



T.C.

SELÇUK ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

ÇOKLU İMGE EŞİKLEME PROBLEMLERİNDE METASEZGİSEL ALGORİTMALARIN PERFORMANS ANALİZİ

Gökhan PEKDEMİR

YÜKSEK LİSANS TEZİ

Bilgisayar Mühendisliği Anabilim Dalı

Şubat-2012
KONYA
Her Hakkı Saklıdır

ÖZET

YÜKSEK LİSANS TEZİ

ÇOKLU İMGE EŞİKLEME PROBLEMLERİNDE METASEZGİSEL ALGORİTMALARIN PERFORMANS ANALİZİ

Gökhan PEKDEMİR

Selçuk Üniversitesi Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Yrd. Doç. Dr. Ömer Kaan BAYKAN

2012, 68 Sayfa

Jüri
Yrd. Doç. Dr. Ömer Kaan BAYKAN
Yrd. Doç. Dr. Nihat YILMAZ
Yrd. Doç. Dr. Mesut GÜNDÜZ

İmge bölütleme, bir imgeyi birbiriyle örtüşmeyen alt imge gruplarına ayırma işlemidir. Çok seviyeli imge eşikleme, en popüler imge bölütleme tekniklerinden biridir ve sıklıkla amaç fonksiyonunun optimizasyonu problemi olarak görülmektedir. Bu çalışmada, çoklu imge eşikleme problemini çözmek amacıyla imgenin gri seviye dağılımını (histogram) kullanan yöntemlerden Otsu'nun yöntemi amaç fonksiyonu olarak kullanılmıştır. En uygun eşik değerlerini belirlemek amacıyla seçilen test imgeleri üzerinde Parçacık Sürü Optimizasyonu (PSO), Ateş böceği Algoritması (AA), Guguk kuşu Optimizasyonu Algoritması (GOA), Etraflı Arama ile Otsu'nun amaç fonksiyonunun hem minimize (sınıf içi değişinti) edildiği hem de maksimize (sınıflar arası değişinti) edildiği deneyler gerçekleştirilmiştir. Yöntemlerin performansları stabilite, çözüm kalitesi ve yakınsama süreleri açısından kıyaslanmıştır. Çözüm kalitesi ve stabilite açısından genellikle GOA'nın, yakınsama süresi açısından PSO'nun daha başarılı olduğu gözlenmiştir.

Anahtar Kelimeler: Ateş böceği Algoritması, Çoklu İmge Eşikleme Problemi, Etraflı Arama, Guguk kuşu Optimizasyonu Algoritması, Otsu'nun yöntemi, Parçacık Sürü Optimizasyonu

ABSTRACT

MS THESIS

A PERFORMANCE ANALYSIS OF METAHEURISTIC ALGORITHMS ON THE MULTILEVEL THRESHOLDING PROBLEMS

Gökhan PEKDEMİR

THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCE OF SELÇUK UNIVERSITY
THE DEGREE OF MASTER OF SCIENCE
IN COMPUTER ENGINEERING

Advisor: Asst. Prof. Dr. Ömer Kaan BAYKAN

2012, 68 Pages

Jury
Asst. Prof. Dr. Ömer Kaan BAYKAN
Asst. Prof. Dr. Nihat YILMAZ
Asst. Prof. Dr. Mesut GÜNDÜZ

Image segmentation is a process that separates the image to non-overlapping sub-image groups. Multilevel image thresholding is one of the most popular image segmentation techniques and it's often seen as an optimization problem of an objective function. In this paper, Otsu's method, which is one of the methods that use the gray-level distribution (histogram) of image, is used as an objective function to solve the multilevel image thresholding problem. To consider the optimal threshold values on the selected test images, several experiments were performed by both minimizing (within-class variance) and maximizing (between-class variance) the Otsu's objective function with Particle Swarm Optimization (PSO), Firefly Algorithm (FA), Cuckoo Optimization Algorithm (COA) and Exhaustive Search. Performances of the metaheuristic techniques were compared in terms of stability, solution quality and convergence time. In general, it was observed that GOA is more successful than others with respect to solution quality and stability; PSO is more successful than others with respect to convergence time.

Keywords: Cuckoo Optimization Algorithm, Exhaustive Search, Firefly Algorithm, Multilevel Image Thresholding Problem, Otsu's method, Particle Swarm Optimization

ÖNSÖZ

Yüksek lisans öğrenim ve tez sürecimin tüm aşamalarında engin bilgi birikimini, deneyimini ve ilgisini hiçbir zaman esirgemeyen değerli hocam ve danışmanım Sayın Yrd. Doç Dr. Ömer Kaan Baykan'a ve çalışmamın şekillenmesine yardımcı olan Selçuk Üniversitesi Mühendislik-Mimarlık Fakültesi Bilgisayar Mühendisliği öğretim üyelerine, araştırma görevlilerine sonsuz teşekkürlerimi sunarım.

