verdiği görülmüştür. Koşma süresi açısından en iyi, kararlılık ve doğruluk açısından DE'ye yakın sonuçlar veren CSO, kümülatif ortalama başarımları değerlendirmesinde üçüncü olmuştur. Kümülatif ortalama başarımları değerlendirme de ICA dördüncü olurken, ona her işlevde en yakın sonuçları veren BBO beşinci olmuştur. Diğer algoritmalarından oldukça uzak değerleri üreten ABC'nin ise bu çalışmada kullanılan başarımları değerlendirmesinde sonuncu olarak en düşük başarımları sergilediği görülmüştür.

Anahtar Sözcükler: Sezgisel algoritmalar, Denektaşı işlevler, Sürü zekası, Başarımları metrikleri, Fonksiyon optimizasyonu.

ABSTRACT

COMPARISON OF HEURISTIC ALGORITHMS WITH PERFORMANCE METRICS ON BENCHMARK FUNCTIONS

Ayşe BAŞTUĞ

Department of Computer Engineering

Anadolu University, Graduate School of Science, May, 2017

Supervisor: Assoc. Prof. Dr. Cihan KARAKUZU

In today's information age, complex problems can be solved in a much shorter time with the increased use of information and communication technologies. Artificial intelligence systems that can think and develop like a human and optimization techniques that try to find the best are being developed with technological advances. The heuristic algorithms which are the techniques used in the optimization process are developed by inspiration from the natural life of the creatures, and reach quickly and easily to the solution that is closest to the best solution. Heuristic algorithms are available in many types and new ones are being developed day by day. In this study, Artificial Bee Colony (ABC), Biogeography-Based Optimization (BBO), Differential Evolution Algorithm (DE), Cuckoo Search Optimization (CSO), Imperialist Competitive Algorithm (ICA), Particle Swarm Optimization (PSO) among many swarm-based heuristic algorithms are used reasons such as widely used in the scientific world, be open source code, and their achievements are better than others. Each of these selected algorithms was run 30 times in 2, 5 and 10 dimensional search spaces with the same initial positions and conditions to find global minimum point on the 8 benchmark functions frequently used in the literature. As a result of runs, the performances of the algorithms were evaluated based on the results of performance metrics such as the best cost, worst cost, accuracy, stability, time and standard deviation. Comparisons of the algorithms according to the cumulative mean performance values showed that the best performance was given by DE, followed by PSO. The CSO, which gives the closest results to the DE in terms of stability and accuracy, is best in terms of running time and is the third in the cumulative average performance evaluation. In the cumulative mean performance evaluation, the ICA was fourth, while the BBO, which gave the closest results to each function, was fifth. ABC, which produces far metric values from the other algorithms, is the last one in the performance evaluation used in this study and it shows the lowest cumulative mean performance value.

Keywords: Heuristic algorithms, Benchmark functions, Swarm intelligence, Performance metrics, Function optimization.

TEŐEKKÖR

Bu Y¼ksek lisans tez alıŐmamın gerekleŐmesinde deęerli bilgilerini benimle paylaŐan, kullandıęı her kelimenin ¼ęrenim hayatıma kattıęı ¼nemini asla unutmayacaęım saygıdeęer danıŐman hocam; Do. Dr. Cihan KARAKUZU'ya, alıŐmam boyunca benden bir an olsun yardımını esirgemeyen arkadaŐım ArŐ. Gör. Nesibe YALIN'a ve alıŐmam s¼resince t¼m zorlukları benimle beraber g¼ę¼sleyen ve hayatımın her evresinde bana destek olup yanımda olan aileme ve sevdiklerime sonsuz teŐekk¼rlerimi sunarım.

AyŐe BAŐTUę

22/05/2017

ETİK İLKE VE KURALLARA UYGUNLUK BEYANNAMESİ

Bu tezin bana ait, özgün bir çalışma olduğunu; çalışmamın hazırlık, veri toplama, analiz ve bilgilerin sunumu olmak üzere tüm aşamalarında bilimsel etik ilke ve kurallara uygun davrandığımı; bu çalışma kapsamında elde edilemeyen tüm veri ve bilgiler için kaynak gösterdiğimi ve bu kaynaklara kaynakçada yer verdiğimi; bu çalışmanın Anadolu Üniversitesi tarafından kullanılan “bilimsel intihal tespit programı’yla tarandığını ve hiçbir şekilde “intihal içermediğini” beyan ederim. Herhangi bir zamanda, çalışmamla ilgili yaptığım bu beyana aykırı bir durumun saptanması durumunda, ortaya çıkacak tüm ahlaki ve hukuki sonuçlara razı olduğumu bildiririm.

Ayşe BAŞTUĞ

İÇİNDEKİLER

	<u>Sayfa</u>
BAŞLIK SAYFASI	i
JÜRİ VE ENSTİTÜ ONAYI.....	ii
ÖZET	iii
ABSTRACT.....	iv
TEŞEKKÜR	v
ETİK İLKE VE KURALLARA UYGUNLUK BEYANNAMESİ.....	vi
İÇİNDEKİLER	vii
TABLolar DİZİNİ.....	ix
ŞEKİLLER DİZİNİ.....	x
SİMGELER DİZİNİ.....	xii
KISALTMALAR DİZİNİ	xiii
1. GİRİŞ.....	1
2. SEZGİSEL ALGORİTMALAR.....	3
2.1. Yapay Arı Kolonisi Algoritması (ABC).....	3
2.2. Biyocoğrafya Tabanlı Optimizasyon (BBO)	4
2.3. Guguk Kuşu Optimizasyonu Algoritması (CSO)	6
2.4. Farksal Gelişim (DE) Algoritması.....	8
2.5. Yayılmacı Rekabetçi Algoritma (ICA)	10
2.6. Parçacık Sürü Optimizasyonu (PSO) Algoritması	11
3. DENEKTAŞI İŞLEVLER.....	13
3.1. Ackley.....	13

3.2. Alpine	14
3.3. Bohachevsky	14
3.4. Griewank	15
3.5. Michalewicz	16
3.6. Rastrigin	17
3.7. Rosenbrock.....	18
3.8. Sphere	19
4. SEZGİSEL ALGORİTMALARIN DENEKTAŞI İŞLEVLER ÜZERİNDE BAŞARIM METRİKLERİYLE KARŞILAŞTIRILMASI	21
4.1. Karşılaştırma İçin Kullanılan Başarım Ölçüm Metrikleri.....	21
4.1.1. En iyi ölçüt metriği	21
4.1.2. En kötü ölçüt metriği.....	22
4.1.3. Doğruluk metriği	22
4.1.4. Kararlılık metriği	22
4.1.5. Zaman metriği.....	22
4.1.6. Standart sapma metriği.....	23
4.2. Denektaşı İşlevler İçin Karşılaştırma	23
4.2.1. Ackley işlevi için karşılaştırma.....	23
4.2.2. Alpine işlevi için karşılaştırma	26
4.2.3. Bohachevsky işlevi için karşılaştırma	29
4.2.4. Griewank işlevi için karşılaştırma	32
4.2.5. Michalewicz işlevi için karşılaştırma	35
4.2.6. Rastrigin işlevi için karşılaştırma	38
4.2.7. Rosenbrock işlevi için karşılaştırma	41
4.2.8. Sphere işlevi için karşılaştırma	44
5. SONUÇLAR VE ÖNERİLER.....	48
KAYNAKÇA.....	50
ÖZGEÇMİŞ	

TABLULAR DİZİNİ

Sayfa

Tablo 4.1. Ackley işlevi için elde edilen metrik değerleri tablosu.....	25
Tablo 4.2. Alpine işlevi için elde edilen metrik değerler tablosu	28
Tablo 4.3. Bohachevsky işlevi için elde edilen metrik değerler tablosu.....	31
Tablo 4.4. Griewank işlevi için elde edilen metrik değerler tablosu	34
Tablo 4.5. Michalewicz işlevi için elde edilen metrik değerler tablosu.....	37
Tablo 4.6. Rastrigin işlevi için elde edilen metrik değerler tablosu.....	40
Tablo 4.7. Rosenbrock işlevi için elde edilen metrik değerler tablosu	43
Tablo 4.8. Sphere işlevi için elde edilen metrik değerler tablosu	46
Tablo 5.1. Algoritmaların denektaşı işlevler için hesaplanan ortalama sıralama değerler tablosu	49

ŞEKİLLER DİZİNİ

Sayfa

Şekil 3.1. Ackley işlevinin $n=2$ için tanımladığı yüzey.....	13
Şekil 3.2. Alpine işlevinin $n=2$ için tanımladığı yüzey	14
Şekil 3.3. Bohachevsky işlevinin $n=2$ için tanımladığı yüzey.....	15
Şekil 3.4. Griewank işlevinin $n=2$ için tanımladığı yüzey	16
Şekil 3.5. Michalewicz işlevinin $n=2$ için tanımladığı yüzey.....	17
Şekil 3.6. Rastrigin işlevinin $n=2$ için tanımladığı yüzey	18
Şekil 3.7. Rosenbrock işlevinin $n=2$ için tanımladığı yüzey	19
Şekil 3.8. Sphere işlevinin $n=2$ için tanımladığı yüzey	20
Şekil 4.1. Ackley işlevi için algoritmaların 30 koşma boyunca elde ettiği ortalama en iyi ölçüt değerleri değişimi (sağ sütun), en iyi maliyet değerini veren koşma için en iyi ölçüt değerleri değişimi (sol sütun).....	26
Şekil 4.2. Alpine işlevi için algoritmaların 30 koşma boyunca elde ettiği ortalama en iyi ölçüt değerleri değişimi (sağ sütun), en iyi maliyet değerini veren koşma için en iyi ölçüt değerleri değişimi (sol sütun).....	29
Şekil 4.3. Bohachevsky işlevi için algoritmaların 30 koşma boyunca elde ettiği ortalama en iyi ölçüt değerleri değişimi (sağ sütun), en iyi maliyet değerini veren koşma için en iyi ölçüt değerleri değişimi (sol sütun)	32
Şekil 4.4. Griewank işlevi için algoritmaların 30 koşma boyunca elde ettiği ortalama en iyi ölçüt değerleri değişimi (sağ sütun), en iyi maliyet değerini veren koşma için en iyi ölçüt değerleri değişimi (sol sütun).....	35
Şekil 4.5. Michalewicz işlevi için algoritmaların 30 koşma boyunca elde ettiği ortalama en iyi ölçüt değerleri değişimi (sağ sütun), en iyi maliyet değerini veren koşma için en iyi ölçüt değerleri değişimi (sol sütun)	38
Şekil 4.6. Rastrigin işlevi için algoritmaların 30 koşma boyunca elde ettiği ortalama en iyi ölçüt değerleri değişimi (sağ sütun), en iyi maliyet değerini veren koşma için en iyi ölçüt değerleri değişimi (sol sütun).....	41

Şekil 4.7. Rosenbrock işlevi için algoritmaların 30 koşma boyunca elde ettiği ortalama en iyi ölçüt değerleri değişimi (sağ sütun), en iyi maliyet değerini veren koşma için en iyi ölçüt değerleri değişimi (sol sütun)	44
Şekil 4.8. Sphere işlevi için algoritmaların 30 koşma boyunca elde ettiği ortalama en iyi ölçüt değerleri değişimi (sağ sütun), en iyi maliyet değerini veren koşma için en iyi ölçüt değerleri değişimi (sol sütun).....	47
Şekil 5.1. Algoritmaların denektaş1 işlevler için hesaplanan ortalama sıralama değerler bar grafikleri	49

SİMGELER DİZİNİ

$accuracy_{f,A}^{(t)}$: A algoritmasının t.iterasyonunda elde edilen doğruluk değeri
$Acc_{f,t}$: A algoritmasının ortalama doğruluk değeri
A	: Optimizasyon algoritması
C_1, C_2	: Öğrenme faktörü
CR	: Çaprazlama değeri
d	: Vadilerin veya kenarların dikliği
E	: Maksimum λ
f	: Uygunluk fonksiyonu
F	: Mutasyon faktörü
g	: Nesil numarası
$H(u)$: Heaviside adım fonksiyonu
I	: Maksimum μ
i	: Birey sayısı
j	: İterasyon sayısı
k	: İterasyon sayısı
K	: Atalet değeri
m	: Mutasyon oranı
$Min^{(t)}$: Arama uzayında elde edilen en kötü değer
$Max^{(t)}$: Arama uzayında elde edilen en iyi değer
n	: Popülasyon boyutu
N	: Sürü büyüklüğü
P_{mod}	: Adaya ait değişiklik sayısı
r	: 0 ile 1 arasında üretilen rastgele bir sayı
s	: Adım büyüklüğü
$Stab_{f,t}^A$: A algoritmasının t.iterasyonunda elde edilen kararlılık değeri
t	: İterasyon dizisi
U_i	: Deneme birey
V_i	: Mutant birey
V_i^j	: j.nesildeki i.mutant birey
V_i^k	: k. iterasyondaki i. parçacığın hız değeri
V_i^{k+1}	: k+1. iterasyondaki i. parçacığın hız değeri
x_i^k	: k. iterasyondaki i. parçacığın konum değeri
x_i^{k+1}	: k+1. iterasyondaki i. parçacığın konum değeri
x_j^t, x_k^t	: Rassal permütasyonla seçilmiş iki farklı çözüm
X_{r1}	: Mevcut birinci birey
X_{r2}	: Mevcut ikinci birey
X_{r3}	: Mevcut üçüncü birey
α	: Ölçekleme faktörü
ϵ	: Düzgün dağılımdan gelen rastgele bir sayı
σ	: Standart sapma
μ	: Göç oranı
λ	: Göçmenlik oranı
$\Gamma(\lambda)$: Standart gamma fonksiyonu

KISALTMALAR DİZİNİ

ABC	: Artificial Bee Colony, Yapay Arı Kolonisi
BBO	: Biogeography-based Optimization, Biyocoğrafya Tabanlı Optimizasyon
DE	: Differential Evolution Algorithm, Farksal Gelişim Algoritması
CSO	: Cuckoo Search Optimization, Guguk Kuşu Optimizasyonu
ICA	: Imperialist Competitive Algorithm, Yayılmacı Rekabetçi Algoritma
PSO	: Particle Swarm Optimization, Parçacık Sürü Optimizasyonu

1. GİRİŞ

Bilgi çağı olan günümüzde insanların bilgi ve iletişim teknolojilerini kullanım oranları her geçen gün artmaktadır. İnsanların çözmekte zorlandıkları veya uzun sürede çözebilecekleri problemler bilgisayarların kullanımı ile çok kısa sürede halledilebilmektedir. Teknolojik ilerlemelerle tıpkı bir insan gibi düşünüp tepkiler verebilen, kendini geliştirebilen sistemler ortaya çıkmıştır. Tam da burada yapay zeka ve optimizasyon olguları karşımıza çıkmaktadır.

Optimizasyon, bir problemin çözümünde belirli kısıtlar altında minimizasyon ya da maksimizasyon yaparak en iyisini bulma çalışmasıdır. Gerçek hayatta kısıtlamaların tümünü aynı anda karşılayabilmek güç bir durum olduğundan tüm kısıtlamaları aynı zamanda sağlayabilmek ve çözümlerin en iyisini olduğunu anlayabilmek amacıyla da optimizasyon teknikleri geliştirilmiştir (Özdemir, 2013, s. 1-2).

Optimizasyon işleminde kullanılan tekniklerden olan klasik yöntemler en iyi çözümü başarıyla elde edebilmekte ancak süre ve maliyet açısından büyük bir külfete neden olmaktadır. Bu yüzden zaman ve maliyeti en aza indirgeyen, canlıların doğal yaşamından ilham alınarak sezgisel algoritmalar geliştirilmiştir.

Sezgisel algoritmaları, klasik algoritmalarından ayıran en önemli farkı seçim yapma yeteneği olmasıdır. Bu özelliği ona en iyi çözümü seçme ve rastgelelik sağlar. Problemler uygun süre dilimi içerisinde, en iyiye yakın fakat her zaman aynı çözüm üretilmeden düşük bellek ihtiyacı ile çözülür. Problemlere uygulanabilirliği daha kolay olduğu için de klasik yöntemlerden çok daha fazla tercih edilmekte olup yüksek başarı oranları elde edilmektedir (Akça, 2011, s. 9-10).

Sezgisel teknikler; fizik tabanlı, biyoloji tabanlı, sürü tabanlı, sosyal tabanlı, kimya tabanlı ve müzik tabanlı olmak üzere altı gruba ayrılmaktadır. Sürü tabanlı yöntemde birbiriyle etkileşim halinde bulunan bireyler bulunur. Belirli sayıdaki birey bir amacı elde etmek için basit kurallar kullanıp birlikte çalışarak kolektif bir zekâ oluşturur (Akyol and Alataş, 2012, s. 37).

Çok sayıda sezgisel algoritma türleri bulunmakta olup gün geçtikçe yeni algoritmalar da türetilmektedir. Bu çalışmada bilimsel camiada yaygın olarak kullanılması, kodlarının açık kaynak kodlu olması gibi özelliklerinden dolayı sürü tabanlı yapay arı kolonisi, biyocoğrafya tabanlı optimizasyon, guguk kuşu arama algoritması, farksal gelişim algoritması, yayılcı rekabetçi algoritma ve parçacık sürü algoritması kullanılmıştır.

Çalışma kapsamında ikinci bölümde kullanmayı tercih ettiğimiz sezgisel algoritmaların her biri ayrı ayrı incelenmektedir. Üçüncü bölümde literatürde üzerinde çokça çalışılan 8 denektaşı işlevden (test fonksiyonları) bahsedilmektedir. Dördüncü bölümde seçilen sezgisel algoritmaların, denektaşı işlevler üzerinde farklı tasarım metrikleriyle karşılaştırılması ve elde edilen sonuçlarla başarımları irdelenmiştir. Son olarak beşinci bölümde ise tüm bu yapılan çalışmalar sonucunda varılan yargılardan söz edilmiştir.

2. SEZGİSEL ALGORİTMALAR

Alışlagelmiş optimizasyon yöntemleriyle çözülmesi zor problemlerin çözümünde sezgisel yaklaşımların uygulandığı algoritmalar sezgisel algoritmalar olarak adlandırılır. Sezgisel optimizasyon algoritmaları; biyolojik tabanlı, fizik tabanlı, sürü tabanlı, sosyal tabanlı, müzik tabanlı ve kimya tabanlı olmak üzere 6 gruba ayrılmıştır. Bu çalışmada kullanılan altı farklı algoritma üzerinde çalışılmıştır. Her bir algoritmanın tanımları ve kaba kodları aşağıdaki alt başlıklarda verilmiştir.