Gökhan PEKDEMİR
KONYA-2012

İÇİNDEKİLER

ÖZET ..... **iv**

ABSTRACT ..... **v**

ÖNSÖZ ..... **vi**

İÇİNDEKİLER ..... **vii**

SİMGELER VE KISALTMALAR ..... **ix**

1. GİRİŞ **1**

1.1. Tezin Amacı 3
1.2. Tezin Organizasyonu 3

2. KAYNAK ARAŞTIRMASI **4**

2.1. İmge Eşikleme 4
2.2. Metasezgisel Algoritmalar 5
2.3. Çoklu İmge Eşikleme Probleminde Metasezgisel Algoritmalar 6

3. İMGE EŞİKLEME **8**

3.1. Otsu'nun Yöntemi 9
3.2. Çoklu İmge Eşikleme 11
3.3. Gri Ton Düzgünlüğü ve Düzensizliği Ölçütü 13
3.4. Etraflı (Tam) Arama 14

4. METASEZGİSEL ALGORİTMALAR **16**

4.1. Guguk Kuşu Optimizasyonu Algoritması 17
4.1.1. Popülasyonu başlatma 18
4.1.2. Yumurtlama süreci 19
4.1.3. Guguk kuşlarının göç süreci 20
4.1.4. Akış şeması 23
4.2. Ateş Böceği Algoritması 24
4.2.1. Popülasyonu başlatma 24
4.2.2. Etkileşimi belirleme 25
4.2.3. Konum güncelleme 26
4.2.4. Akış şeması 27
4.3. Parçacık Sürü Optimizasyonu Algoritması 28
4.3.1. Popülasyonu başlatma 28
4.3.2. Hız ve konum güncelleme 30
4.3.3. Akış şeması 31

5. DENEYSEL SONUÇLAR VE PERFORMANS ANALİZİ **32**

5.1. Sınıf İçi Değişinti ile İmge Eşikleme Uygulaması 32

5.2. Sınıflar Arası Değişinti ile İmge Eşikleme Uygulaması 49

6. SONUÇLAR VE ÖNERİLER 62

6.1. Sonuçlar 62
6.2. Öneriler 63

KAYNAKLAR 64

ÖZGEÇMİŞ 68

SİMGELER VE KISALTMALAR

Simgeler

C_i : İmgedeki i. sınıf
c_1, c_2 : Öğrenme faktörleri
f_O : Amaç fonksiyonunun maliyeti
f_i : i. gri seviyenin tekrarlanma sıklığı
f_i^* : i. turda elde edilen en iyi çözüm
f_{opt} : Amaç fonksiyonunun genel en iyi değeri
f_{ort} : Her bir i. tura ait f_i^* çözümlerinin ortalaması
g_{best} : En iyi parçacık konumu
g_{max} : İmgedeki en büyük gri seviye değeri
g_{min} : İmgedeki en küçük gri seviye değeri
I : İmge
$I(x, y)$: x,y koordinatına sahip pikselin gri seviye değeri
K : İmgedeki sınıf sayısı
L : İmgedeki gri seviyeleri temsil eden tamsayılar kümesi
MxN : İmgenin boyutları
N_b : Problemin boyutu
N_{max} : Maksimum popülasyon sayısı
N_{pop} : Popülasyon sayısı
n_i : i. turda gerçekleştirilen iterasyon sayısı
n_{ort} : Her bir i. tura ait n_i değerlerinin ortalaması
O( ) : Büyük O notasyonu
p_{best_i} : i. parçacığın en iyi konumu
p_i : i. gri seviyenin olasılık dağılımı
$r_{i,j}$: x_i ve x_j çözümleri arasındaki uzaklık
S : Standart sapma
T_{max}^* : f_i^* çözümlerinden uyguluk değeri en iyi tur çözümünün eşik değerleri
t_i : i. tura ait hesaplama süresi
t_{ort} : her bir i. tura ait t_i değerlerinin ortalaması
u_T : İmgenin yoğunluklu ortalama gri seviye değeri
u_i : i. sınıfın yoğunluklu ortalama gri seviye değeri
v_i : i. parçacığın hızı
w : Parçacığın eylemsizlik ağırlığı
x_i : i. ateş böceği çözümü
(x, y) : İmgenin uzamsal koordinatları
ω_i : i. sınıfın olasılık yoğunluğu
α : [0 1] aralığında rastgele değer
β : x_i çözümünün x_j üzerinde etkililiği
β_0 : Etkililik parametresi
γ : Işık soğurma katsayısı
ε : Hata miktarı
λ : Hedef ortama uzaklık katsayısı
φ : Sapma açısı
$\sigma^2{}_i$: i. sınıfın değişinti değeri

Kısaltmalar

AA	: Ateş böceği Algoritması
BBA	: Bakteriyel Besin Algoritması
BÇA	: Bal Arısı Çiftleşme Algoritması
DEA	: Diferansiyel Evrim Algoritması
GA	: Genetik Algoritmalar
GOA	: Guguk kuşu Optimizasyonu Algoritması
GTD	: Gri Ton Düzgünlüğü
GTDS	: Gri Ton Düzensizliği
KHK	: Kümeleme hata kriteri
KKA	: K-Ortalama Kümeleme Algoritması
KKO	: Karınca Koloni Optimizasyonu
PSO	: Parçacık Sürü Optimizasyonu
STA	: Simüle Edilmiş Tavlama Algoritması
TAA	: Tabu Arama Algoritması
YY	: Yumurtlama yarıçapı

1. GİRİŞ

İmge bölütleme, bir imgeyi birbiriyle örtüşmeyen alt imge gruplarına ayırma işlemidir (Sezgin, 2002). İmge eşikleme, imge bölütleme işlemlerinde kullanılan yöntemlerden biridir. İmge, ikili eşikleme olarak adlandırılan yöntemle iki gri seviye grubuna veya çoklu eşikleme olarak adlandırılan yöntemle ikiden fazla gri seviye grubuna ayrılabilmektedir. İmgeyi gruplara ayırabilmek için uygun eşik değerinin tespit edilmesi amacıyla geliştirilmiş birçok yöntem mevcuttur. Teknik yazında farklı temellere dayanan ve bazıları uygulamaya özgü geliştirilmiş, eşikleme sırasında kullandıkları bilgilere göre 6 ana başlığa ayrılan, birçok imge eşikleme yöntemi bulunmaktadır (Sezgin ve Sankur, 2004).

Hem ikili ve hem de çoklu eşikleme yöntemleri parametrik ve parametrik olmayan (kritere dayalı) eşikleme yöntemleri olarak ikiye ayrılmaktadır (Sezgin, 2002). Parametrik eşikleme yöntemlerinde hesaplama süresi göreceli olarak yüksektir ve kabul edilen model parametreleri ile imge histogramı arasındaki farkın artması ile başarımı azalır (Sezgin, 2002). Parametrik olmayan yöntemler, sınıf içi değişinti (Otsu, 1979), entropi (Pun, 1980; Kapur ve ark., 1985), bulanık benzerlik ölçütü (Li ve ark., 1995), Bayes hatası (Kittler ve Illingworth,1986) vb. bir kriteri optimal hale getiren eşik değerlerini belirlemeyi temel alır. Parametrik olmayan eşikleme yöntemlerinin daha doğru sonuçlar verdiği teknik yazında belirtilmektedir (Yen ve ark., 1995). Ancak parametrik olmayan (kritere dayalı) yöntemlerin iki eksikliği bulunmaktadır. Bunlardan biri imgedeki sınıf sayısını otomatik olarak belirlemenin zor olması ve bunun genellikle kullanıcı tarafından belirlenmesinin gerekmesi; ikincisi de çok seviyeli (çoklu) eşiklemede hesaplama süresinin uzun olmasıdır (Sezgin, 2002).

Teknik yazında ikili imge eşikleme amacıyla bir çok yöntem önerilmiş (Chu ve ark., 2004; Gonzales-Baron ve Butler; 2006) ve sonrasında uygulamalarda daha kesin ve iyi bir bölütleme elde etmek amacıyla çoklu imge eşikleme gerçekleştirmek üzere uyarlanmıştır. Bu yöntemlerin temel problemi, çoklu imge eşikleme problemine uyarlandığı zaman etraflı (tam) aramaya bağlı olarak hesaplama karmaşıklığının üstel olarak artmasıdır (Sathya ve Kayalvizhi, 2011). Çoklu imge eşikleme probleminde etraflı (tam) arama ile ikili eşikleme yöntemi gerçekleştirildiği takdirde L gri seviyeye sahip bir imgeyi k adet eşik değeri ile bölütlemek istenirse yöntemin hesaplama karmaşıklığı üstel bir şekilde büyüyerek $O(L^{k-1})$ şeklinde ifade edilir (Hammouche ve

ark., 2010). Büyük O notasyonu algoritmanın büyüme hızını gösterir ve zaman karmaşıklığını belirtir.

Bu durumun üstesinden gelmek amacıyla birtakım çalışmalar gerçekleştirilmiştir:

- Otsu'nun yönteminde olduğu gibi belirli bir fonksiyonun hesaplama aşamalarını hızlandırmak amaçlı teknikler (Lin, 2001; Liao ve ark., 2001; Dong ve ark, 2008).
- Ardışık sınıflandırma olarak tabir edilen, imgenin ikili eşikleme ile iki ayrı dağılıma bölünmesi ve kriter sağlanana kadar varyansı büyük olan dağılımın tekrar ikili eşiklemeye tabi tutulması (Yen ve ark., 1995; Sezgin ve Tasaltin, 2000). Bu sınıflandırma teknikleri hızlı algoritmalar olmasına rağmen yetersiz tekniklerdir ve en uygun eşik değeri değerlerini vermemektedir (Hammouche ve ark, 2010).
- Hesaplama aşamalarını hızlandırmak amaçlı tekniklerden bazılarında tekrarlamalı bir yapı sunulmuştur (Yin ve Chen, 1997). Kapur'un sınıflar arası entropiyi maksimize etmeye dayalı yöntemi, iteratif bir algoritma kullanılarak gerçekleştirilmiştir. (Luo ve Tian, 2000).

Son yıllarda, hızlı bir şekilde çoklu imge eşikleme gerçekleştirmek üzere belirtilen yöntemlere alternatif olarak metasezgisel algoritmalar kullanılmaktadır. Metasezgisel, alt sezgiselin işleyişini, verimli bir şekilde yüksek kaliteli çözümler üretmesi amacıyla değiştiren ve yönlendiren tekrarlamalı bir üst süreçtir (Blum ve Roli, 2003). Bu alt sezgisel, düşük veya yüksek seviyeli bir işlem, basit bir yerel arama, yapıcı bir yöntem olabilir (Blum ve Roli, 2003). Metasezgisel algoritmalar, zamanında yerel en iyiden uzaklaşma, yüksek kaliteli çözümler üretme gibi avantajlar sağlar. (Hammouche ve ark., 2010). Çoklu eşikleme problemi lineer olmayan bir optimizasyon problemidir ve birçok yerel en iyiler içerebilmektedir.