2.1. Yapay Arı Kolonisi Algoritması (ABC)

Yapay arı kolonisi algoritması, Derviş Karaboğa tarafından 2005 yılında, bal arılarının nektar bulmadaki zeki davranışlarından esinlenilerek geliştirilmiştir. Bal arılarında sürü kolektif zekâyı ortaya çıkaran, besin kaynağı seçimi modeli üç bileşenden oluşmaktadır. Besin kaynağı, çalışan arılar ve çalışmayan arılardan oluşan bu seçim modelinde süreç kaynak bulma, kaynak seçme ve kaynağı terk etme ile gerçekleşmektedir (Karaboga, 2005, s. 1-3).

Bir besin kaynağının kalitesi; yuvaya yakınlığı, zenginliği, enerji yoğunluğu ve bu enerjiyi çıkarmanın kolaylığı gibi birçok faktöre bağlıdır.

Çok boyutlu arama uzayında kaynakları aramaya çıkan işçi arılardır. Her bir kaynaktan nektarlar toplarlar. Besin kaynağının uzaklığı, yönü ve nektarların kalitesi hakkında hafızasında tuttuğu bilgileri sergiledikleri farklı danslar ile gözcü arılara iletirler. Daha sonrasında bu arılar üç farklı davranıştan birini sergilerler. Ya kaynağa gidip işçi arı olup nektar toplarlar, ya yeni kaynaklar aramak için kâşif arı olurlar ya da kovanda durup gözcü arı olarak bilgileri değerlendirip kaynak seçimi yaparlar. Algoritma kabullenmelerine göre işçi arı ile gözcü arıların sayıları eşittir. Her bir besin kaynağı olası mümkün çözümleri, kaynaklardan toplanan nektarın kalitesi de bu olası çözümlerin başarımlarını değerlendirmektedir (Akça, 2011, s. 21-24).

Gözcü arılar, işçi arıların danslar ile kendisine iletmiş bilgilerden yola çıkarak belirli olasılık hesaplamaları ile kaliteli yiyecek kaynağını seçerler. Daha sonra seçtikleri bu kaynağa giderek işçi arıların yaptıklarını yani nektar toplama işlemini yaparlar. Kâşif arılar ise yeni rastgele besin kaynakları arayarak nektar depolama işlemini gerçekleştiren arılardır (Babayiğit and Özdemir, 2012, s. 618-619).

Yapay Arı Algoritmasının Kaba (Sözde) Kodu:

Begin

Popülasyonu başlat

While (*iterasyon sonlandı mı*)

Bölgesel arama için konumları belirle

İşçi arılar kaynaktan nektarları toplasın

Gözcü arılar uygunluk hesabı yaparak nektarların kalitesini değerlendirsin

Kaşif arılar rastgele yeni kaynaklar arasın

Yeni kaynakların uygunluğu değerlendirilsin

En iyi çözümler güncellensin

end

end

2.2. Biyocoğrafya Tabanlı Optimizasyon (BBO)

Dan Simon tarafından 2008 yılında geliştirilen popülasyon tabanlı, genetik algoritma ve parçacık sürü algoritması gibi biyolojik tabanlı optimizasyonlarla benzer özelliklere sahip bir algoritmadır. Ancak biyoloji tabanlı optimizasyon algoritmalarından farklı tek ve eşsiz özelliklere de sahiptir.

Biyocoğrafyanın matematiksel modeline göre çözüm adasındaki türler bir adadan başka bir adaya göçü esnasında bazı türlerin nesli tükenirken yeni türler de ortaya çıkar. Her çözüm adasının yağış, bitki örtüsü çeşitliliği, topografik özellikler, sıcaklık gibi habitat yaşam alanı değerlerine bağımlı olan habitat yaşam uygunluğu endeksi vardır (HIS). Ayrıca her çözüm adasını bir de yaşanılabilirliği karakterize eden yaşam alanından bağımsız, uygunluk endeksi değişkenine (SIV) sahiptir (Simon, 2008, s. 702). Yaşam ortamı H , rastgele başlayan N 'ye (SIVs) ait bir vektör olup istenilen konumdaki adaya ulaşmak için göç ve mutasyon adımlarını kullanılır. Göç oranı μ ve göçmenlik oranı λ bilgileri her bir yaşam ortamına iletilir (Saraçoğlu vd., 2013, s. 50). Her bireyde var olan μ ve λ eşitlikleri K türleri sayısının fonksiyonu olarak eşitlik 2.1'de ve eşitlik 2.2'de verilmiştir. E maksimum λ , I maksimum μ ve n popülasyon boyutudur.

$$\lambda_k = \frac{E_k}{n} \quad (2.1)$$

$$\mu_k = I \left(1 - \frac{K}{n}\right) \quad (2.2)$$

σ standart sapma olup, habitata ait göç modifikasyonu, aşağıdaki gibi tanımlanmıştır (Simon, 2008, s. 704-706).

```

 $\alpha$  ve  $\lambda_i$  olasılığı ile  $H_i$  seç
if  $H_i$  seçilmiş ise
  for  $j = 1:n$ 
     $\alpha$  ve  $\lambda_i$  olasılığı ile  $H_j$  seç
    if  $H_j$  seçilmiş ise
       $H_j$  içerisinde SIV için rastgele  $\sigma$  seç
       $\sigma$  ile  $H_i$  içerisinde rastgele bir SIV değiştir
    end if
  end for
end if

```

Habitat göç algoritmasına göre görüyoruz ki göç operatörü, çözümler arasındaki bilgiyi etkili bir şekilde paylaşmaktadır. Ancak habitat mutasyon algoritmasında görülecek olan rastgele mutasyon nedeniyle arama yeteneği zayıflayabilir (Gong vd., 2010, s. 2750). Popülasyon çeşitliliğini artırmak için kullanılan mutasyon oranı m eşitlik 2.3 ile ifade edilir.

$$m(s) = m_{\max} \left(1 - \frac{P_i}{P_{\max}}\right) \quad (2.3)$$

m_{\max} kullanıcının tanımladığı bir parametredir.

P_{mod} Adaya ait değişiklik sayısı olup habitat mutasyon işlemi aşağıdaki gibi tanımlanmıştır (Simon, 2008, s. 706).

```

for  $j = 1:N$ 
   $P_i$  olasılığını,  $\lambda_i$  ve  $\mu_i$  kullanarak hesapla
   $\alpha$   $P_i$  olasılığı kullanarak SIV  $H_{i(j)}$  seç
  if  $H_{i(j)}$  seçilmiş ise
     $H_{i(j)}$  ile rastgele üretilen SIV'ı yer değiştir
  end if
end for

```

Yukarıda tanımlanan işlemler ile biyocoğrafya tabanlı optimizasyon algoritması aşağıdaki gibi tanımlanmıştır (Saraçoğlu vd., 2013, s. 50).

Begin

Rastgele bir popülasyon sayısı belirle

Popülasyon içerisindeki her bireyin uygunluğu denetle

while (*iterasyon sonlandı mı*)

En iyi yaşamları kaydet

λ, μ ve S türlerinin sayısını HSI eşleştir

Olasılıksal bir göç adası seç

Seçilen adaya dayanan rastgele seçilen SIV'ları farklı adalara göç ettir

Mutasyon algoritmasındaki popülasyonların en kötü yarılarını farklılaştır

Her birey için uygunluk kontrolü yap

Popülasyonu en iyiden en kötüye doğru sırala

end while

end

Biyocoğrafya tabanlı optimizasyon algoritması daha çok video kodlamada blok temelli hareket kodlama, renkli resim bölme, uydu görüntüsü sınıflandırılması, özellik seçimi gibi uygulamalarda kullanılmaktadır (Ammu, Sivakumar and Rejimoan, 2013, s. 159).

2.3. Guguk Kuşu Optimizasyonu Algoritması (CSO)

Guguk kuşu arama algoritması, bazı guguk kuşu türlerinin kuluçka parazitliği yapma temeline dayanan ve Levy uçuşunu kullanan 2009 yılında Xin-She Yang ve Suash Deb tarafından geliştirilen bir algoritmadır (Yang and Deb, 2009, s. 210-211).

Bazı guguk kuşu türleri gözetlediği farklı bir kuşun yuvasının sahibi yuvadan uzaklaştığında bu yuvaya gelir ve diğer yumurtalara çok benzeyen kendi yumurtasını diğer kuşun yumurtalarının içerisinde bırakır. Bu fazla yumurtanın fark edilmemesi amacıyla da yuvadaki diğer yumurtalardan birini alarak oradan uzaklaşır. Varlığı fark edilmeyen guguk kuşu yavrusu genellikle daha önce doğduğu üvey kardeşlerini farklı hareketlerle yuvadan uzaklaştırmaya çalışır. Üç hafta gibi kısa bir süre sonrasında yavru guguk kuşu, yuva sahibi kuştan daha büyük olmasına rağmen beslenmesi üvey annenin

bakımı ile sürdürülür. Altıncı haftanın sonunda yuvayı da bozarak kendine eş bakmaya çıkar. Eğer yuva sahibi kuş guguk yavrusunu fark ederse onu yuvadan atar veya o yuvayı terk eder (Anonim, 2012).

Guguk Kuşu Algoritması Sözde Kodu (Yang and Deb, 2009, s. 210-211) :

Begin

Amaç fonksiyonu $f(x)$, $x = (x_1, \dots, x_d)^T$ olsun.

n adet ev sahibi yuvanın $x_i (i = 1, 2, \dots, n)$ ilk popülasyonunu oluştur

while (iterasyon sonlandı mı)

Levy uçuşu yardımı ile rastgele guguk kuşu al

f_i uygunluk fonksiyonunu hesapla

n yuva arasından rastgele bir j yuvası seç

if ($f_i > f_j$),

j yeni çözüm yapılınsın

end if

Kötü yuvaların P_a kadarı yuvadan vazgeç ve yeni yuvalar oluştur

En iyi çözümleri hafızada tut (veya nitelikli çözüme sahip yuvaları hafızada tut)

Tüm çözümleri sırala ve o andaki en iyi çözümü bul

end while

Tüm sonuçları göster

end

CSO, P_a parametresi tarafından kontrol edilen yerel rassal yürüyüşün ve küresel keşfedici rassal yürüyüşün dengeli bir kombinasyonunu kullanır. Yerel rassal yürüyüş eşitlik 2.4 ile gerçekleştirilmektedir. x_j^t ve x_k^t rassal permütasyonla seçilmiş iki farklı çözümdür. $H(u)$ heaviside adım fonksiyonudur. ϵ düzgün dağılımdan gelen rastgele bir sayı olup s de adım büyüklüğüdür.

$$x_i^{t+1} = x_i^t + \alpha \cdot s \otimes H(P_a - \epsilon) \otimes (x_j^t - x_k^t) \quad (2.4)$$

Levy uçuşu eşitlik 2.5'te verilen küresel rassal yürüyüş ile gerçekleştirilir. λ , levy katsayısı olup sabit bir değerdir.

$$x_i^{t+1} = x_i^t + \alpha \cdot L.(s, \lambda) \quad (2.5)$$

Ve L fonksiyonu hesabı da eşitlik 2.6'teki gibidir.

$$L(s, \lambda) = \frac{\lambda \Gamma(\lambda) \sin(\pi\lambda/2)}{\pi} \cdot \frac{1}{s^{1+\lambda}} \quad (s \geq s_0 \geq 0) \quad (2.6)$$

Burada α , sıfırdan büyük olup ölçekleme faktörüdür. $\Gamma(\lambda)$, dağılımı sıfırdan büyük adımlarla olan standart gamma fonksiyonudur. Bazı durumlarda $\alpha = (L/10)$ kullanılırken, bazen de uzak uçuşların engellenmesi için $\alpha = 0(L/100)$ olarak kullanılır (Yang, 2014, s. 131; Karagül, 2014, s. 777-778).

Guguk kuşu algoritması mühendislik problemleri çözümünde, yapay sinir ağlarında, zaman planlamalarında, dağıtık sistem uygulamalarında kullanılmaktadır.

2.4. Farksal Gelişim (DE) Algoritması

DE, Price ve Storn (Price and Storn, 1997, s. 341-343) tarafından 1997 senesinde özellikle sürekli verilerin söz konusu olduğu optimizasyon problemlerinde etkili sonuçlar verebilen popülasyon temelli sezgisel algoritmadır. Algoritmanın temelinde, popülasyon içinden rasgele seçilen iki bireyin arasındaki farkın üçüncü bir bireye eklenmesi vardır. Her iterasyonda uygunluk fonksiyonu değerlerine bakılarak problem için en iyi olan çözümler ve bireyler araştırılmaktadır. Başlangıçta oluşturulan bireylere bir durdurma şartı sağlanıncaya kadar sırasıyla mutasyon, çaprazlama ve seçim işlemleri uygulanarak algoritmanın işleyişi tamamlanır.

Mutasyon işleminde, mevcut bireyin kromozomunun üzerindeki bazı genlerin bir miktar değiştirilmesiyle yeni bir mutant birey oluşturulur. Mutant birey (V_i) oluşturulurken, mevcut bireyden farklı olan üç birey (X_{r1} , X_{r2} , X_{r3}) daha seçilir. İki bireyin farkının mutasyon faktörü (F) oranı ile üçüncü bireyin genlerine etki edilir. Yani mutant birey oluşumu eşitlik 2.7 ile hesaplanır (Zorarpacı and Özel, 2004, s. 52).

$$V_i^j = X_{r3}^j + F \cdot (X_{r1}^j - X_{r2}^j) \quad (2.7)$$

Çaprazlama işleminde mevcut (X_i) ve mutant birey (V_i)'in çaprazlanması sonucunda deneme birey (U_i) oluşur. Çaprazlama değeri (CR) 0 ile 1 arasında seçilen bir değerdir. Çalışmamızda bu değer 0,2 olarak alınmıştır. r değeri ise 0 ile 1 arasında üretilen rastgele bir sayıdır. Bu değer CR'den küçükse deneme birey mutant bireyden, CR'den büyükse deneme birey mevcut bireyden oluşturulur. r değeri, mutant bireylerden (V_i , $g+1$) en azından bir sonraki nesile (U_i , $g+1$) birey üretilmesini

garantiler. Çaprazlama yaparak yeni birey oluşumu eşitlik 2.8 ile hesaplanır, eşitlikteki g nesil numarasını gösterir (Eser and Yüzgeç, 2014, s. 203-204).

$$U_{i, g + 1} = \begin{cases} V_{i, g + 1}, & \text{eğer } r \leq CR \\ X_{i, g}, & \text{eğer } r > CR \end{cases} \quad (2.8)$$

Seçim işleminde, mevcut birey ile yeni oluşturulan deneme bireylerin uygunluk fonksiyonu (f) değerine bakılarak, sonraki jenerasyon için seçim işlemi yapılır. Uygunluk değeri iyi olan birey tercih edilir. Seçim işlemi eşitlik 2.9'da görüldüğü gibi yapılır (Keskintürk, 2006, s.91).

$$x_{i, g + 1} = \begin{cases} x_{U, g + 1}, & \text{eğer } f(x_{U, g + 1}) \leq f(x_{i, g}) \\ x_{i, g}, & \text{diğer durumlar} \end{cases} \quad (2.9)$$

DE için algoritma sözde kodu aşağıdaki gibi verilebilir.

Begin

Popülasyonu başlat

While (*iterasyon sonlandı mı*)

Yeni populasyonu sıfırla

Mutasyon hesapla

Çaprazlama yap

Uygunluk değeri hesapla

if *uygunluk(yeni_birey) ≤ uygunluk(mevcut_birey)*

yeni bireyi popülasyona kat

else

mevcut bireyi popülasyona kat

end if

yeni popülasyonu, populasyona aktar

Uygunluk değeri en iyi olan bireyi bul

En iyi bireyin uygunluk değerini hesapla

end while

end

2.5. Yayılmacı Rekabetçi Algoritma (ICA)

ICA, Atash Gargari ve Lucas tarafından 2007 senesinde ülkelerin güçlü olabilmek için birbirleri ile olan mücadelecı yarışını konu alan sezgisel optimizasyon algoritmasıdır.

Başlangıç popülasyonundaki ülkelerin içerisinde en iyileri seçilerek emperyalist yapılır ve kalan diğer ülkeler emperyalistlerin kolonisi yapılarak dağıtım gerçekleştirilir. Koloniler en uygun emperyalistlere yönelmeye çalışır. Her bir imparatorluğun gücü emperyalistin ve emperyaliste bağlı koloninin toplam gücüne bağlıdır. Algoritma işleyişi boyunca emperyalistler arasındaki yarış sürer. Başarılı olamayan emperyalistler elenir. Yarış esnasında güçlü imparatorlukların güçleri artarken, zayıf imparatorlukların güçleri azalarak yıkılma sürecine girer. Tek bir imparator ve ona bağlı diğer ülkelerden oluşan tek dünya modeli oluşana kadar yarış devam eder (Gargari and Lucas, 2007, s. 4662; Altunbey and Alataş, 2015, s. 37).

ICA için algoritma sözde kodu aşağıdaki gibi gösterilebilir (Xu vd., 2014, s. 231).

Begin

İmparatorlukları baslat

while(durdurma kriteri)

Emperyalistlere koloni dağıt (Asimilasyon)

Kolonilerin bir bölümünü seçip daha sonra bunları eşit sayıda yeni üretilen ülkeler ile değiştir.

if (koloni emperyalistten daha iyi)

Koloni ve emperyalistin rollerini değiştir.

end if

for(her bir imparator için)

Toplam maliyet hesapla

end for

Zayıf imparatorluktan zayıf kolonileri at. Diğer imparatorlar bu koloni için savaşsın.

if (imparatorluğun kolonisi yok)

İmparatorluğu yık

end if

end while

end

2.6. Parçacık Sürü Optimizasyonu (PSO) Algoritması

Balık, böcek, kuş gibi sürüleri ile birlikte hareket eden canlılardan ilham alınarak Kennedy ve Eberhart (1995, s. 1943-1945) geliştirilen bir sezgisel optimizasyon yöntemidir. Kuşlar yiyecek kaynaklarının yerini bilmedikleri için, kaynağa yakınlığı fazla olan diğer kuşları takip ederek ulaşmaya çalışırlar. Problemin çözüm uzayı bu besin arayışları esnasındaki konumlarıdır. Sürü popülasyonundaki her bir bireye parçacık denir. Popülasyon içerisindeki her bir parçacık diğer parçacıklarla bağlantı içerisinde olur. Her parçacık arama uzayında bulunduğu konumları, ne kadar hızla ilerlediği, tüm iterasyonlar boyunca elde ettiği en iyi uygunluk değerini ve hangi konumda iken bu uygunluk değerini elde ettiği gibi bilgileri hafızada tutmalıdır. Çünkü her boyutta yapacağı hız ve yön değişimleri, kendisinin ve diğer parçaların daha önce hafızada tuttuğu en iyi konumların birleşiminden oluşacaktır (Tamer and Karakuzu, 2006, s. 302-303).