Teknik yazında, Genetik Algoritmalar (GA), Karınca Koloni Optimizasyonu (KKO), Parçacık Sürü Optimizasyonu (PSO), Tabu Arama Algoritması (TAA), Diferansiyel Evrim Algoritması (DEA), Simüle Edilmiş Tavlama Algoritması (STA), Bal Arısı Çiftleşme Algoritması (BÇA), Bakteriyel Besin Algoritması (BBA) ile Otsu'nun (1975) veya Kapur'un (1985) yöntemi birleştirilerek çoklu eşikleme gerçekleştirilen birçok çalışma mevcuttur (Yin ve Chen, 1999; Zhiwei ve ark., 2006; Maitra ve Chatterjee, 2008; Sathya ve Kayalvizhi, 2011; Hammouche ve ark., 2010). Bu çalışmaların yanı sıra teknik yazında var olan ancak çoklu imge eşikleme

problemlerinde kullanılmamış birtakım yeni metasezgisel algoritmalar mevcuttur (Rajabioun, 2011).

1.1. Tezin Amacı

Bu çalışmanın amacı, yeni geliştirilen bazı metasezgisel teknikleri; Otsu'nun sınıflar içi değişintiyi minimize etme ve sınıflar arası değişintiyi maksimize etmeye dayalı yöntemleri ile bir arada kullanarak çoklu imge eşikleme problemine çözüm aramak, önerilen yöntemlerin performansını belirlenen kriterlere uygun olarak değerlendirmektir. Bu amaçla Ateş böceği Algoritması (AA) (Yang, 2008), Guguk kuşu Optimizasyonu Algoritması (GOA) (Rajabioun, 2011) ile Parçacık Sürü Optimizasyonu (PSO) (Kennedy ve Eberhart, 1995), Otsu'nun yöntemine ait amaç fonksiyonu ile birleştirilerek imge eşikleme işlemi gerçekleştirilmiştir. Yöntemlerin performansı; işlem zamanı, stabilitesi (gürbüzlüğü) ve çözüm kalitesi bakımından standart test imgeleri üzerinde değerlendirilmiştir.

1.2. Tezin Organizasyonu

Çalışma kapsamında, ikinci bölümde imge eşikleme problemi ve metasezgisel algoritmalar ile ilgili teknik yazında yer alan çalışmalar incelenmektedir. Üçüncü bölümde imge eşikleme hakkında bilgi verilmekte ve Otsu'nun imge eşikleme yöntemi ile çoklu imge eşikleme problemi açıklanmaktadır. Dördüncü bölümde PSO, GOA ve AA kapsamlı bir şekilde açıklanmakta ve tanımlanmaktadır. Beşinci bölümde bu üç algoritmanın Otsu yöntemi ile birlikte çoklu imge eşikleme problemine uyarlanması açıklanmış, test imgeleri üzerine uygulanması sonucu elde edilen sonuçlar tartışılmıştır. Son bölümde ise araştırma sonuçlarının genel bir değerlendirmesi yapılarak, gelecekte gerçekleştirilebilecek çalışmalara yönelik öneriler sunulmaktadır.

2. KAYNAK ARAŞTIRMASI

Teknik yazında yer alan kaynaklar; imge eşikleme, metasezgisel algoritmalar, çoklu imge eşiklemede metasezgisel algoritmalar olarak 3 ayrı başlıkta incelenmiştir.

2.1. İmge Eşikleme

Otsu (1979), imge eşiklemede parametrik olmayan ve daha önce denenmemiş bir yöntem önermiştir. Bu yöntemde en uygun eşik değerinin belirlenmesi, nesne ve arka plan piksellerine ilişkin ağırlıklandırılmış toplam sınıf içi değişintilerin minimize edilmesi ile gerçekleştirilmiştir.

Pun (1980), bir imgeyi bölütlemek amacıyla imgenin gri seviye histogramının entropisini temel alan otomatik eşik seçim yöntemi önermiştir. Bu yöntem ile imge, ikili eşikleme olarak tabir edilen nesne ve arka plan sınıflarına, histogram entropisinin maksimize edilmesiyle ayrılmaktadır. Ayrıca çalışmada mevcut yöntemin çoklu eşiklemeye ve çok boyutlu histogramlara uyarlanabilirliği tartışılmıştır.

Kapur ve ark. (1985), Pun tarafından önerilen iki entropi tabanlı yöntemi, imge eşikleme üzerinde dikkatlice analiz etmiş ve entropi hesabı aşamasında nesne ve arka plana ait gri seviyelerin olasılık yoğunluklarını daha basit bir hale indirgeyerek imge eşikleme için yeni bir maksimum entropi kuralı sunmuştur.

Pal ve Pal (1993), teknik yazında önerilen; gri seviye histogramını, bulanık küme yaklaşımlarını, uzamsal ilişkileri kullanan imge eşikleme tekniklerini gözden geçirmiş ve özetlemiştir.

Yin ve Chen (1997), en uygun eşikleme yöntemlerini daha pratik hale getirmek amacıyla tekrarlanan bir yapı önermiştir. Önerilen tekrarlamalı yaklaşımda algoritma ikili eşikleme ile başlamakta ve elde edilen başlangıç sonuçlarını kullanarak çoklu eşikleme elde etmektedir. Bu işlem, iki ardışık iterasyona ait amaç fonksiyonu değerleri arasındaki fark değişmeyene kadar devam eder. Böylece algoritma çalışma zamanını azaltırken eşik değeri sayısını da otomatik olarak belirlemektedir.