PSO popülasyonu başlangıçta rastgele konumlarla (parçacıklarla) başlatılır, birey ve sürünün hafızasında tuttuğu en iyi konumları kullanarak güncellemeler yapılarak en uygun çözümü elde edilmeye çalışılır. Her iterasyonda her bir parçacığın konumu bir önceki konumuna ve diğer parçacıkların konumlarına bağlı olarak güncellenir. Parçacıkların kendi daha önceki konumlarına bakması algoritmanın mantıksal yönünü, diğer parçacıkların en iyi konumlarına bakması da algoritmanın sosyal yönünü gösterir. Bir parçacığın geçerli iterasyona kadar en iyi uygunluk değerini veren pozisyonu parçacığın yerel en iyisi 'pbest' olarak isimlendirilir ve tüm sürünün geçerli iterasyona kadar en iyi uygunluk değerini veren parçacık küresel en iyi parçacık 'gbest' olarak isimlendirilir. Bu yerel ve küresel en iyi parçacıklar hafızada tutulur. Son iterasyonda beirlenen küresel en iyi parçacık problemin çözümü olarak atanır. Parçacıklara ait hız ve konum formülasyonları eşitlik 2.10 ve 2.11'de verildiği gibidir.

$$V_i^{k+1} = K(V_i^k + C_1 rand() (p_{best}^k - x_i^k) + C_2 rand() (g_{best} - x_i^k)) \quad (2.10)$$

$$x_i^{k+1} = x_i^k + V_i^{k+1} \quad (2.11)$$

İterasyon sayısı k 'dır. k . iterasyondaki i . parçacığın hız değeri V_i^k , $k+1$. iterasyondaki i . parçacığın hız değeri V_i^{k+1} 'dir. C_1 parçacıkların kendi önceki

deneyimlerini öğrenme faktörü, C_2 tüm parçacıkların deneyimlerini öğrenme faktörüdür. C_1 ve C_2 'ye, genelde 2'ye yakın değerler verilir. Çalışmamızda $C_1=1,5$ ve $C_2=2$ olarak alınmıştır. $rand$ ile 0 ile 1 arasında rastgele bir sayı üretilir. K da atalet değeridir. Hız değeri bulunduğundan sonra konum hesabı yapılır. k . iterasyondaki i . parçacığın konum değeri x_i^k , $k+1$. iterasyondaki i . parçacığın konum değeri x_i^{k+1} 'dir (Tamer and Karakuzu, 2006, s. 303).

PSO için algoritma adımları aşağıdaki gibi özetlenebilir.

Begin

for

Her parçacığı rastgele konumda başlat.

end for

do

for

Parçacıkların en uygun değerlerini hesapla

if yeni uygunluk değeri < pbest konumundaki uygunluk değeri

Yeni konumu pbest yap

end if

Parçacıkların içerisindeki en iyi pbest değerini gbest yap

for

Her parçacığın hızını hesapla

Her parçacığın konumunu güncelle

end for

while (*iterasyon sonlandı mı*)

end

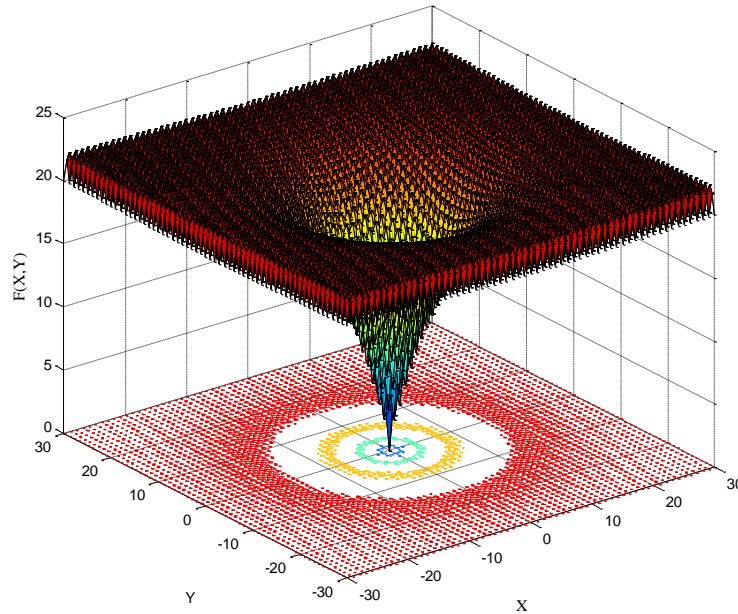
3. DENEKTAŞI İŞLEVLER

Denektaş işlevler optimizasyon algoritmalarının farklı özelliklerini test etmek amacıyla literatürdeki çalışmalarda kullanılmaktadır. Birçok denektaş işlev mevcut olmasına rağmen bu çalışmada deneme amaçlı, problemlerde en çok tercih edilen 8 adet tek veya çok çukurlu işlevler seçilmiştir. Her birinin denklemi, arama uzayı, küresel minimumu ve iki boyutlu uzay için grafiksel çizimi gibi bilgileri alt başlıklar altında özetlenmiştir.

3.1. Ackley

İşlev eşitlik 3.1 (Molga and Smutnicki, 2005, s. 15) ile tanımlanmakta olup çok çukurlu engebeli bir fonksiyondur. Eşitlikte n boyut sayısını göstermekte olup, bu çalışmada $-30 \leq x_i \leq 30, i = 1, 2, \dots, n$ aralığındaki arama uzayı kullanılmıştır. İşlevin $n=2$ için yüzeyi Şekil 3.1’de verilmiştir. Şekilden de görüleceği üzere işlevin küresel minimumu $x_1=x_2=\dots=x_n=0$ pozisyonunda 0 değerindedir. Tavsiye edilen değerler a için 20, b için 0,2, c için 2π ’dir

$$f(x) = -a \exp \left(-b \sqrt{\frac{1}{d} \sum_{i=1}^n x_i^2} \right) - \exp \left(-\frac{1}{d} \sum_{i=1}^n \cos(cx_i) \right) + a + \exp(1) \quad (3.1)$$

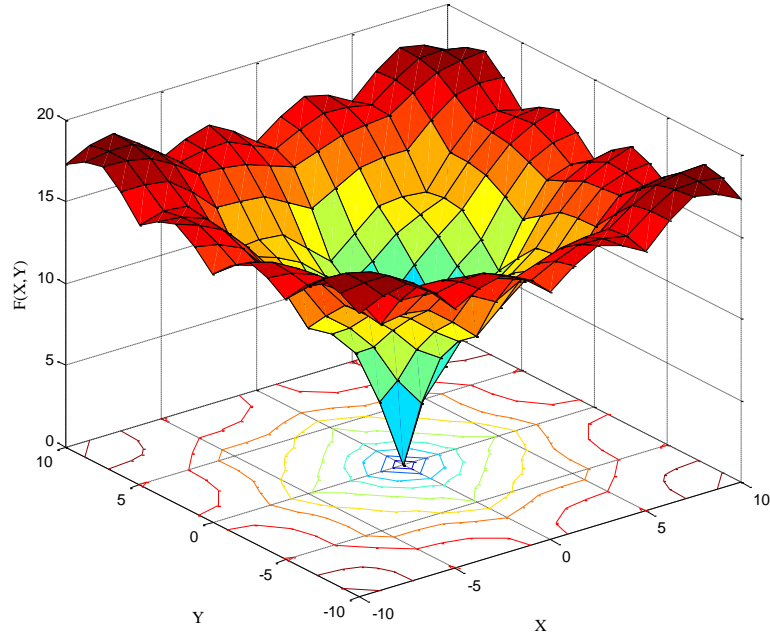


Şekil 3.1. Ackley işlevinin $n=2$ için tanımladığı yüzey

3.2. Alpine

İşlev eşitlik 3.2 (Rahnamayan, Tizhoosh and Salama, 2007, s. 1613) ile tanımlanmakta olup çok çukurlu Ackley fonksiyonuna göre daha az engebeli bir fonksiyondur. Eşitlikte n boyut sayısını göstermekte olup, bu çalışmada $-10 \leq x_i \leq 10$, $i = 1, 2, \dots, n$ aralığındaki arama uzayı kullanılmıştır. İşlevin $n=2$ için yüzeyi Şekil 3.2’de verilmiştir. Şekilden de görüleceği üzere işlevin küresel minimumu $x_1=x_2=\dots=x_n=0$ pozisyonunda 0 değerindedir.

$$f(x) = \sum_{i=1}^n |x_i \sin(x_i) + 0,1x_i| \quad (3.2)$$

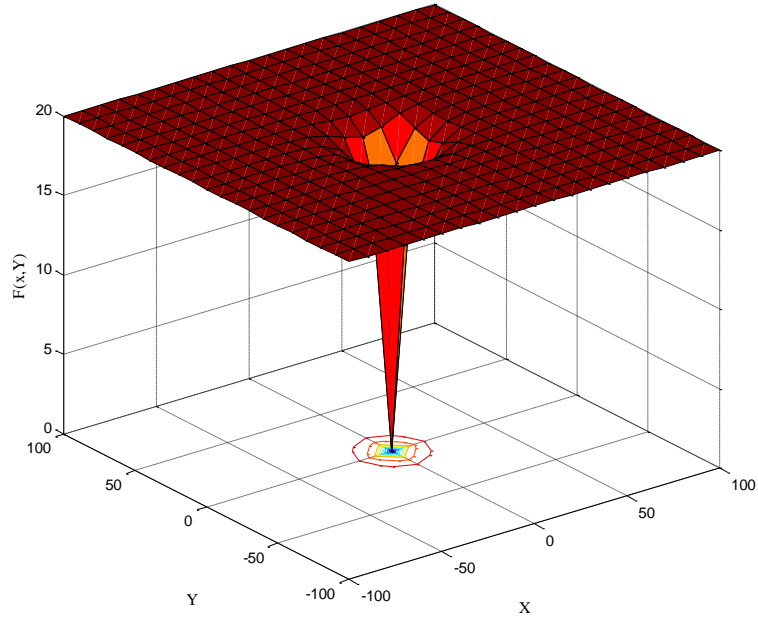


Şekil 3.2. Alpine işlevinin $n=2$ için tanımladığı yüzey

3.3. Bohachevsky

İşlev eşitlik 3.3 (Jamil and Yang, 2013, s. 158) ile tanımlanmakta olup çok çukurlu engebeleri az bir fonksiyondur. Bu çalışmada $-100 \leq x_i \leq 100$, $i = 1, 2, \dots, n$ aralığındaki arama uzayı kullanılmıştır. İşlevin iki boyutlu arama uzayındaki yüzeyi Şekil 3.3’te verilmiştir. Şekilden de görüleceği üzere işlevin küresel minimumu $x_1=x_2=\dots=x_n=0$ pozisyonunda 0 değerindedir.

$$f(x) = x_1^2 + 2x_2^2 - 0,3 \cos(3\pi x_1) - 0,4 \cos(4\pi x_2) + 0,7 \quad (3.3)$$

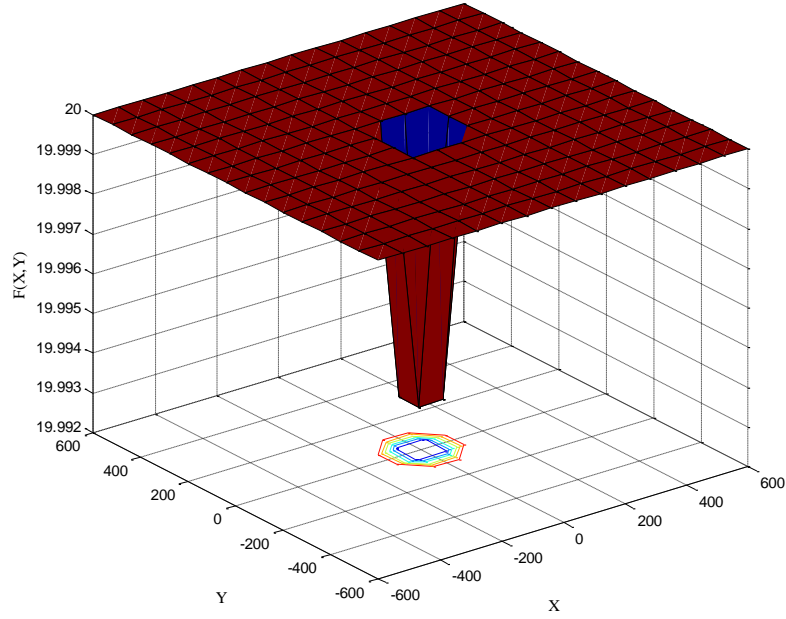


Şekil 3.3. Bohachevsky işlevinin $n=2$ için tanımladığı yüzey

3.4. Griewank

İşlev eşitlik 3.4 (Molga and Smutnicki, 2005, s. 11) ile tanımlanmakta olup çok çukurlu geniş dağılmış minimum noktalara bir fonksiyondur. Eşitlikte n boyut sayısını göstermekte olup, bu çalışmada $-600 \leq x_i \leq 600$, $i = 1, 2, \dots, n$ aralığındaki arama uzayı kullanılmıştır. İşlevin $n=2$ için yüzeyi Şekil 3.4'te verilmiştir. Şekilden de görüleceği üzere işlevin küresel minimumu $x_1=x_2=\dots=x_n=0$ pozisyonunda 0 değerindedir.

$$f(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \quad (3.4)$$

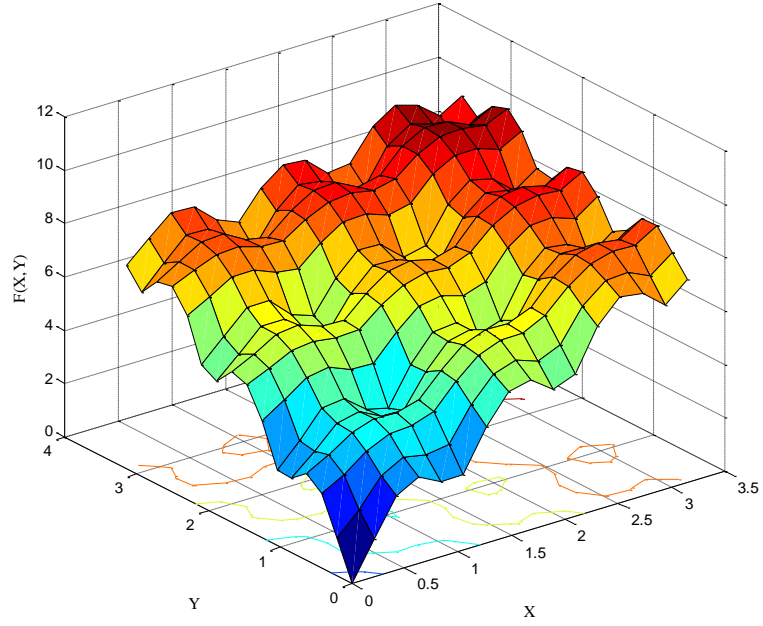


Şekil 3.4. Griewank işlevinin $n=2$ için tanımladığı yüzey

3.5. Michalewicz

İşlev eşitlik 3.5 (Molga and Smutnicki, 2005, s. 20) ile tanımlanmakta olup çok çukurlu $n!$ yerel optimum noktaya sahip bir fonksiyondur. Eşitlikte d vadilerin veya kenarların dikliğini, n boyut sayısını göstermekte olup, bu çalışmada $0 \leq x_i \leq \pi$, $i = 1, 2, \dots, n$ aralığındaki arama uzayı kullanılmıştır. d değeri büyüdükçe arama yapmak zorlaşmaktadır. İşlevin $n=2$ için yüzeyi Şekil 3.5'te verilmiştir. İşlevin küresel minimumu 2 boyutlu ise -1,8013, 5 boyutlu ise -4,6876, 10 boyutlu ise -9,6601'dir.

$$f(x) = - \sum_{j=1}^n \sin(x_j) \left[\sin\left(\frac{jx_j^2}{\pi}\right) \right]^{2d}, d = 10 \quad (3.5)$$

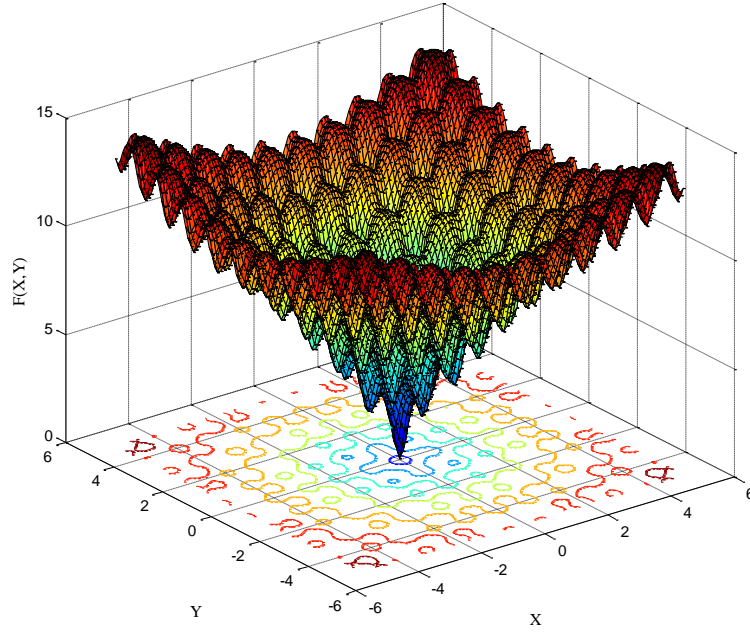


Şekil 3.5. Michalewicz işlevinin $n=2$ için tanımladığı yüzey

3.6. Rastrigin

İşlev eşitlik 3.6 (Rahnamayan, Tizhoosh and Salama, 2007, s. 1612) ile tanımlanmakta olup çok çukurlu çok engebeli zor bir yüzeye sahip bir fonksiyondur. Eşitlikte n boyut sayısını göstermekte olup, bu çalışmada $-5,12 \leq x_i \leq 5,12$, $i = 1, 2, \dots, n$ aralığındaki arama uzayı kullanılmıştır. İşlevin $n=2$ için yüzeyi Şekil 3.6'da verilmiştir. Şekilden de görüleceği üzere işlevin küresel minimumu $x_1=x_2=\dots=x_n=0$ pozisyonunda 0 değerindedir.