Sezgin (2002), teknik yazında var olan 44 farklı eşikleme yöntemi hakkında açıklayıcı bilgiler sunmuştur. İkili imge eşikleme ve çoklu imge eşikleme yöntemleri hakkında bilgi verilmiş ve ikili eşikleme yöntemlerine dayalı imge analizi uygulamaları gerçekleştirilmiştir. Ayrıca imge bölütleme değerlendirmede kullanılan ölçütler

hakkında bilgiler verilmiş, ikili ve çoklu eşikleme yöntemlerinin değerlendirmesine yönelik birer başarım ölçütü tanımlanmıştır.

Sezgin ve Sankur (2004), teknik yazındaki imge eşikleme yöntemlerini incelenmiş, formülize etmiş, kategorize etmiş ve performanslarını kıyaslamıştır. 40 adet eşikleme yöntemi gruplara ayrılmış ve performans ölçütleri ile imgeler üzerine uygulanarak performans analizi gerçekleştirilmiştir.

Eichmann ve Lüssi (2005), imgeyi çoklu eşikleme amacıyla teknik yazında kullanılan ikili eşikleme yöntemlerinden Otsu'nun (1979) ve Kapur'un (1985) da içinde bulunduğu 4 farklı yöntemi amaç fonksiyonu olarak kullanmıştır. Eşik değeri tespitinde taraflı (tam) arama kullanılması durumunda meydana gelen hesaplama karmaşıklığını azaltan dinamik programlama teknikleri sunmuşlardır. Ayrıca imge bölütlemede uygun sınıf sayısının otomatik olarak kararlaştırılması üzerine yöntemler önermişlerdir.

Lin (2001), çoklu eşikleme gerçekleştirmek amacıyla Otsu'nun sınıf içi değişinti fonksiyonunu amaç fonksiyonu olarak kullanmış ve çalışmalarında Otsu'nun yönteminin yarattığı hesaplama karmaşıklığını azaltan yöntem önermişlerdir.

2.2. Metasezgisel Algoritmalar

Kennedy ve Eberhart (1995), kuş sürülerinin davranışlarından esinlenerek PSO adlı zeki bir optimizasyon yöntemi önermiştir. Önerilen yöntemde her bir çözüm, çözüm uzayındaki bir kuşu temsil etmekte ve parçacık olarak adlandırılmaktadır. Her bir parçacık, uygulanan problemin fonksiyonuna bağlı olarak bir uygunluk değerine ve çözüm uzayında gezinmelerini kontrol eden bir hız vektörüne sahiptir ve parçacıklar hızlarını mevcut olan en iyi parçacığa göre yenilerler.

Shi ve Eberhart (1998) PSO tekniğine, eski hız vektörünün yeni hız vektörüne etkisini ayarlayan yeni bir parametre olan ağırlık katsayısı ekleyerek yöntemi iyileştirme yoluna gitmiş ve performansını değerlendirmiştir.

Yang (2008), kitabında PSO, GA, KKO, Fotosentetik Algoritma, Enzim Algoritması, Arı Algoritmaları, Harmonik Arama gibi doğa esintili modern metasezgisel algoritmaları anlatmış ve ateş böceklerinin ışıldama ve parlaklıklarına bağlı olarak aralarındaki etkileşim davranışını temel alarak geliştirdiği AA'yı açıklamıştır.

Yang (2009), AA'nın çok biçimli optimizasyon uygulamalarında kullanımını açıklamıştır, Algoritma performansını, diğer metasezgisel algoritmalarla birtakım benzetim uygulamaları gerçekleştirerek kıyaslamıştır.

Lukasik ve Zak (2009), sürekli ve kısıtlı optimizasyon üzerinde AA'nın performansını değerlendirmiştir. Algoritmayı, birtakım test fonksiyonları üzerinde PSO ile kıyaslamıştır. Ayrıca çalışmasında, algoritmada kullanılan tüm parametreleri belirli bir aralıkta test fonksiyonları üzerinde değerlendirmiş ve en iyi sonucu veren parametre değerlerini sunmuştur.

Yang (2010), AA'nın lineer olmayan tasarım problemlerine nasıl uygulanması gerektiğini açıklamıştır. Ayrıca çalışmasında yeni optimizasyon algoritmalarını test etmek amacıyla birkaç yeni test fonksiyonu önermiştir.

Rajabioun (2011), sürekli ve lineer olmayan optimizasyon problemleri için uygun olan GOA'yı geliştirmiştir. GOA'yı, guguk kuşlarının kuluçka asalağı olarak yumurtalarını başka yuvalara yumurtlama davranışını temel alarak geliştirmiştir. Algoritmayı birtakım test fonksiyonları üzerinde, GA ve PSO ile kıyaslayarak değerlendirmiştir.

2.3. Çoklu İmge Eşikleme Probleminde Metasezgisel Algoritmalar

Yang ve ark. (2003), GA tabanlı bağıl entropi çoklu eşikleme yöntemi önermiştir. Önerilen yöntemde bağıl entropi yöntemi uygunluk fonksiyonu olarak kullanılmıştır. Önerilen yöntem, klasik eşikleme yöntemleri ile deneysel olarak işlem zamanı bakımından karşılaştırılmıştır.