$$f(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10] \quad (3.6)$$

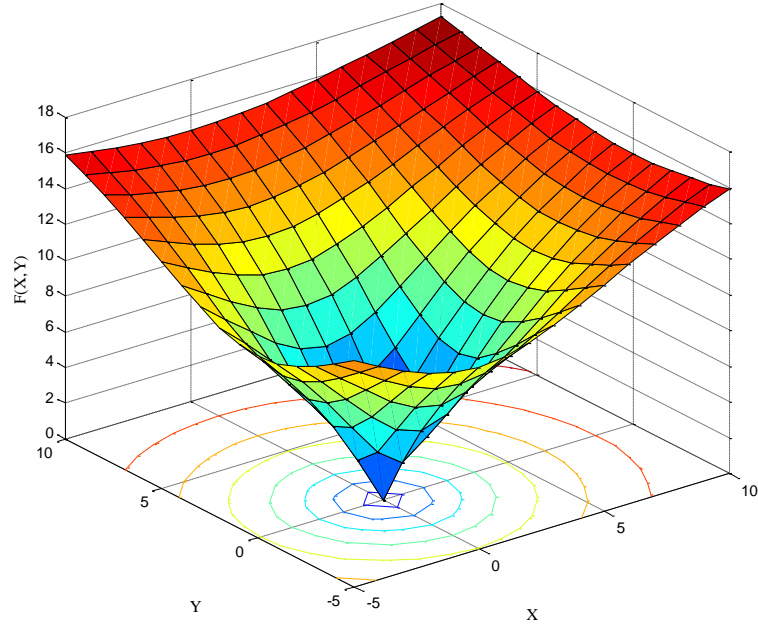


Şekil 3.6. Rastrigin işlevinin $n=2$ için tanımladığı yüzey

3.7. Rosenbrock

İşlev eşitlik 3.7 (Molga and Smutnicki, 2005, s. 6; Rosenbrock, 1960, s. 178) ile tanımlanmakta olup tek çukurlu bir fonksiyondur. Rosenbrock vadisi ya da muz fonksiyonu olarak da bilinir. Küresel optimumu uzun, dar, parabolik şekilli vadi boyunca uzanır. Vadiyi bulmak önemsizdir. Ancak küresel optimuma yakınsaması zor olduğu için optimizasyon algoritmalarının performansını test etmekte sıklıkla kullanılır. Eşitlikte n boyut sayısını göstermekte olup, bu çalışmada $-5 \leq x_i \leq 10$, $i = 1, 2, \dots, n$ aralığındaki arama uzayı kullanılmıştır. İşlevin $n=2$ için yüzeyi Şekil 3.7’de verilmiştir. Şekilden de görüleceği üzere işlevin küresel minimumu $x_1=x_2=\dots=x_n=0$ pozisyonunda 0 değerindedir.

$$f(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2] \quad (3.7)$$

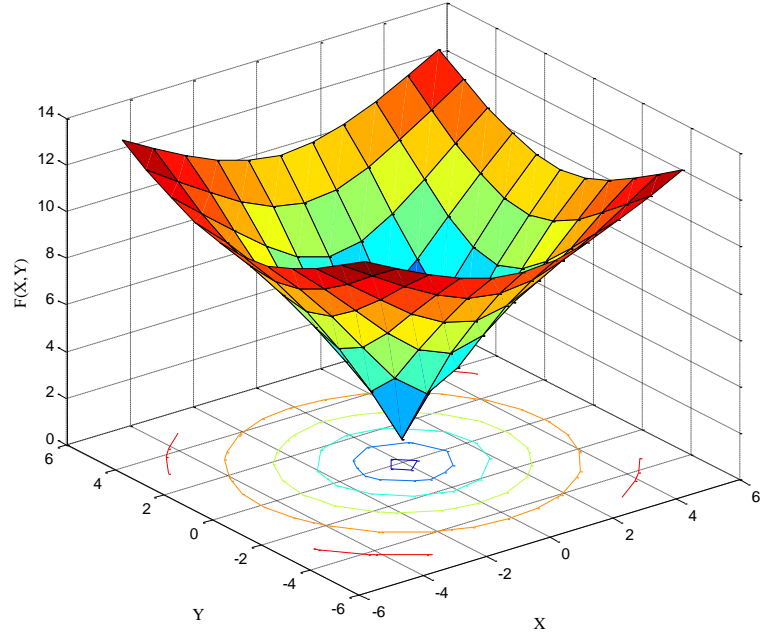


Şekil 3.7. Rosenbrock işlevinin $n=2$ için tanımladığı yüzey

3.8. Sphere

İşlev eşitlik 3.8 (Rahnamayan, Tizhoosh and Salama, 2007, s. 1612) ile tanımlanmakta olup tek çukurlu oldukça basit bir fonksiyondur. Eşitlikte n boyut sayısını göstermekte olup, bu çalışmada $-5,12 \leq x_i \leq 5,12$, $i = 1, 2, \dots, n$ aralığındaki arama uzayı kullanılmıştır. İşlevin $n=2$ için yüzeyi Şekil 3.8’de verilmiştir. Şekilden de görüleceği üzere işlevin küresel minimumu $x_1=x_2=\dots=x_n=0$ pozisyonunda 0 değerindedir.

$$f(x) = \sum_{i=1}^n x_i^2 \quad (3.8)$$



Şekil 3.8. Sphere işlevinin $n=2$ için tanımladığı yüzey

4. SEZGİSEL ALGORİTMALARIN DENEKTAŞI İŞLEVLER ÜZERİNDE BAŞARIM METRİKLERİYLE KARŞILAŞTIRILMASI

Bu bölümde, daha önce belirlediğimiz denektaşı işlevlerin küresel minimumlarını 2, 5 ve 10 boyutlu arama uzayında bulmak için aynı başlangıç şartları altında bu çalışmada kullanılan algoritmalar çalıştırılmıştır. Her boyut için arama uzayında başlangıçta rastgele pozisyonlar belirlenip kaydedilmiştir. Sezgisel algoritmalar denektaşı işlevler için bu başlangıç konumlarından aramaya başlatılmışlardır. Eşit şartlarda kıyaslama yapmak için algoritmalar her işlev için 30'ar defa, 14 popülasyon/sürü büyüklüğü(N) ve 100 nesil sayısı (iterasyon) ile koşturulmuştur. Sürü büyüklüğü eşitlik 4.1'de önerilen denklem kullanılarak belirlenmiştir (Nobile vd., 2015, s. 5; Hansen vd., 2011, s. 5757). 2 boyutlu arama uzayında bu değer 13, 5 boyutlu arama uzayında bu değer 14, 10 boyutlu arama uzayında bu değer 16 çıkmış olup çalışmada bu değerlerin ortalaması olan 14 değeri seçilmiştir.

$$N = \text{round}(10 + 2\sqrt{D}) \quad (4.1)$$

Koşumlar sonucunda karşılaştırma için bu çalışmada kullanılan başarımler için gerekli bilgiler elde edilmiştir. Programın çalışması sonucu metrik ölçüm değerleri ondalıklı kısımları virgülden sonra dört basamak olacak şekilde yuvarlanarak alınmıştır.

Tüm kodlamalar Matlab (R2013a) programı üzerinde geliştirilmiştir. Kullanılan bilgisayar, Intel i7 Core 2,6 GHz işlemci ve 16 GB RAM hafıza alanına sahip olup Windows 10 sürümü işletim sistemi ile çalışmaktadır.

4.1. Karşılaştırma İçin Kullanılan Başarım Ölçüm Metrikleri

Bu çalışmada kullanılan sezgisel algoritmaların denektaşı işlevlerin üç farklı boyutta minimum değer aldığı konumları bulma görevi üzerinde gösterdikleri arama başarımları aşağıdaki alt başlıklarda kısaca tanımlanmıştır.

4.1.1. En iyi ölçüt metriği

İlk metriğimiz olan en iyi ölçüt değeri, algoritmaların belirlenen sayılarda çalıştırılmaları sonucunda en son nesilde elde edilen minimum maliyet değerlerinin ortalamasıdır.

4.1.2. En kötü ölçüt metriği

İkinci metriğimiz olan en kötü ölçüt değeri ise, algoritmaların belirlenen sayılarda çalıştırılmaları sonucunda en son nesilde elde edilen maksimum maliyet değerlerinin ortalamasıdır.

4.1.3. Doğruluk metriği

Üçüncü metriğimiz doğruluk ölçütüdür. Herhangi bir optimizasyon algoritmasının genel hedefi, kabul edilen zaman içerisinde yüksek kalitede bir yaklaşım başarmaktır. f uygunluk fonksiyonu, $Max^{(t)}$ arama uzayında elde edilen en iyi değer, $Min^{(t)}$ arama uzayında elde edilen en kötü değer, A optimizasyon algoritması ve t de iterasyon dizisi olsun. Her bir iterasyondaki doğruluk değeri eşitlik 4.2 ve ortalama doğruluk değeri eşitlik 4.3'teki gibi hesaplanır (Weicker, 2003, s. 69).

$$accuracy_{f,A}^{(t)} = \frac{F(best_A^{(t)}) - Min_f^{(t)}}{Max_f^{(t)} - Min_f^{(t)}} \quad (4.2)$$

$$Acc_{f,t} = Avg(accuracy_f^{(t)}) \quad (4.3)$$

Optimizasyon doğruluğu 0 ile 1 arasında değişen bir değer olup, mümkün olan en iyi doğruluk değeri 1'dir.

4.1.4. Kararlılık metriği

Dördüncü metriğimiz kararlılıktır. Kararlılık algoritmalar için önemli bir husus olup değeri 0 ile 1 arasında değişmektedir. Sıfıra ne kadar yakın olursa o kadar çok kararlı bir algoritmadır diyebiliriz. f uygunluk fonksiyonu, A optimizasyon algoritması ve t de iterasyon dizisi olsun. Kararlılık eşitlik 4.4'te verildiği gibi hesaplanır (Weicker, 2003, s. 70).

$$Stab_{f,t}^A = Avg(\max\{0, accuracy_{f,A}^{(t-1)} - accuracy_{f,A}^{(t)}\}) \quad (4.4)$$

4.1.5. Zaman metriği

Beşinci metriğimiz her bir algoritmanın çalışması esnasında nesil (iterasyon) başına mili saniye cinsinden geçen zamandır. Bu metrik algoritmanın koşma hızı bakımından bilgi verecektir.

4.1.6. Standart sapma metriği

Son olarak altıncı metriğimiz, her çalışmanın son nesilinde elde edilen en iyi ölçüt değerlerinin standart sapmasıdır. Standart sapma değerinin azalması algoritmanın tek tip en iyi maliyet çözümünü elde ettiğini ve çok fazla sapma yapmadan kararlı bir şekilde çözüm bulduğunu gösterir. Öte yandan standart sapma arttıkça her koşmada algoritmanın birbirinden farklı çözümler bulduğu bu sebeple uygunluk değerinde sapmanın fazlalaştığı anlaşılmalıdır.

4.2. Denektaşı İşlevler İçin Karşılaştırma

Bu alt bölümde, karşılaştırma için kullanılan denektaşı işlevler için koşturulan algoritmaların verdiği sonuçlar her bir işlev için ayrı ayrı irdelenmektedir.

4.2.1. Ackley işlevi için karşılaştırma

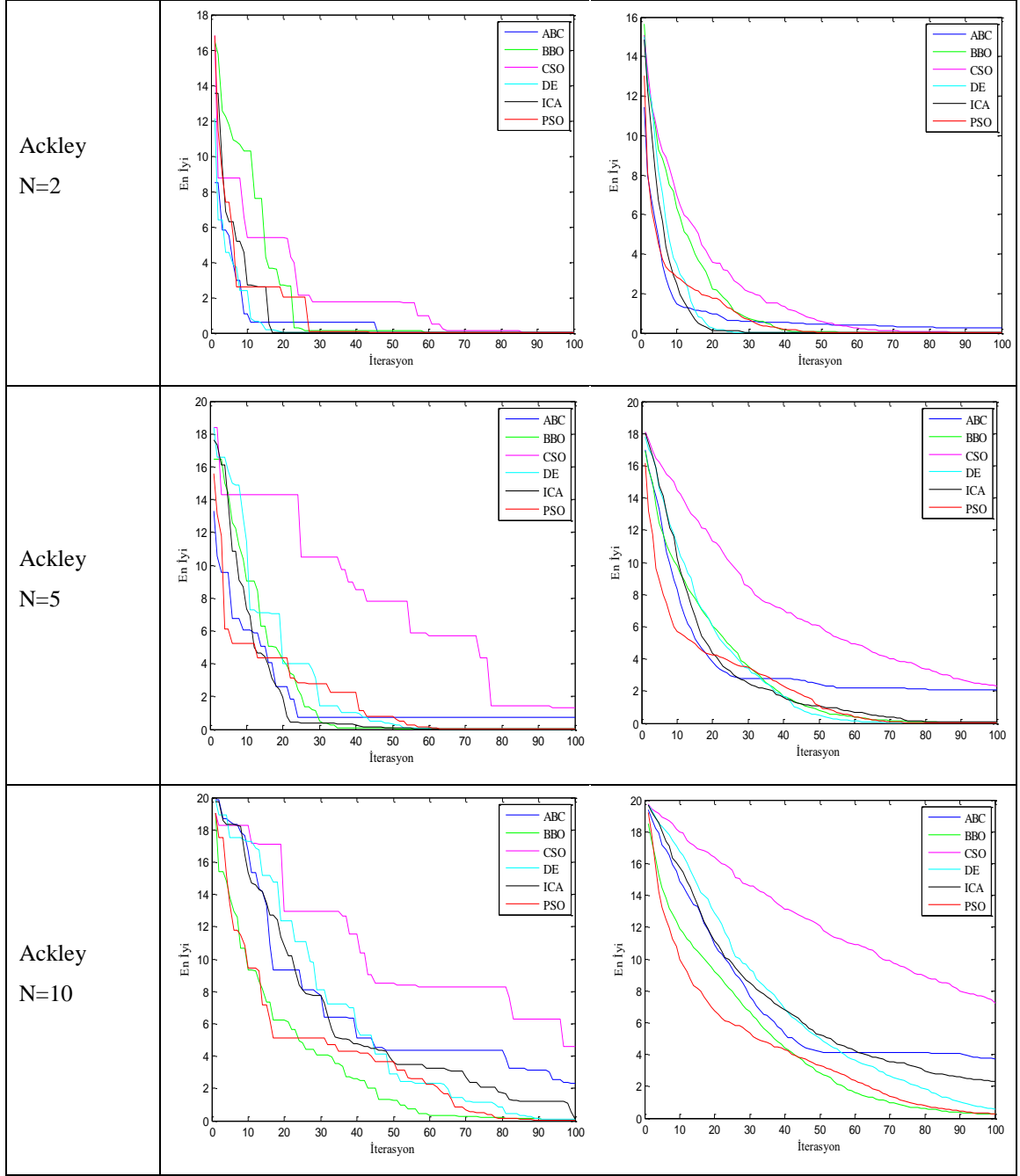
Ackley işlevinin 2, 5 ve 10 boyutlu arama uzayında ayrı ayrı 30'ar kez tüm algoritmaların çalıştırılmasından elde edilen metrik değerleri Tablo 4.1'de verilmiştir. Algoritmalar için en iyi, en kötü, doğruluk, kararlılık, zaman ve standart sapma metrik değerlerine göre sıralamaları yapılmış olup tablonun son satırında ortalama sıralama değeri hesaplanıp verilmiştir. Metriklerin her birinde sıralanma düzeyleri birbirinden farklı olmaktadır. Bu yüzden metrikleri toplu olarak değerlendirip karşılaştırmak daha sağlıklı bir çözüm sunacaktır. Ackley için en iyi maliyet değerlerini ICA ve PSO verirken, süre açısından en süratli işlemlerini bitiren algoritma CSO'dur. Şekil 4.1'de her boyutta 30 koşum boyunca algoritmaların iterasyona göre elde ettiği ortalama en iyi ölçüt değerlerinin grafikleri sağ sütunda, koşumların tümü sonucunda en iyi maliyet değerini veren koşma için en iyi ölçüt değer değişim grafikleri sol sütunda verilmiştir. 2. ve 5. boyutta en hızlı yakınsama davranışını ICA gösterirken, 10. boyutta en hızlı yakınsayan BBO olmuştur. PSO da tüm boyutlarda iyi bir yakınsama eğilimi göstermektedir.

Ackley işlevi için ABC'nin standart sapmasının az olmasının dışında hiçbir metrikte sıralaması iyi olmadığı gözlenmiştir. DE ise doğruluk ve kararlılığı yüksek sonuçlar vermiştir. Ortalama sıralama değerine bakıp sonuçları kümülatif olarak değerlendirirsek, en iyi sonuçları veren algoritmanın bu işlev için PSO olduğu, ikinci ortalamaya sıralamaya sahip DE olduğu, diğer algoritmaların ortalama sıralama

değerleri oldukça uzakta kaldıkları gözlenmektedir. Ackley oldukça zor ve çok çukurlu engebeli bir yüzeye sahip işlev olup problemin farklı boyutlardaki çözümünde PSO kullanımının uygun olduğu görülmektedir.