Wei ve Kangling (2008), Otsu'nun yönteminin çoklu eşikleme yöntemi olarak kullanılması durumunda eşik sayısının artmasıyla birlikte işlem zamanın üssel olarak arttığını ifade ederek Otsu'nun sınıf içi minimum değişinti kriteri ile PSO algoritmasını birleştirmiş, daha verimli ve işlem zamanını azaltan çoklu eşikleme yöntemi önermiştir.

Hammouche ve ark. (2010), çalışmasında çoklu imge eşikleme problemini çözmek amacıyla altı farklı metasezgisel tekniği (GA, KKO, PSO, TAA, DEA, STA) örnek test imgeleri üzerine uygulamış ve yöntemlerin performanslarını süreleri, eşik değerleri ve karesel hataları bakımından karşılaştırmıştır.

Sathya ve Katyalvizhi (2010), PSO ile Otsu'nun sınıflar arası değişinti yöntemi ve Kapur'un entropi yöntemini birleştirerek yeni bir çoklu imge eşikleme yöntemi

önermiştir. Önerdiği yöntemi GA ile süre, iterasyon sayısı ve çözüm kalitesi açısından 10 farklı test imgesi üzerinde karşılaştırmıştır.

Horng ve Jiang (2010), Ateşböceği Algoritması tabanlı yeni bir çoklu maksimum entropi eşikleme yöntemi önermiştir. Önerilen yöntemin performansı PSO, İşbirlikçi Öğrenme Yöntemi tabanlı PSO ve BÇA ile karşılaştırılmıştır.

Liang ve Yin (2011), Otsu'nun eşikleme yöntemini KKO ile birleştirerek yeni bir çoklu imge eşikleme yöntemi önermiştir. Önerilen yöntemin performansı hız ve güvenilirlik açısından değerlendirilmiştir.

Sathya ve Katyalvizhi (2011), Değiştirilmiş Bakteriyel Besin Algoritması ile Otsu'nun sınıflar arası değişinti yöntemi ve Kapur'un entropi yöntemini birleştirerek yeni bir çoklu imge eşikleme yöntemi önermiştir. Ayrıca önerdiği yöntemi; BBA, PSO ve GA ile hesaplama süresi ve stabilite açısından karşılaştırmıştır.

3. İMGE EŞİKLEME

İmge eşikleme, imge bölütme amacıyla kullanılan eski, basit ve popüler tekniklerden biridir (Pal ve Pal, 1993). İmge bölütleme, homojen bir küme içerisinde aynı karakteristiğe sahip alanların arka plandan ayrılması, belirgin bir hale getirilmesi işlemidir (Şengür ve ark., 2005).

İmge eşikleme, imgenin sadece parlaklık değerini dikkate alarak bölütleme gerçekleştiren işlemdir (Sezgin, 2002). Eşikleme işlemi, imgenin birden fazla özelliğini kullanarak bölütleme gerçekleştiren yöntemlere kıyasla daha hızlı sonuç vermektedir. Bu nedenle gerçek zamanlı çalışmayı gerektiren bölütleme uygulamalarında daha çok tercih edilmektedir. Ancak eşikleme işlemi her imge için iyi bir bölütleme sonucu döndürmeyebilir (Sezgin, 2002).

Eşikleme işleminin başarımı, daha sonraki imge işlemlerinin başarımını etkilemesinden dolayı eşik değerinin otomatik bir şekilde ve mümkün olduğunca imgedeki bilgiyi açığa çıkaracak şekilde belirlenmesi büyük bir önem teşkil etmektedir.

Gri seviye bir imgeyi, nesne ve arka plan olarak iki farklı gri seviye grubuna ayırma işlemi, ikili eşikleme olarak tanımlanır (Sezgin, 2002). Şekil 3.1'de örnek bir imge (http://www.cs.cmu.edu/~cil/v-images.html adresinden alınmıştır) ve ikili eşikleme sonrası çıktı imgesi görülmektedir. İkili eşikleme sonrası, maksimum 256 gri seviyeye sahip bir I imgesindeki (x, y) koordinatına sahip piksellerin $I(x, y)$ gri seviye değerlerinden, belirlenen t eşik değerine göre küçük ve eşit olanları ($I(x, y) \leq t$) siyah, büyük olanları ($I(x, y) > t$) beyaz ile ifade edilerek iki gri seviyeli bir imge haline dönüştürülür.



(a) (b)

Şekil 3.1. (a) Örnek imge ve (b) ikili eşiklenmiş imge

Farklı temellere dayanan ve bazıları uygulamaya özgü geliştirilmiş imge eşikleme metotlarını, eşikleme sırasında kullandıkları bilgilere göre histogram şekline dayalı yöntemler, gri seviye topaklamaya dayalı yöntemler, histogram entropisine dayalı yöntemler, nesne özelliklerine dayalı yöntemler, uzamsal bilgileri kullanan yöntemler ve yerel uyarlamalı yöntemler olarak 6 alt başlığa ayırmak mümkündür (Sezgin ve Sankur, 2004). Bu yöntemler ayrıca parametrik ve parametrik olmayan (kritere dayalı) eşikleme yöntemleri olarak ikiye ayrılmaktadır (Sezgin, 2002). Parametrik eşikleme yöntemlerinde hesaplama süresi göreceli olarak yüksektir ve kabul edilen model parametreleri ile imge histogramı arasındaki farkın artması ile başarımı azalır (Sezgin, 2002). Parametrik olmayan yöntemler, sınıf içi değişinti (Otsu, 1979), entropi (Pun, 1980; Kapur ve ark., 1985), bulanık benzerlik ölçütü (Li ve ark., 1995), Bayes hatası (Kittler ve Illingworth,1986) vb. bir kriteri optimal hale getiren eşik değerlerini belirlemeyi temel alır. Parametrik olmayan eşikleme yöntemlerinin daha doğru sonuçlar verdiği teknik yazında belirtilmektedir (Yen ve ark., 1995). Bu yöntemler arasında özellikle Otsu'nun yöntemi (Otsu, 1979), en iyi ve çoğunlukla kullanılan genel bir eşikleme tekniğidir (Don ve ark., 2008).

3.1. Otsu'nun Yöntemi

Otsu'nun ikili eşikleme yöntemi, imge eşikleme alanında yapılan ilk çalışmalardan biri olan ve Otsu (1979) tarafından önerilen topaklamaya dayalı eşikleme yöntemidir. Güçlü ve stabil bir yöntem olması sebebiyle imge eşikleme alanında gerçekleştirilen çalışmalarda referans olarak alınmakta ve kullanılmaktadır (Eichmann ve Lüssi, 2005).

Yöntem ile ilgili notasyonlar ve temel tanımlar aşağıda belirtilmektedir:

$I(x, y)$	: Gri seviyeli bir imge
(x, y)	: İmgenin uzamsal koordinatları
$L = \{0, 1, 2, \dots, L-1\}$	: Gri seviyeleri temsil eden pozitif tamsayı kümesi
MxN	: İmgenin boyutları

Boyutları MxN olarak tanımlanan bir imgenin gri seviye frekansları (gri seviyelerinin tekrarlanma sıklığı) $f_0, f_1, \dots, f_{L-1}$ olarak tanımlanırsa i. gri seviyenin olasılık dağılımı olarak ifade edilen p_i, Denklem 3.1'deki gibi hesaplanır:

$$p_i = \frac{f_i}{MxN}\,, \quad \sum_{i=0}^{L-1} f_i = MxN\,, \quad i = 0, 1, \dots, L-1 \tag{3.1}$$

İmge, seçilen bir t eşik değerine göre imge, nesne ($C_O = \{0, 1, \dots\ t\}$) ve arka plan ($C_B = \{t+1, \dots\ L-1\}$) olarak iki sınıfa ayrılmaktadır. Bu iki sınıfın olasılık yoğunlukları ise Denklem 3.2've 3.3'e göre hesaplanır.

$$\omega_O(t) = \sum_{i=0}^{t} p_i \tag{3.2}$$
$$\omega_B(t) = \sum_{i=t+1}^{L-1} p_i = 1 - \omega_O(t) \tag{3.3}$$

Bu iki sınıfın yoğunluklu ortalama gri seviye değerleri ise aşağıdaki eşitliklerle ifade edilmektedir:

$$u_O(t) = \sum_{i=0}^{t} i.\, p_i\, / \omega_O(t) \tag{3.4}$$
$$u_B(t) = \sum_{i=t+1}^{L-1} i.\, p_i\, / \omega_B(t) \tag{3.5}$$

En eski eşikleme yöntemlerinden biri olan bu yöntemde en uygun eşik değerinin belirlenmesi nesne (O) ve arka plan (B) piksellerine ilişkin ağırlıklandırılmış toplam sınıf içi değişintilerin minimize edilmesi ile gerçekleştirilmektedir (Otsu, 1979). Değişinti (varyans), ölçüm değerleri ile aritmetik ortalamanın farklarının karelerinin ortalaması olarak ifade edilmektedir (Ünver ve ark., 2011). Nesne ve arka plana ait sınıfların değişintileri sırasıyla şu şekilde hesaplanmaktadır (Liu ve Yu, 2009):

$$\sigma^2{}_O(t) = \sum_{i=0}^{t} p_i.\, (i - u_O(t))^2 / \omega_O \tag{3.6}$$
$$\sigma^2{}_B(t) = \sum_{i=t+1}^{L-1} p_i.\, (i - u_B(t))^2 / \omega_B \tag{3.7}$$

Buna göre minimize edilmesi gereken ağırlıklandırılmış toplam sınıf içi değişinti ($\sigma^2{}_{sınıf_ici}$) Denklem 3.8've 3.9'a göre tespit edilmektedir (Liu ve Yu, 2009).

$$\sigma^2{}_{sınıfici}(t) = \omega_O(t).\, \sigma^2{}_O(t) + \omega_B(t).\, \sigma^2{}_B(t) \tag{3.8}$$
$$t = Arg\ \mathrm{Min}_{\mathrm{t}\epsilon\mathrm{G}}\, \sigma^2{}_{sınıfici}(t) \tag{3.9}$$

Sınıf içi değişintilerinin minimize edilmesi ayni zamanda sınıflar arası değişintinin ($\sigma^2{}_{sınıfara}(t)$) en büyüklenmesine karşılık geldiğinden bu yönteme ilişkin

en uygun eşik değeri aşağıdaki ifadeler yardımıyla da hesaplanabilmektedir (Liu ve Yu, 2009):

$$\sigma^2{}_{sınıfara}(t) = \omega_O(t).[u_O(t) - u_T]^2 + \omega_B(t).[u_B(t) - u_T]^2 \quad (3.10)$$

$$u_T = \sum_{i=0}^{L-1} p_i.i \quad (3.11)$$

$$t = Arg\ \mathrm{Max}_{\mathrm{t}\epsilon\mathrm{G}}\ \sigma^2{}_{sınıfara}(t) \quad (3.12)$$