Tablo 4.1. Ackley işlevi için elde edilen metrik değerleri tablosu

		ABC		BBO		CSO		DE		ICA		PSO	
		Değer	Sıra	Değer	Sıra	Değer	Sıra	Değer	Sıra	Değer	Sıra	Değer	Sıra
Ackley n=2	En iyi	0,2879	6	0,0061	3	0,0169	4	0,0860	5	$1,8312 \cdot 10^{-7}$	1	$9,3601 \cdot 10^{-7}$	2
	En kötü	21,0028	6	0,3308	4	12,1834	5	0,0860	3	$1,8312 \cdot 10^{-7}$	1	0,0057	2
	Doğruluk	0,2647	5	0,3166	4	0,6027	3	0,7602	1	0,0389	6	0,6764	2
	Kararlılık	0,0331	5	0,1037	6	$8,6245 \cdot 10^{-5}$	1	0,0128	3	0,0070	2	0,0282	4
	Zaman	0,1038	6	0,0463	1	0,0556	2	0,0663	3	0,0753	4	0,0797	5
	Standart Sapma	1,8165	1	3,5584	6	3,2740	5	2,7160	4	2,6065	3	2,0589	2
Ackley n=5	En iyi	2,0477	5	0,0147	4	2,3862	6	0,0018	3	0,2058	1	$6,3714 \cdot 10^{-4}$	2
	En kötü	20,6643	6	1,7860	4	14,1894	5	0,0073	2	0,2058	1	0,0135	3
	Doğruluk	0,1440	4	0,5234	1	0,1105	5	0,4376	3	0,0042	6	0,4860	2
	Kararlılık	0,0063	6	0,0204	5	$2,4055 \cdot 10^{-4}$	2	0,0103	3	$8,0967 \cdot 10^{-4}$	1	0,0200	4
	Zaman	0,1037	6	0,0744	5	0,0562	1	0,0669	2	0,0768	3	0,0792	4
	Standart Sapma	3,2268	3	4,1828	1	4,5865	4	4,7390	5	4,6962	6	3,0545	2
Ackley n=10	En iyi	3,7784	5	0,3084	1	7,2654	6	0,5235	3	2,2777	4	0,4870	2
	En kötü	21,2586	6	2,2946	4	18,1955	5	1,3892	2	2,2777	3	0,7136	1
	Doğruluk	0,0721	4	0,2108	1	0,0330	6	0,1114	3	0,0013	5	0,1879	2
	Kararlılık	0,0019	3	0,0067	5	$6,7743 \cdot 10^{-5}$	1	0,0028	4	$2,5302 \cdot 10^{-4}$	2	0,0134	6
	Zaman	0,1034	5	0,1213	6	0,0606	1	0,0678	2	0,0784	3	0,0814	4
	Standart Sapma	4,3350	5	4,8329	6	3,7555	1	5,6817	2	5,0098	3	4,0270	4
Ortalama Sıralama		4,83		3,72		3,5		2,94		3,5		2,54	



Şekil 4.1. Ackley işlevi için algoritmaların 30 koşma boyunca elde ettiği ortalama en iyi ölçüt değerleri değişimi (sağ sütun), en iyi maliyet değerini veren koşma için en iyi ölçüt değerleri değişimi (sol sütun)

4.2.2. Alpine işlevi için karşılaştırma

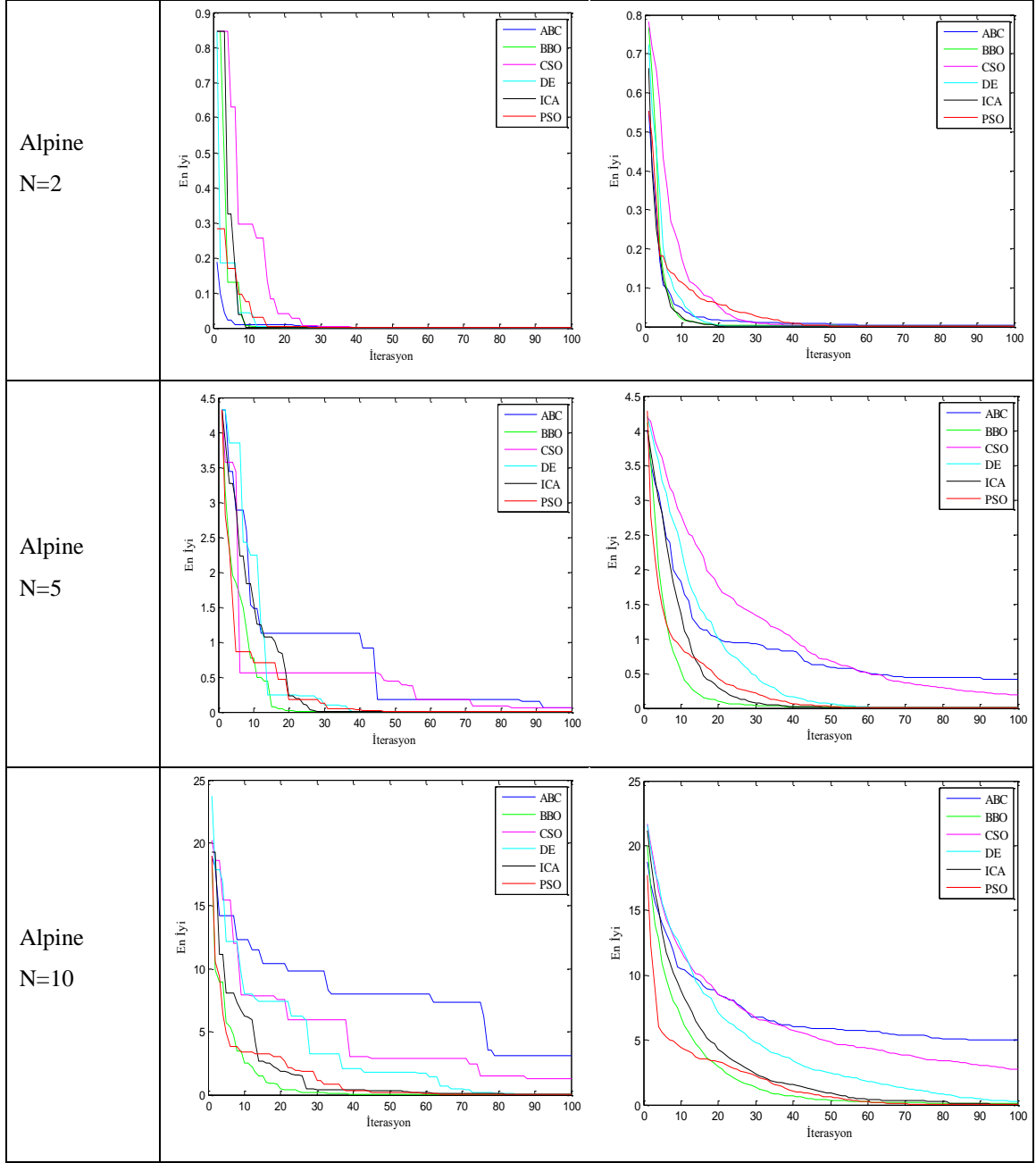
Alpine işlevinin 2, 5 ve 10 boyutlu arama uzayında 30'ar kez algoritmaların çalıştırılmasından elde edilen metrik değerleri Tablo 4.2'deki gibidir. Algoritmalar için en iyi, en kötü, doğruluk, kararlılık, zaman ve standart sapma metriklerinin değerlerine göre sıralamaları yapılmış olup en son satırda da ortalama sıralamaları hesaplanmıştır.

Alpine için en iyi maliyet değerlerini PSO ve ICA bulmaktadır. Şekil 4.2’de 3 boyutta 30 koşma boyunca algoritmaların iterasyona göre elde ettiği ortalama en iyi değerlerinin grafikleri sağ sütunda, koşumların tümü sonucunda en iyi maliyet değerini veren koşma için en iyi ölçüt değerinin değişim grafikleri sol sütunda verilmiştir. Şekillerden ICA ve BBO’nun yakınsama eğilimleri bakımından en iyi algoritmalar olduğu ve genel olarak birbirine benzerlik gösterdiği, CSO ve ABC’nin yakınsama eğilimlerinin ise boyut sayısı arttıkça kötüleştiği değerlendirilebilir. PSO’nun ise farklı boyutlarda iyi yakınsama eğilimi göstermekle birlikte her daim en iyilere yakın bir başarımla sergilediği gözlenmiştir.

Alpine işlevi için ABC’nin standart sapmasının çok az olmasının dışında her metrikte ortalama sıralama bakımından nispeten kötü başarımla gösterdiği standart sapma metriğinin PSO’nunkinden sonra geldiği tablodan görülebilir. CSO diğerlerine göre hızlı, doğru ve en kararlı yapıdadır. Ortalama sıralama değerlerine bakıp sonuçları toplu olarak değerlendirirsek en iyi sonuçları veren PSO, ona en yakın ikinci algoritma DE’dir. Diğerlerinin sıralamaları CSO, ICA, BBO, ABC şeklindedir. Alpine, Ackley kadar zor bir yüzeye sahip değildir ancak yine de çukur sayısı fazladır. Ackley de olduğu gibi Alpine işlevinde de tüm metrikler beraber değerlendirildiğinde en uygun çözümü PSO’nun verdiği görülmüştür.

Tablo 4.2. Alpine işlevi için elde edilen metrik değerler tablosu

		ABC		BBO		CSO		DE		ICA		PSO	
		Değer	Sıra	Değer	Sıra	Değer	Sıra	Değer	Sıra	Değer	Sıra	Değer	Sıra
Alpine n=2	En iyi	0,0027	6	0,0011	5	$4,0597 \cdot 10^{-4}$	4	$3,986 \cdot 10^{-10}$	1	$6,5147 \cdot 10^{-7}$	3	$7,0550 \cdot 10^{-8}$	2
	En kötü	7,8009	6	0,0168	4	0,0842	3	$2,2545 \cdot 10^{-8}$	1	$1,8882 \cdot 10^{-5}$	2	0,5873	5
	Doğruluk	0,8468	4	0,4083	5	0,9640	1	0,9158	3	0,4075	6	0,9369	2
	Kararlılık	0,0285	5	0,1464	6	$7,3217 \cdot 10^{-4}$	1	0,0107	2	0,0160	3	0,0202	4
	Zaman	0,0999	6	0,0452	1	0,0547	2	0,0650	3	0,0875	5	0,0782	4
	Standart Sapma	0,1029	2	0,1250	5	0,1643	6	0,1227	4	0,1222	3	0,0941	1
Alpine n=5	En iyi	0,3526	6	$1,9206 \cdot 10^{-4}$	4	0,1759	5	$8,3213 \cdot 10^{-4}$	3	$3,1299 \cdot 10^{-4}$	1	$1,9206 \cdot 10^{-4}$	2
	En kötü	15,3581	6	0,1364	4	2,2146	5	0,0040	3	$3,1299 \cdot 10^{-4}$	1	0,0577	2
	Doğruluk	0,3675	5	0,7360	2	0,5451	4	0,6275	3	0,0730	6	0,7689	1
	Kararlılık	0,0144	6	0,0699	5	0,0014	1	0,0081	3	0,0076	2	0,0299	4
	Zaman	0,1232	6	0,0886	3	0,0632	1	0,0796	2	0,1651	4	0,0970	5
	Standart Sapma	0,7829	1	0,7244	3	1,0214	6	1,0419	4	0,8700	5	0,6605	2
Alpine n=10	En iyi	4,2929	6	0,0656	2	2,6166	5	0,2531	4	0,1160	3	0,0121	1
	En kötü	33,2790	6	1,3704	4	7,9246	5	1,2178	3	0,1160	2	0,0945	1
	Doğruluk	0,0938	5	0,5210	1	0,1070	4	0,2615	3	0,0064	6	0,4774	2
	Kararlılık	0,0024	3	0,0182	5	$7,1777 \cdot 10^{-4}$	1	0,0032	4	0,0012	2	0,0264	6
	Zaman	0,0983	5	0,1168	6	0,0602	1	0,0674	2	0,0776	3	0,0796	4
	Standart Sapma	3,1277	2	3,6365	3	3,9684	4	4,8828	6	4,4593	5	2,6283	1
Ortalama Sıralama		4,66		3,77		3,27		3		3,44		2,72	



Şekil 4.2. Alpine işlevi için algoritmaların 30 koşma boyunca elde ettiği ortalama en iyi ölçüt değeri değişimi (sağ sütun), en iyi maliyet değerini veren koşma için en iyi ölçüt değeri değişimi (sol sütun)

4.2.3. Bohachevsky işlevi için karşılaştırma

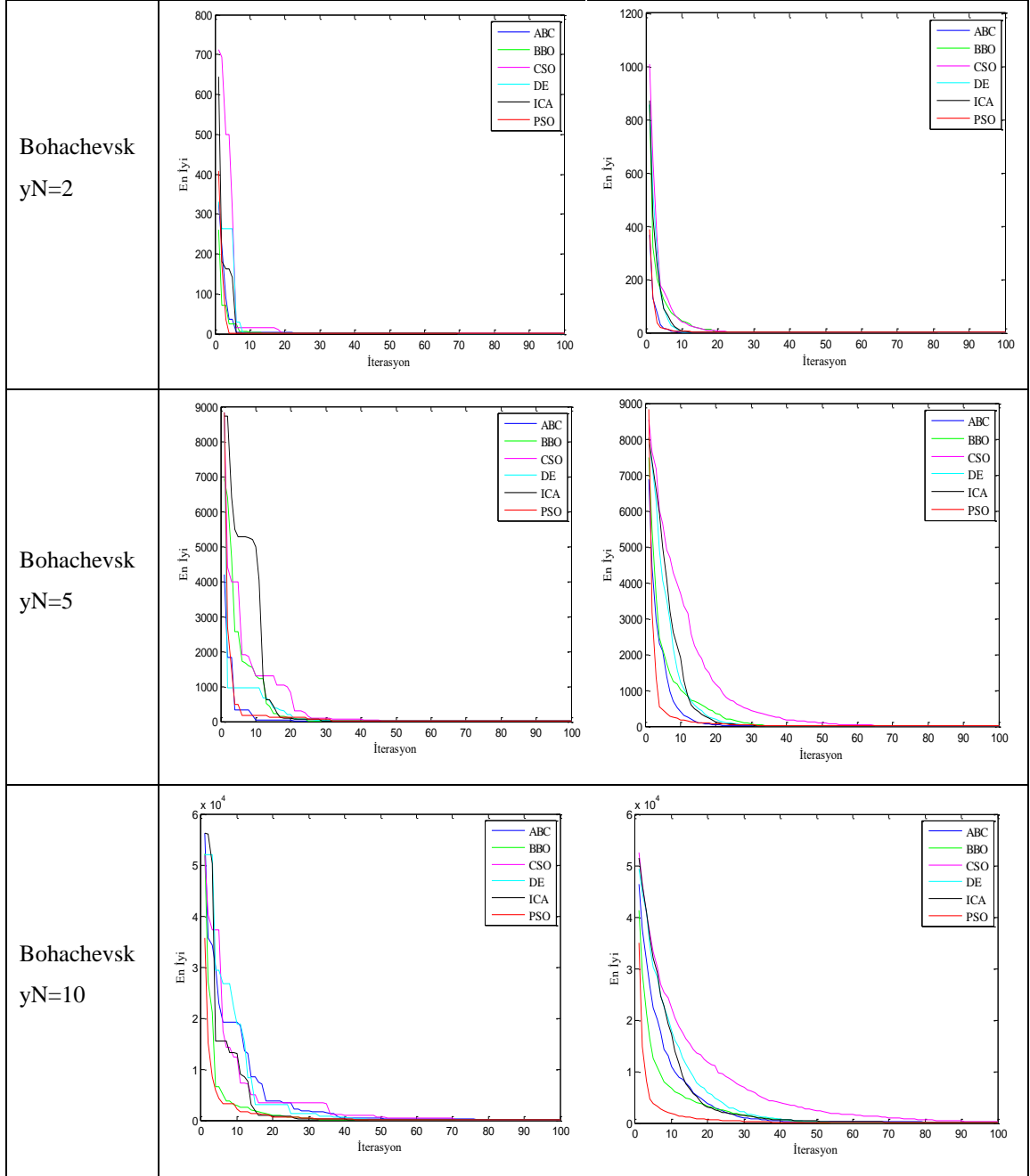
Bohachevsky işlevinin 2, 5 ve 10 boyutlu arama uzayında ayrı ayrı 30'ar kez algoritmaların çalıştırılmasından elde edilen metrik değerleri Tablo 4.3'teki gibidir. Algoritmalar için en iyi, en kötü, doğruluk, kararlılık, zaman ve standart sapma metriklerinin matematiksel değerlerine göre sıralamaları yapılmış olup son satırda da ortalama sıralama değerleri hesaplanarak verilmiştir. Bohachevsky işlevi için en iyi

maliyet deęerlerini PSO vermiř olup dięerleri ile arasında aık fark saęlamıřtır. Őekil 4.3'te her boyutta 30 kořum boyunca algoritmaların iterasyona gre elde ettięi ortalama en iyi lt deęerlerinin deęiřim grafikleri saę stunda, belirlenen iyi kořma iin en iyi lt deęerinin deęiřim grafikleri sol stunda verilmiřtir. 2. boyutta algoritmaların en iyiye yakınsama eęilimleri birbirine ok yakın iken boyut sayısı arttıķa farklar artmakta CSO ve ABC dięerlerinden uzaklařmaktadır. Tm boyutlarda genel olarak en hızlı yakınsamayı yapan algoritma ise PSO'dur.

Bohachevsky iřlevi iin ABC standart sapma metrięinde 2. olmasının dıřında neredeyse tm metriklerde sonuncu sıradadır. Doęruluk ve kararlılıkta en iyi sonuları veren DE, geen sre bakımından en hızlı olan CSO'dur. Ortalama sıralama deęerlerine bakarak deęerlendirecek olursak en iyi iki algoritma eřit ortalama sıralamasına sahip olan PSO ve DE'dir. En kt ikili ise bunlardan oldukça uzak sıralamaları bulunan ABC ve BBO'dur. Bohachevsky engebesi az ve ok ukurlu bir iřlev olup problemin farklı boyutlardaki zmnde PSO ve DE kullanımının stnlk saęladıęı grlmektedir.

Tablo 4.3. Bohachevsky işlevi için elde edilen metrik değerler tablosu

		ABC		BBO		CSO		DE		ICA		PSO	
		Değer	Sıra	Değer	Sıra	Değer	Sıra	Değer	Sıra	Değer	Sıra	Değer	Sıra
Bohachevsky n=2	En iyi	0,2719	6	0,0225	5	0,0025	4	0,0000	3	$3,8859 \cdot 10^{-5}$	2	$8,564 \cdot 10^{-11}$	1
	En kötü	$1,4728 \cdot 10^4$	6	0,8709	4	29,7467	5	$1,480 \cdot 10^{-17}$	1	$3,8863 \cdot 10^{-5}$	2	0,0034	3
	Doğruluk	0,3187	5	0,3641	4	0,6594	3	0,8136	1	0,2291	6	0,6640	2
	Kararlılık	0,0401	5	0,1000	6	$1,1274 \cdot 10^{-4}$	1	0,0036	2	0,0093	3	0,0295	4
	Zaman	0,1102	6	0,0494	1	0,0612	2	0,0703	3	0,0809	4	0,0838	5
	Standart Sapma	40,2032	2	103,1674	3	153,7187	6	110,9041	5	106,8368	4	39,2234	1
Bohachevsky n=5	En iyi	0,0429	2	0,9227	6	0,1759	3	0,2154	4	0,0000	1	0,9043	5
	En kötü	$4,6692 \cdot 10^4$	4	21,1980	5	152,69	6	0,0050	1	0,3474	3	0,0891	2
	Doğruluk	0,0393	5	0,4288	1	0,0495	4	0,4110	2	0,0004	6	0,3598	3
	Kararlılık	0,0029	3	0,0184	6	$1,3215 \cdot 10^{-4}$	1	0,0048	4	$1,7342 \cdot 10^{-4}$	2	0,0123	5
	Zaman	0,1037	6	0,0753	3	0,0553	1	0,0673	2	0,0779	4	0,0843	5
	Standart Sapma	971,9655	2	$1,0285 \cdot 10^3$	3	$2,0485 \cdot 10^3$	6	$1,5052 \cdot 10^3$	4	$1,6595 \cdot 10^3$	5	953,6845	1
Bohachevsky n=10	En iyi	111,1121	5	3,7910	2	202,8562	6	5,5841	3	11,3526	4	1,7985	1
	En kötü	$9,9242 \cdot 10^4$	6	61,8531	4	$2,3889 \cdot 10^3$	5	13,1601	3	11,3526	2	6,0220	1
	Doğruluk	0,0030	4	0,0565	2	0,0012	5	0,0212	3	$7,6858 \cdot 10^{-6}$	6	0,0598	1
	Kararlılık	$1,0240 \cdot 10^{-4}$	3	0,0017	5	$4,4887 \cdot 10^{-6}$	2	$5,6158 \cdot 10^{-4}$	4	$1,7080 \cdot 10^{-6}$	1	0,0039	6
	Zaman	0,1026	5	0,1213	6	0,0642	1	0,0687	2	0,0786	3	0,0820	4
	Standart Sapma	$8,4768 \cdot 10^3$	3	$6,0546 \cdot 10^3$	2	$1,0691 \cdot 10^4$	6	$1,0182 \cdot 10^4$	4	$1,0229 \cdot 10^4$	5	$3,8129 \cdot 10^3$	1
Ortalama Sıralama		4,33		3,77		3,72		2,83		3,5		2,83	



Şekil 4.3. Bohachevsky işlevi için algoritmaların 30 koşma boyunca elde ettiği ortalama en iyi ölçüt değeri değişimi (sağ sütun), en iyi maliyet değeri veren koşma için en iyi ölçüt değeri değişimi (sol sütun)

4.2.4. Griewank işlevi için karşılaştırma

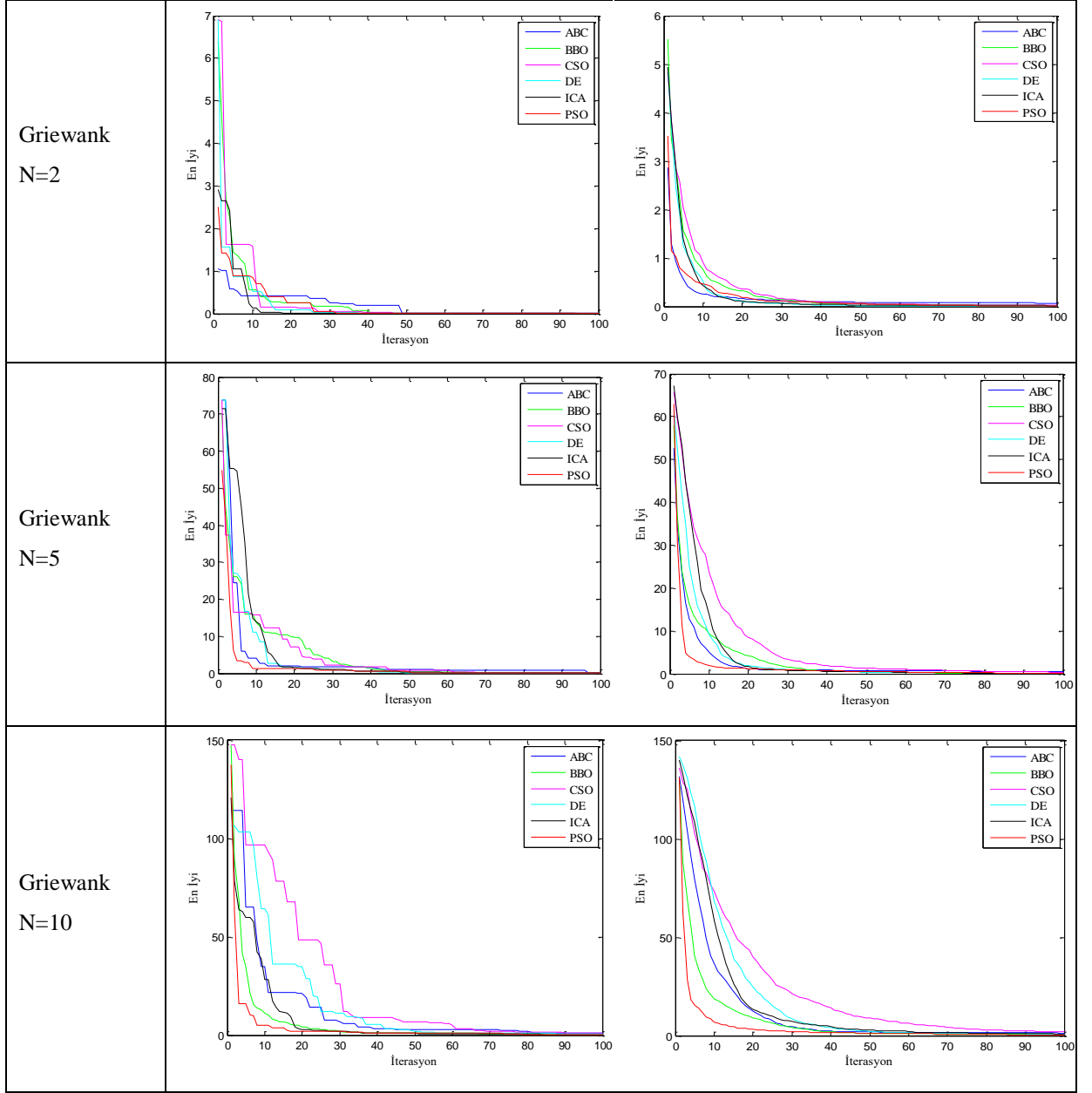
Griewank işlevinin 2, 5 ve 10 boyutlu arama uzayında 30’ar kez algoritmaların çalıştırılmasından elde edilen metrik değerler Tablo 4.4’teki gibidir. Algoritmalar için en iyi, en kötü, doğruluk, kararlılık, zaman ve standart sapma metriklerinin matematiksel değerlerine göre sıralamaları yapılmış olup her bir algoritma için ortalama sıralama hesaplanmıştır. Griewank işlevi için en iyi başarımlı değerlerini PSO, takiben

de BBO vermiştir. Şekil 4.4'te her boyutta 30 koşma boyunca algoritmaların iterasyona göre elde ettiği ortalama en iyi ölçüt değerlerinin grafikleri sağ sütunda, en iyi koşmaya ait en iyi ölçüt değerinin değişim grafikleri de sol sütunda verilmiştir. Boyutlardaki en iyi çalışma ve ortalama çalışma grafikleri birbirine benzemektedir. 2. ve 5. boyutta en hızlı yakınsama davranışını ABC ve PSO gösterirken, 10. boyutta en hızlı yakınsayanlar PSO ve BBO'dur. Her boyutta en kötü yakınsama CSO'nunkidir.

Griewank işlevi için ABC'nin hiçbir metrikte sıralaması iyi değildir. DE ise doğruluk ve kararlılık karşılaştırmasında 1., CSO geçen zaman açısından 1.'dir. Ortalama sıralama değerlerine bakıp tüm metrikleri bir arada değerlendirecek olursak en iyi sonuçları veren algoritma DE'dir. Diğer algoritmaların ortalama sıralaması bu değerden çok uzakta olup birbirine yakın değerlerdedir. Griewank çok çukurlu geniş dağılmış minimum noktalara sahip yüzeyli bir işlev olup problemin farklı boyutlardaki çözümünde DE kullanımının uygun olduğu görülmektedir. PSO bu işlevde sadece en iyi maliyet hesabında önde olup, diğer metriklerde arkada kalarak ortalama sıralamasını geriletmiştir.

Tablo 4.4. Griewank işlevi için elde edilen metrik değerler tablosu

		ABC		BBO		CSO		DE		ICA		PSO	
		Değer	Sıra	Değer	Sıra	Değer	Sıra	Değer	Sıra	Değer	Sıra	Değer	Sıra
Griewank n=2	En iyi	0,0753	6	0,0180	4	0,0131	2	0,0031	1	0,0211	5	0,0136	3
	En kötü	81,7261	6	0,1023	3	0,7325	4	0,0530	2	0,0215	1	1,7109	5
	Doğruluk	0,6820	4	0,3962	5	0,8361	2	0,8501	1	0,2119	6	0,7517	3
	Kararlılık	0,0290	4	0,1125	6	9,0100 10 ⁻⁴	1	0,0039	2	0,0121	3	0,0539	5
	Zaman	0,1046	6	0,0460	1	0,0564	2	0,0647	3	0,0802	5	0,0763	4
	Standart Sapma	0,3464	1	0,8679	6	0,8571	5	0,7507	4	0,7399	3	0,3653	2
Griewank n=5	En iyi	0,5753	6	0,0856	1	0,3916	5	0,1446	3	0,1563	4	0,1370	2
	En kötü	160,4964	6	1,1820	3	2,4701	5	0,6283	2	0,1563	1	1,7282	4
	Doğruluk	0,4215	4	0,4508	2	0,3462	5	0,4532	1	0,0106	6	0,4345	3
	Kararlılık	0,0082	4	0,0177	5	0,0018	1	0,0042	3	0,0025	2	0,0379	6
	Zaman	0,1240	6	0,0888	3	0,0685	1	0,0782	2	0,0926	4	0,0968	5
	Standart Sapma	6,5878	6	8,8798	3	12,8851	1	10,9715	2	13,1765	4	6,5271	5
Griewank n=10	En iyi	1,2740	5	0,4902	2	2,0066	6	0,7354	4	0,5703	3	0,2108	1
	En kötü	354,6565	6	1,4793	4	10,1471	5	1,0481	2	0,5703	1	1,3801	3
	Doğruluk	0,2291	2	0,1620	3	0,1125	5	0,1202	4	0,0023	6	0,2665	1
	Kararlılık	0,0032	3	0,0106	5	7,8557 10 ⁻⁴	2	0,0043	4	5,4127 10 ⁻⁴	1	0,0218	6
	Zaman	0,1016	5	0,1184	6	0,0624	1	0,0676	2	0,0773	3	0,0801	4
	Standart Sapma	25,3949	3	20,1151	2	34,6429	6	32,7671	5	29,3046	4	14,8543	1
Ortalama Sıralama		4,61		3,55		3,27		2,61		3,44		3,5	



Şekil 4.4. Griewank işlevi için algoritmaların 30 koşma boyunca elde ettiği ortalama en iyi ölçüt değerleri değişimi (sağ sütun), en iyi maliyet değerini veren koşma için en iyi ölçüt değerleri değişimi (sol sütun)

4.2.5. Michalewicz işlevi için karşılaştırma

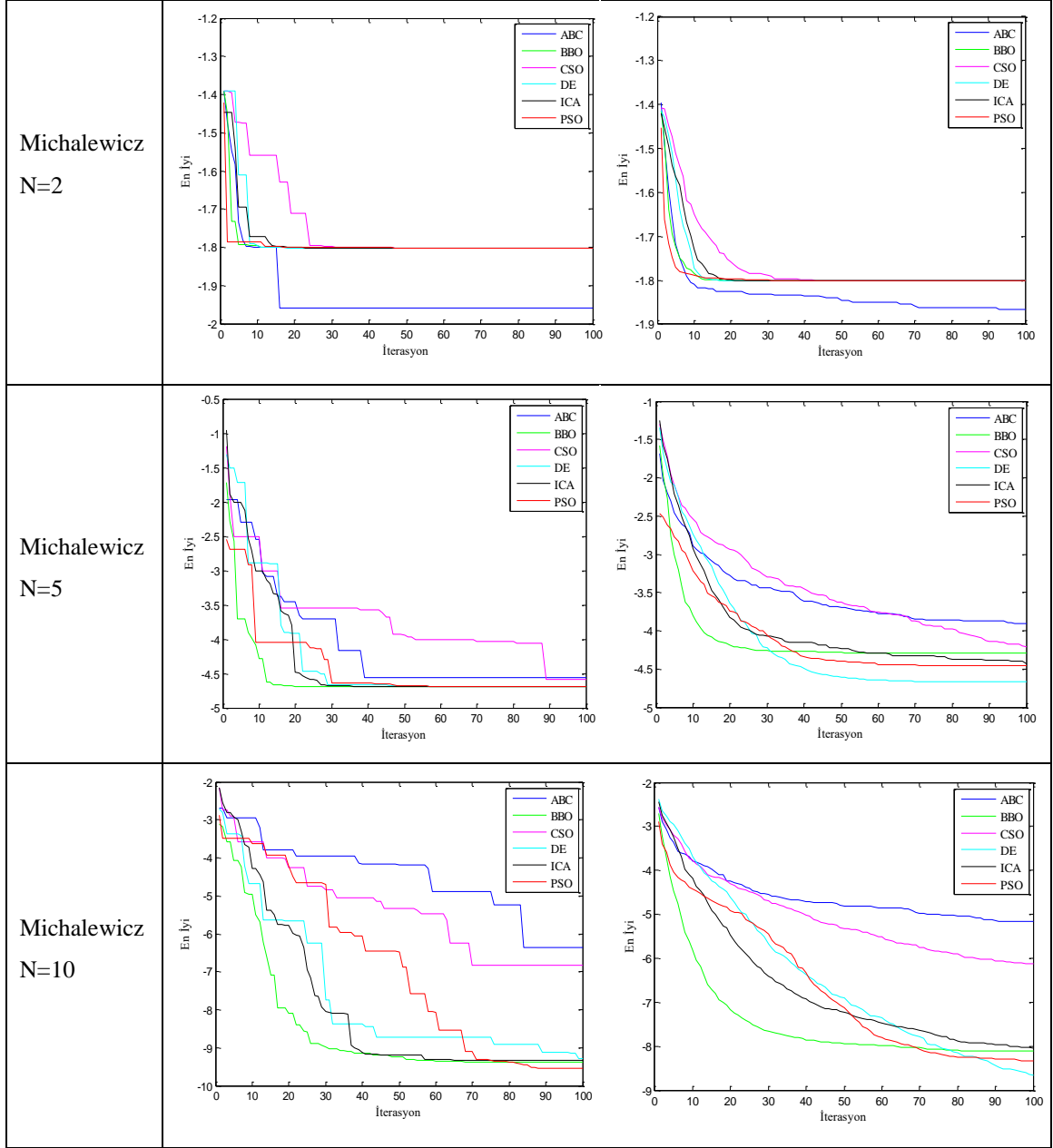
Michalewicz işlevinin 2, 5 ve 10 boyutlu arama uzayında 30'ar kez algoritmaların çalıştırılmasından elde edilen metrik değerleri Tablo 4.5'te verildiği gibidir. Algoritmalar için en iyi, en kötü, doğruluk, kararlılık, zaman ve standart sapma metriklerinin matematiksel değerlerine göre sıralamaları yapılmış olup son satırda da ortalama sıralama değerleri hesaplanıp verilmiştir. Michalewicz için en iyi maliyet değerlerini DE vermektedir. Şekil 4.5'te her boyutta 30 koşma boyunca algoritmaların

iterasyona göre elde ettiği ortalama en iyi ölçüt değerlerinin grafikleri sağ sütunda, en iyi sonucu veren koşmanın en iyi ölçüt değerinin değişim grafikleri sol sütunda verilmiştir. Boyutlara bakıldığında en hızlı yakınsamayı gösteren BBO olsa da en iyi değerleri veren DE'dir. ABC en iyi değere yaklaşmış olsa da boyut sayısı arttıkça daha da başarısız olmaktadır.

Michalewicz işlevi için standart sapması az olan algoritmalar BBO ve ABC'dir. Boyut sayısı büyüdükçe DE'nin kararlılık ve doğruluk sıralamaları azalırken, ABC ve BBO'nun sıralama değerleri büyümekte yani algoritmaların doğruluk ve kararlılıkları bozulmaktadır. CSO istenen doğruluk ve kararlılık değerlerine ulaşmada daha başarılı olup, süre açısından da en hızlısıdır. Tabloya toplu olarak bakıp tümünü değerlendirecek olursak en iyi ortalama sıralamaya sahip DE, ikinci olarak da CSO'dur. Michalewicz çok çukurlu ancak çok zor olmayan bir problemdir. Çözümünde DE nispeten başarılı sonuç vermiştir.

Tablo 4.5. Michalewicz işlevi için elde edilen metrik değerler tablosu

		ABC		BBO		CSO		DE		ICA		PSO	
		Değer	Sıra	Değer	Sıra	Değer	Sıra	Değer	Sıra	Değer	Sıra	Değer	Sıra
Michalewicz n=2	En iyi	-1,8616	6	-1,8013	4	-1,8013	5	-1,8013	1	-1,8013	3	-1,8013	2
	En kötü	-0,2133	6	-1,7907	4	-1,1933	5	-1,8013	1	-1,8013	3	-1,8013	2
	Doğruluk	0,9861	1	0,4411	4	0,9370	2	0,3884	6	0,4003	5	0,8081	3
	Kararlılık	0,0035	2	0,1785	5	2,2706 10 ⁻⁴	1	0,0692	4	0,0315	3	0,3726	6
	Zaman	0,1047	6	0,0455	1	0,0507	2	0,0640	3	0,0828	5	0,0790	4
	Standart Sapma	0,0703	4	0,0543	2	0,1009	6	0,0760	5	0,0687	3	0,0389	1
Michalewicz n=5	En iyi	-3,9623	6	-4,1969	5	-4,2636	4	-4,6670	1	-4,4282	2	-4,3978	3
	En kötü	-0,6176	6	-4,0459	4	-2,5064	5	-4,6645	1	-4,4282	2	-4,2009	3
	Doğruluk	0,6113	3	0,2484	5	0,6122	2	0,7376	1	0,0640	6	0,5909	4
	Kararlılık	0,0174	4	0,0804	6	0,0028	1	0,0083	2	0,0095	3	0,0341	5
	Zaman	0,1098	6	0,0789	3	0,0624	1	0,0709	2	0,0864	4	0,0875	5
	Standart Sapma	0,5215	2	0,4742	1	0,6489	4	0,8634	6	0,7363	5	0,5290	3
Michalewicz n=10	En iyi	-5,2351	6	-8,1767	3	-6,3175	5	-8,6485	1	-8,0527	4	-8,6174	2
	En kötü	-1,3954	6	-7,5696	3	-3,8800	5	-6,9365	4	-8,0527	2	-8,5106	1
	Doğruluk	0,3308	4	0,2761	5	0,3315	3	0,4500	1	0,0354	6	0,4002	2
	Kararlılık	0,0106	4	0,0377	6	0,0029	1	0,0058	3	0,0053	2	0,0341	5
	Zaman	0,1096	5	0,1268	6	0,0625	1	0,0735	2	0,0863	3	0,0871	4
	Standart Sapma	0,6067	1	1,2316	3	0,9912	2	1,7641	6	1,4686	4	1,6717	5
Ortalama Sıralama		4,33		3,88		3,11		2,88		3,61		3,44	



Şekil 4.5. Michalewicz işlevi için algoritmaların 30 koşma boyunca elde ettiği ortalama en iyi ölçüt değerleri değişimi (sağ sütun), en iyi maliyet değerini veren koşma için en iyi ölçüt değerleri değişimi (sol sütun)

4.2.6. Rastrigin işlevi için karşılaştırma

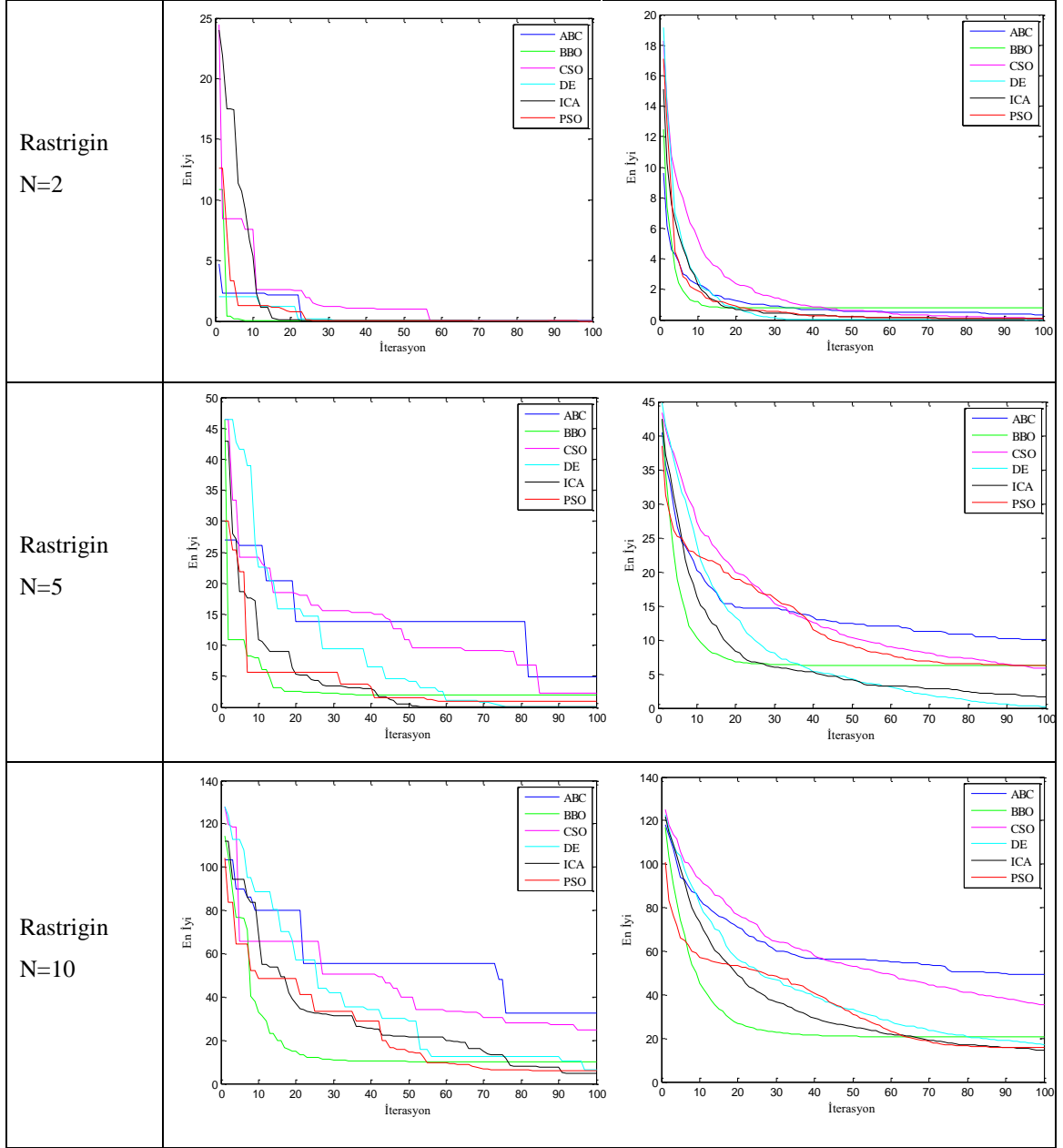
Rastrigin işlevinin 2, 5 ve 10 boyutlu arama uzayında 30’ar kez algoritmaların çalıştırılmasından elde edilen metrik değerleri Tablo 4.6’daki gibidir. Algoritmalar için en iyi, en kötü, doğruluk, kararlılık, zaman ve standart sapma metriklerinin matematiksel değerlerine göre sıralamaları yapılmış olup tablonun son satırında da ortalama sıralama değerleri verilmiştir. Rastrigin için en iyi maliyet değerlerini DE ve ICA vermekte olup, özellikle DE 2 boyut için çok iyi bir başarıyı sergilemiştir. Şekil

4.6’da her boyutta 30 kořma boyunca algoritmaların iterasyona gre elde ettiđi ortalama en iyi lt deđerlerinin deđiřim grafikleri sađ stunda, en iyi kořma iin ise aynı deđerim grafikleri de sol stunda verilmiřtir. 2. boyutta iken yakınsama eđilimleri birbirine ok benzemekte iken, boyut sayısı arttıa zellikle ABC ve BBO kresel minimum deđerinden uzaklařmaktadır. BBO da diđerlerine gre hemen yakınsamaya bařlasa da en iyi deđerini elde edememektedir.

Rastrigin iřlevi iin ABC ve BBO’nun standart sapmaları ideal ancak diđer sonuları istenildiđi gibi olmadıđı iin sıralamaları ktdr. alıřma sresi bakımından en hızlı CSO sonra DE’dir. Ortalama sıralamalarına gre 1. DE olup standart sapması haricinde her metrikte beklenen sonuları sunduđu grlmektedir. Rastrigin olduka zor ve ok ukurlu engebeli bir yzeye sahip iřlev olup problemin farklı boyutlardaki zmnde en uygun algoritma DE’dir.

Tablo 4.6. Rastrigin işlevi için elde edilen metrik değerler tablosu

		ABC		BBO		CSO		DE		ICA		PSO	
		Değer	Sıra	Değer	Sıra	Değer	Sıra	Değer	Sıra	Değer	Sıra	Değer	Sıra
Rastrigin n=2	En iyi	0,2503	4	0,5307	6	0,0635	3	$3,789 \cdot 10^{-15}$	1	0,0332	2	0,2653	5
	En kötü	42,8856	6	0,8682	3	5,3082	5	$1,237 \cdot 10^{-13}$	1	0,0332	2	3,6740	4
	Doğruluk	0,2726	5	0,3013	4	0,6163	2	0,7781	1	0,0904	6	0,5920	3
	Kararlılık	0,0307	5	0,0879	6	$5,6431 \cdot 10^{-4}$	1	0,0033	3	0,0075	1	0,0325	4
	Zaman	0,0936	6	0,0413	1	0,0433	2	0,0605	3	0,0706	4	0,0739	5
	Standart Sapma	1,6267	1	1,8807	2	2,8810	6	2,5059	5	2,1565	3	2,2766	4
Rastrigin n=5	En iyi	10,1845	6	5,3397	3	5,9174	4	0,2868	1	1,3323	2	6,1188	5
	En kötü	106,2344	6	9,3803	3	29,6496	5	1,9940	2	1,3323	1	15,8633	4
	Doğruluk	0,0398	5	0,0634	4	0,0698	2	0,2567	1	0,0053	6	0,0667	3
	Kararlılık	0,0018	3	0,0198	6	0,0002	1	0,0021	4	0,0009	2	0,0061	5
	Zaman	0,0940	6	0,0696	3	0,0506	1	0,0612	2	0,0733	4	0,0761	5
	Standart Sapma	6,2905	2	6,1742	1	9,5933	5	10,4225	6	8,1931	4	6,7420	3
Rastrigin n=10	En iyi	45,9536	6	20,7903	4	34,1621	5	17,6710	3	13,3920	1	13,9688	2
	En kötü	196,3210	6	31,2203	3	84,9062	5	42,5140	4	13,3920	1	17,7396	2
	Doğruluk	0,0090	5	0,0165	3	0,0104	4	0,0173	2	$6,9879 \cdot 10^{-4}$	6	0,0223	1
	Kararlılık	$2,5830 \cdot 10^{-4}$	4	0,0024	6	$6,1834 \cdot 10^{-5}$	1	$1,6701 \cdot 10^{-4}$	3	$1,1979 \cdot 10^{-4}$	2	0,0020	5
	Zaman	0,0930	5	0,1111	6	0,0475	1	0,0619	2	0,0730	3	0,0766	4
	Standart Sapma	18,2585	2	17,5380	1	21,3532	4	25,7843	6	24,8035	5	20,5887	3
Ortalama Sıralama		4,61		3,61		3,16		2,77		3,5		3,72	



Şekil 4.6. Rastrigin işlevi için algoritmaların 30 koşma boyunca elde ettiği ortalama en iyi ölçüt değeri değişimi (sağ sütun), en iyi maliyet değeri veren koşma için en iyi ölçüt değeri değişimi (sol sütun)

4.2.7. Rosenbrock işlevi için karşılaştırma

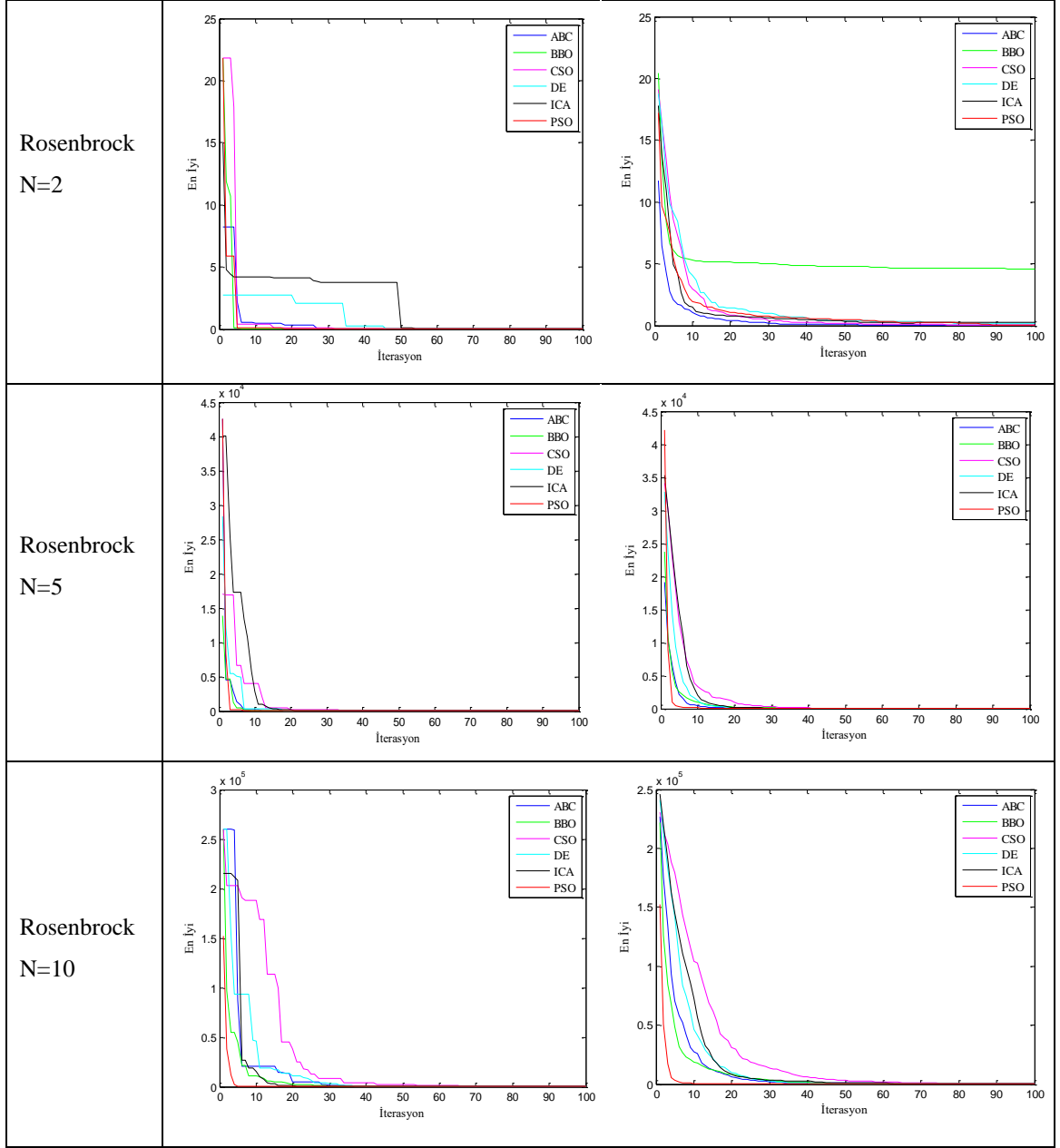
Rosenbrock işlevinin 2, 5 ve 10 boyutlu arama uzayında 30'ar kez algoritmaların çalıştırılmasından elde edilen metrik değerler Tablo 4.7'de verilmiştir. Algoritmalar için en iyi, en kötü, doğruluk, kararlılık, zaman ve standart sapma metriklerinin matematiksel değerlerine göre sıralamaları yapılmış olup tablonun son satırında da ortalama sıralamaları verilmiştir. Rastrigin işlevi için en iyi maliyet değerlerini DE ve PSO vermektedir. Şekil 4.7'de her boyutta 30 koşma boyunca algoritmaların iterasyona

göre elde ettiği ortalama en iyi ölçüt değerlerinin değişim grafikleri sağ sütunda, en iyi koşma için en iyi ölçüt değerlerinin değişim grafikleri sol sütunda verilmiştir. Boyut sayısı yükseldikçe CSO'nun en iyi değere ulaşması gecikmekte, PSO'nun da yakınsama düzeyi artmaktadır.

Rosenbrock işlevi için ABC ve BBO'nun standart sapmaları ideal ancak diğer sonuçları istenildiği gibi olmadığı için sıralamaları kötüdür. Bu işlev için en hızlı çalışan algoritma CSO'dur. Ortalama sıralamalarına göre metriklerin toplu olarak değerlendirilmesi sonucu en iyi olan algoritma DE'dir. Diğer algoritmaların başarımları DE'nin başarımlarından oldukça uzak olduğundan Rosenbrock gibi tek çukurlu bir işlevde DE kullanılması başarılı çözümler üretecektir.

Tablo 4.7. Rosenbrock işlevi için elde edilen metrik değerler tablosu

		ABC		BBO		CSO		DE		ICA		PSO	
		Değer	Sıra	Değer	Sıra	Değer	Sıra	Değer	Sıra	Değer	Sıra	Değer	Sıra
Rosenbrock n=2	En iyi	0,0508	1	7,1771	6	0,0669	2	0,0829	3	0,2477	5	0,1813	4
	En kötü	2,7939 10 ⁵	1	9,9119	4	12,9374	5	1,6184	3	0,3853	2	350,9869	6
	Doğruluk	0,5078	4	0,0410	6	0,7461	1	0,6384	2	0,3380	5	0,6256	3
	Kararlılık	0,0423	5	0,0228	4	1,8923 10 ⁻⁴	1	0,0014	2	0,0073	3	0,0483	6
	Zaman	0,1145	6	0,0504	1	0,0521	2	0,0711	4	0,1017	5	0,0888	3
	Standart Sapma	1,8501	2	1,6869	1	3,6559	6	3,1871	5	2,6586	3	2,8835	4
Rosenbrock n=5	En iyi	16,0227	4	83,0673	6	5,4750	3	3,5573	2	51,0070	5	2,3884	1
	En kötü	6,6830 10 ⁵	6	109,2609	4	433,7356	5	57,4264	3	51,0070	2	25,8098	1
	Doğruluk	0,0221	5	0,0376	4	0,0548	3	0,0901	2	6,5090 10 ⁻⁴	6	0,1643	1
	Kararlılık	0,0016	4	0,0093	5	3,0872 10 ⁻⁵	1	2,8045 10 ⁻⁴	3	1,5944 10 ⁻⁴	2	0,0124	6
	Zaman	0,0951	6	0,0697	3	0,0499	1	0,0604	2	0,0739	4	0,0764	5
	Standart Sapma	2,4417 10 ³	1	3,1990 10 ³	2	6,2416 10 ³	6	4,1070 10 ³	3	5,8823 10 ³	5	4,3029 10 ³	4
Rosenbrock n=10	En iyi	403,8956	6	101,9570	3	289,0692	5	65,1620	2	180,0527	4	13,8020	1
	En kötü	1,7381 10 ⁶	6	153,8552	2	3,9721 10 ³	5	274,9244	4	180,0527	3	25,8404	1
	Doğruluk	0,0012	5	0,0087	2	0,0013	4	0,0046	3	3,2898 10 ⁻⁶	6	0,0270	1
	Kararlılık	4,0921 10 ⁻⁵	4	5,5331 10 ⁻⁴	5	3,9455 10 ⁻⁶	2	3,7907 10 ⁻⁵	3	8,3626 10 ⁻⁷	1	0,0034	6
	Zaman	0,0951	5	0,1122	6	0,0486	1	0,0624	2	0,0739	3	0,0802	4
	Standart Sapma	3,6915 10 ⁴	3	2,8844 10 ⁴	2	5,4587 10 ⁴	6	4,5643 10 ⁴	4	4,9910 10 ⁴	5	1,6572 10 ⁴	1
Ortalama Sıralama		4,11		3,66		3,27		2,88		3,83		3,22	



Şekil 4.7. Rosenbrock işlevi için algoritmaların 30 koşma boyunca elde ettiği ortalama en iyi ölçüt değerleri değişimi (sağ sütun), en iyi maliyet değerini veren koşma için en iyi ölçüt değerleri değişimi (sol sütun)

4.2.8. Sphere işlevi için karşılaştırma

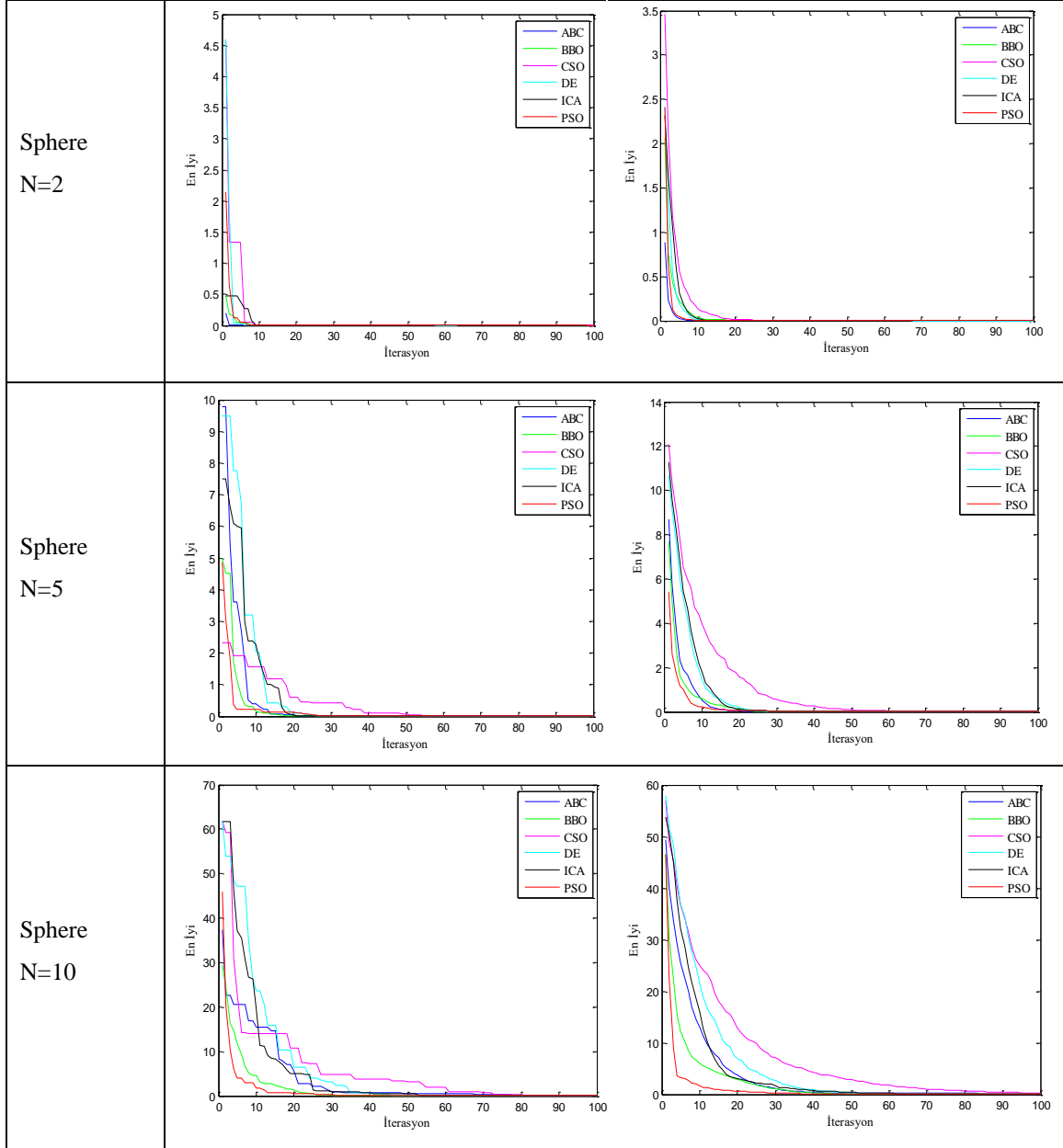
Sphere işlevi için 2, 5 ve 10 boyutlu arama uzayında 30'ar kez algoritmaların çalıştırılmasından elde edilen metrik değerler Tablo 4.8'deki gibidir. Algoritmalar için en iyi, en kötü, doğruluk, kararlılık, zaman ve standart sapma metriklerinin matematiksel değerlerine göre sıralamaları yapılmış olup tablonun son satırında da ortalama sıralama değerleri hesaplanarak verilmiştir. Sphere için en iyi başarımların değerlerini DE ve PSO vermektedir. Şekil 4.8'de her boyutta 30 koşma boyunca

algoritmaların iterasyona göre elde ettiđi ortalama en iyi ölçüt deđerlerinin grafikleri sađ sütunda, en iyi çözüme ulaşan kořma için en iyi ölçüt deđerlerinin deđişim grafikleri sol sütunda verilmiştir. Boyut sayısı yükseldikçe CSO'nun en iyi deđere ulaşması gecikmekte, PSO ise her boyutta en iyi yakınsama eğilimi gösteren algoritma olmaktadır.

Sphere işlevi için ABC diđer işlevlerde de olduğu gibi en son sırada olup kullanımı çok da başarılı olmayacak bir algoritmadır. Ortalama sıralamalarına göre metriklerin toplu olarak karşılaştırılması sonucu en iyi olan algoritma PSO, ikinci olan ise DE'dir.

Tablo 4.8. Sphere işlevi için elde edilen metrik değerler tablosu

		ABC		BBO		CSO		DE		ICA		PSO	
		Değer	Sıra	Değer	Sıra	Değer	Sıra	Değer	Sıra	Değer	Sıra	Değer	Sıra
Sphere n=2	En iyi	6,1864 10 ⁻⁵	6	2,2927 10 ⁻⁷	5	3,9944 10 ⁻⁸	4	4,194 10 ⁻²³	1	1,176 10 ⁻¹²	3	5,050 10 ⁻¹⁵	2
	En kötü	27,1793	6	0,0042	4	0,0477	5	8,594 10 ⁻²¹	1	2,5798 10 ⁻⁷	2	4,4089 10 ⁻⁶	3
	Doğruluk	0,9792	1	0,5017	5	0,9621	3	0,9580	4	0,5453	6	0,9743	2
	Kararlılık	0,0039	3	0,1264	6	1,0951 10 ⁻⁴	1	0,0027	2	0,0151	5	0,0070	4
	Zaman	0,0963	6	0,0417	2	0,0430	1	0,0619	3	0,0748	5	0,0752	4
	Standart Sapma	0,1435	1	0,2511	3	0,4170	6	0,3163	5	0,3074	4	0,2392	2
Sphere n=5	En iyi	0,0052	6	2,8311 10 ⁻⁶	3	0,0023	5	1,2986 10 ⁻⁸	2	3,6691 10 ⁻⁶	4	4,6627 10 ⁻⁹	1
	En kötü	49,5602	6	0,0182	4	0,2281	5	2,3917 10 ⁻⁷	1	3,6691 10 ⁻⁶	2	8,5475 10 ⁻⁵	3
	Doğruluk	0,8684	2	0,7866	4	0,7575	5	0,8238	3	0,0484	6	0,8917	1
	Kararlılık	0,0062	2	0,0336	6	0,0013	1	0,0084	3	0,0086	4	0,0186	5
	Zaman	0,1202	6	0,0860	3	0,0617	1	0,0781	2	0,0923	4	0,0980	5
	Standart Sapma	1,1903	6	3,1990 10 ³	5	6,2416 10 ³	1	4,1070 10 ³	4	5,8823 10 ³	2	4,3029 10 ³	3
Sphere n=10	En iyi	0,1115	5	5,9338 10 ⁻⁴	2	0,2088	6	0,0014	3	0,0109	4	4,2857 10 ⁻⁵	1
	En kötü	109,9104	6	0,0557	4	3,1521	5	0,0064	2	0,0109	3	7,6328 10 ⁻⁴	1
	Doğruluk	0,5384	3	0,5756	2	0,3327	5	0,5053	4	0,0053	6	0,7050	1
	Kararlılık	0,0058	3	0,0201	5	0,0014	2	0,0097	4	0,0013	1	0,0288	6
	Zaman	0,0940	5	0,1107	6	0,0462	1	0,0633	2	0,0715	3	0,0781	4
	Standart Sapma	8,3924	3	6,3245	2	12,1380	6	11,7195	5	10,8250	4	5,2166	1
Ortalama Sıralama		4,16		3,54		3,5		2,83		3,77		2,72	



Şekil 4.8. Sphere işlevi için algoritmaların 30 koşma boyunca elde ettiği ortalama en iyi ölçüt değerleri değişimi (sağ sütun), en iyi maliyet değerini veren koşma için en iyi ölçüt değerleri değişimi (sol sütun)

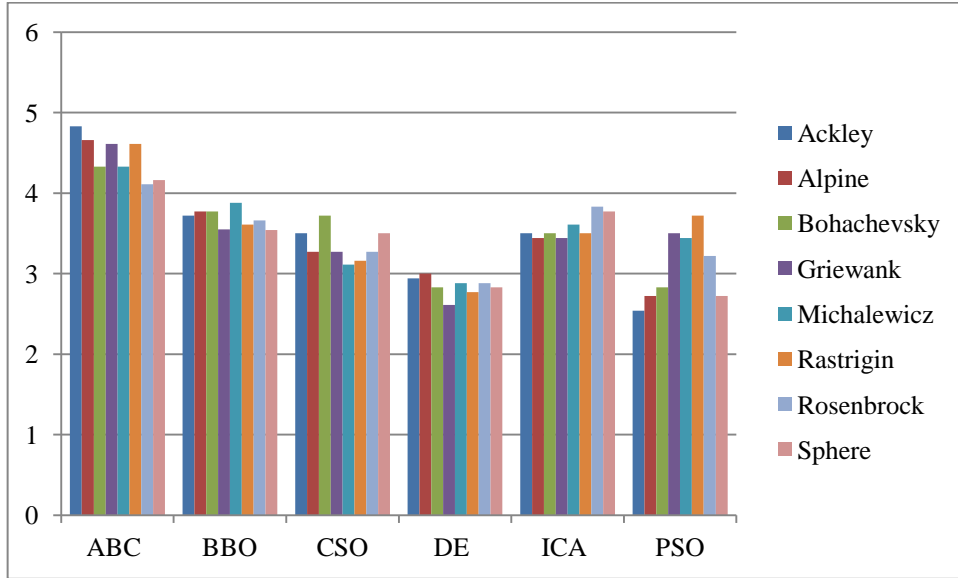
5. SONUÇLAR VE ÖNERİLER

Önceki bölümde, tanımlanan 6 metriğe göre algoritmaların başarımları her bir denektaşı işlev için ayrı ayrı detaylıca tartışıldı. Genel olarak her bir işlev için verilen tablolardan da görüleceği üzere en iyi başarımları PSO ve DE'nin verdiğini söyleyebiliriz. Daha sonrasında CSO ve ICA gelmekte, BBO ve ABC ise genel olarak en sonuncu olmaktadır. Süre, doğruluk ve kararlılık metrikleri bakımından CSO ve DE başarılı iken PSO da en iyi ölçütü bulmakta ve en iyi yakınsama davranışı göstermektedir.

Algoritmaların kümülatif olarak başarımlarının değerlendirilmesi için, algoritmaların 8 denektaşı işlev üzerinde gösterdikleri başarımlardan hesaplanan ortalama sıralamaları toplu olarak Tablo 5.1'de verilmiştir. Tablonun son satırında verilen kümülatif ortalama başarımların sıralama değerlerinden matematiksel olarak ve Şekil 5.1'de verilen bar grafikten de görsel olarak görüleceği üzere, bu çalışmada ele alınan denektaşı işlevlerin küresel minimum değerini aldığı konumlarının belirlenmesi sorunu için en iyi başarımları DE algoritmasının gösterdiği açıkça görülmektedir. İkinci en iyi kümülatif başarımları veren PSO, 8 denektaşı işlev içerisinde Ackley, Alpine ve Sphere işlevlerinde en iyi ortalamaya sahip iken ki buna en yakın ikinci ortalamaya sahip DE diğer işlevlerde en iyi kümülatif ortalamaya sahip olmaktadır. Hızlı çalışan CSO algoritması Griewank, Michalewicz ve Rastrigin işlevleri için PSO'ya göre daha iyi ortalama sırada olmakta ancak diğerlerinde PSO'nun bir hayli gerisinde kaldığı için kümülatif ortalama da üçüncülüğü almaktadır. Rosenbrock ve Sphere harici tüm işlevlerde ortalama olarak ICA'nın gerisinde kalan BBO kümülatif ortalama sıralama değerinde beşinci, ICA ise dördüncü olmaktadır. ICA ve BBO her işlev için ayrı ayrı ortalama sıralama değerinde de kümülatif ortalama sıralama değerinde de birbirine en yakın değerlere sahip olup üzerinde çalıştığımız işlevlerde birbirine en yakın sonucu veren iki algoritma olduğu ancak ICA'nın daha başarılı olduğu gözlemlenmektedir. Kümülatif ortalama sıralamasında ve test amaçlı çalıştığımız her işlevde en kötü ortalama değerleri veren ABC'nin şekil ve tablolardan da görüldüğü üzere seçtiğimiz 6 algoritma içerisinde sonuncu olduğu alenen görülmektedir.

Tablo 5.1. *Algoritmaların denektaşı işlevler için hesaplanan ortalama sıralama değerler tablosu*

	ABC	BBO	CSO	DE	ICA	PSO
Ackley	4,83	3,72	3,5	2,94	3,5	2,54
Alpine	4,66	3,77	3,27	3	3,44	2,72
Bohachevsky	4,33	3,77	3,72	2,83	3,5	2,83
Griewank	4,61	3,55	3,27	2,61	3,44	3,5
Michalewicz	4,33	3,88	3,11	2,88	3,61	3,44
Rastrigin	4,61	3,61	3,16	2,77	3,5	3,72
Rosenbrock	4,11	3,66	3,27	2,88	3,83	3,22
Sphere	4,16	3,54	3,5	2,83	3,77	2,72
Kümülatif Ortalama Sıralama Değeri	4,455	3,6875	3,35	2,8425	3,57375	3,08625



Şekil 5.1. *Algoritmaların denektaşı işlevler için hesaplanan ortalama sıralama değerler bar grafikleri*

KAYNAKÇA

- Akça, M.R. (2011). *Yapay Arı Kolonisi Algoritması Kullanılarak Gezgin Satıcı Probleminin Türkiye'deki İl Ve İlçe Merkezlerine Uygulanması*. Yayımlanmamış Yüksek Lisans Tezi. Konya: Selçuk Üniversitesi.
- Akyol, S. and Alataş, B. (2012). Güncel Sürü Zekâsı Optimizasyon Algoritmaları, *Neşehir Üniversitesi Fen Bilimleri Enstitü Dergisi*, 1 , 36-50.
- Altunbey, F. and Alataş, B. (2015). Sosyal Ağ Analizi İçin Sosyal Tabanlı Yapay Zekâ Optimizasyon Algoritmalarının İncelenmesi. *International Journal of Pure and Applied Sciences*, 1(1), 33-52.
- Ammu, P.K., Sivakumar, K.C. and Rejimoan, R. (2013). Biogeography-based optimization - A Survey. *Int J Electron Comput Sci Eng*, 2(1), 154-160.
- Babayiğit ,B. and Özdemir, R. (2012). Modifiye Yapay Arı Koloni Algoritması ile Nümerik Fonksiyon Optimizasyonu. *ELECO '2012 Elektrik - Elektronik ve Bilgisayar Mühendisliği Sempozyumu*, Bursa, 618-622.
- Eser, M. and Yüzgeç, U. (2014). Kaotik Tabanlı Diferansiyel (Farksal) Gelişim Algoritması , *ISITES2014 Karabuk*, 201-210.
- Gargari, E. and Lucas, C. (2007). Imperialist Competitive Algorithm: An Algorithm for Optimization Inspired by Imperialistic Competition, *2007 IEEE Congress on Evolutionary Computation*, 4661-4667.
- Gong, W., Cai, Z., Ling, C. X., Li, H. (2010). A real-coded biogeography-based optimization with mutation. *Applied Mathematics and Computation*, 216(9), 2749-2758.
- Anonim, Guguk kuşu hakkında ansiklopedik bilgi, Türkiye, http://www.turkcebilgi.com/ansiklopedi/guguk_kusu (Erişim Tarihi: 23.04.2017).
- Hansen, N., Ros, R., Mauny, N., Schoenauer, M., Auger, A. (2011). Impacts of invariance in search: When CMA-ES and PSO face ill-conditioned and non-separable problems. *Applied Soft Computing*, 11(8), 5755-5769.
- Jamil, M. and Yang, X. (2013). A Literature Survey of Benchmark Functions For Global Optimization Problems. *International Journal of Mathematical Modelling and Numerical Optimisation*, 4(2), 150-194.
- Karaboga, D. (2005). *An Idea Based On Honey Bee Swarm For Numerical Optimization*. Technical report-tr06. Kayseri: Erciyes University, Engineering Faculty, Computer Engineering Department.
- Karagül, K. (2014). Guguk Kuşu Algoritması: Bir Plastik Atık Toplama Uygulaması, *15th International Symposium on Econometrics, Operations Research and Statistics*, Suleyman Demirel University, 775-784.

- Kennedy, J. and Eberhart, R. C. (1995). Particle Swarm Optimization. *Proc. of the IEEE Int. Conference on Neural Networks*, 4, 1942-1948.
- Keskintürk, T. (2006). Diferansiyel Gelişim Algoritması. *İstanbul Ticaret Üniversitesi Fen Bilimleri Dergisi*, Yıl: 5, Sayı: 9, 85-99.
- Molga, M. and Smutnicki, C. (2005). Test functions for optimization needs. *Test functions for optimization needs*, <http://new.zsd.iar.pwr.wroc.pl/files/docs/functions.pdf> (Erişim Tarihi: 23.04.2017).
- Nobile, M. S., Pasi, G., Cazzaniga, P., Besozzi, D., Colombo, R., Mauri, G. (2015, August). Proactive Particles in Swarm Optimization: A self-tuning algorithm based on Fuzzy Logic. In *Fuzzy Systems (FUZZ-IEEE), 2015 IEEE International Conference on*, 1-8.
- Özdemir, M. (2013). *Zaman Kısıtı Altında Takım Oryantiring Problemlerinin Yapay Arı Kolonisi Yaklaşımı İle Çözümü*. Yayınlanmamış Yüksek Lisans Tezi. İstanbul: İstanbul Üniversitesi.
- Rahnamayan, S., Tizhoosh, H. R. and Salama, M. M. (2007). A novel population initialization method for accelerating evolutionary algorithms. *Computers & Mathematics with Applications*, 53(10), 1605-1614.
- Rosenbrock, H.H. (1960). An automatic method for finding the greatest or least value of a function. *The Computer Journal*, 3(3), 175-184.
- Saraçoğlu, B., Güvenç, U., Dursun M., Poyraz G., Duman S. (2013). Biyocoğrafya Tabanlı Optimizasyon Metodu Kullanarak Asenkron Motor Parametre Tahmini. *İleri Teknoloji Bilimleri Dergisi*, 2(1), 46-54.
- Simon, D. (2008). Biogeography-based optimization. *IEEE transactions on evolutionary computation*, 12(6), 702-713.
- Storn, R. and Price, K. (1997). Differential evolution-A simple and efficient heuristic for global optimization over continuous spaces. *J. Global Optimization*, 11, 341-359.
- Tamer, S. and Karakuzu, C. (2006). Parçacık Sürüsü Optimizasyon Algoritması ve Benzetim Örnekleri. *ELECO 2006 Elektrik-Elektronik-Bilgisayar Sempozyumu, Elektronik Bildirileri Kitabı*, s. 302-306.
- Weicker, K. (2003). *Farm Evolutionary algorithms and dynamic optimization problems*. Yayınlanmamış Doktora Tezi. Berlin: Stuttgart Üniversitesi.
- Xu, S., Wang, Y. and Huang A. (2014). Application of Imperialist Competitive Algorithm on Solving the Traveling Salesman Problem. *Algorithms*, 7(2), 229-242.

Yang, X.S. and Deb, S. (2009). Cuckoo search via Lévy flights. *2009 World Congress on Nature & Biologically Inspired Computing*, 210-214.

Yang, X. S. (2014). *Nature-inspired optimization algorithms*. Londra:Elsevier.

Zorarpacı, E. and Özel, S.A. (2004). Nitelik Seçme Problemi İçin Diferansiyel Gelişim Algoritması Ve Yapay Arı Kolonisi Optimizasyon Tekniğini Kullanan Melez Yöntem. *DEÜ Mühendislik Fakültesi Mühendislik Bilimleri Dergisi*, 16 (48), 49-60.