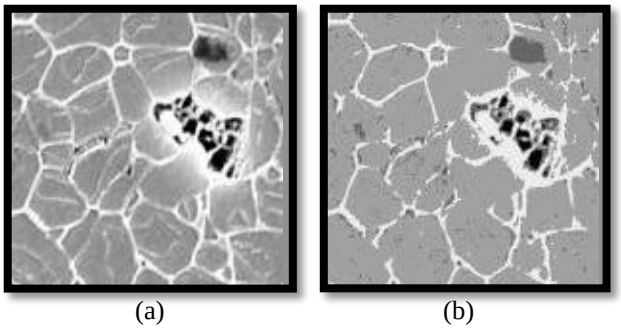
Sınıf içi ve sınıflar arası değişintinin toplamı, toplam değişintiye ($\sigma^2{}_T$) eşittir ve Denklem 3.13'deki gibi ifade edilir (Liu ve Yu, 2009):

$$\sigma^2{}_T = \sigma^2{}_{sınıfici}(t) + \sigma^2{}_{sınıfara}(t) \text{ ve } \sigma^2{}_T = \sum_{i=0}^{L-1} p_i.(i - u_T)^2 \quad (3.13)$$

Bölüm 3.1'de belirtilen eşitlikler daha detaylı ve kapsamlı olarak Eichmann ve Lüssi'nin (2005) çalışmasında görülmektedir.

3.2. Çoklu İmge Eşikleme

Çoklu eşikleme, gri seviyeli bir imgeyi ikiden fazla gri seviye grubuna ayırma işlemi olarak tanımlanabilir (Sezgin, 2002). Bu sayede imge, birtakım gri seviye grubuyla ifade edilebilir hal almaktadır. Çoklu imge eşikleme; bölütleme, sınıflandırma ve nesne ayırt etmeyi içeren çoğu imge işleme uygulamasında kullanılan önemli bir tekniktir (Rueda, 2008). Şekil 3.2'de örnek bir imge (http://www.cs.cmu.edu/~cil/v-images.html adresinden alınmıştır) ve çoklu eşikleme sonrası çıktı imgesi görülmektedir.

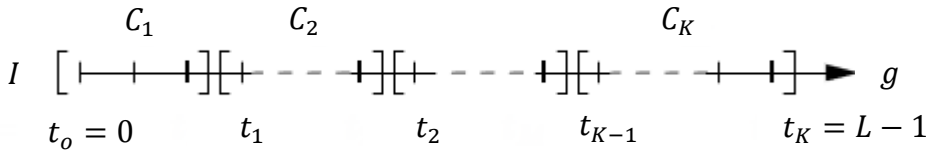


(a) (b)

Şekil 3.2. (a) Örnek imge ve (b) 3 adet eşik değeri ile çoklu eşiklenmiş imge

Literatürde imge eşikleme gerçekleştirmek amacıyla birçok amaç fonksiyonu tanımlanmıştır (Sezgin ve Sankur, 2004; Chu ve ark., 2004; Gonzales-Baron ve Butler, 2006). Bu yöntemler çoklu imge eşikleme amaçlı kullanılmak üzere genişletilebilir niteliktedirler ve bu yöntemler genelde imgenin histogramından çıkarım yapılarak tespit edilmiştir (Hammouche ve ark., 2010). Sadece histogramı temel alarak analiz eden ve değerlendiren eşikleme yöntemleri, her zaman basit, verimli ve gerçek zamanlı sistemlerde kullanılmak üzere uygundur (Eichmann ve Lüssi, 2005). Bu yöntemlerden ikili eşikleme amaçlı geliştirilen Otsu'nun (1979) değişinti fonksiyonu, en popüler yöntemlerdendir.

Çoklu imge eşikleme, MxN adet L={0,1, …, L-1} gri seviyeli piksele sahip I imgesini, K-1 adet ($t_1 < t_2 < \cdots < t_{K-1}$) eşik değerine sahip T=($t_1, t_2, \ldots, t_{K-1}$) eşik değerleri kümesini kullanarak K adet sınıfa ($C_1, C_2, \ldots, C_K$) topaklama işlemidir. Burada sınırları belirten $t_0 = g_{min}$ ve $t_K = g_{max}$ değerleri sırasıyla imgedeki minimum gri seviyeli pikselin değeri ve maksimum gri seviye değeri olarak tanımlanmaktadır ($g_{min} = 0$, $g_{max} = \mathrm{L} - 1$).



Şekil 3.3. İmgedeki gri seviyelerin sınıflara ayrımı (Eichmann ve Lüssi, 2005)

Otsu'nun yöntemi, çoklu imge eşikleme problemini çözmek amacıyla uyarlanabilir. Bu durumda sınıfların olasılık yoğunlukları Denklem 3.2 ve 3.3'ten uyarlanarak şu şekilde ifade edilir (Eichmann ve Lüssi, 2005):

$$\omega_k = \sum_{i \in C_k} p_i \tag{3.14}$$

$$\omega_k(t_{k-1}, t_k) = \sum_{i=(t_{k-1}+1)}^{t_k} p_i \tag{3.15}$$

Sınıfların yoğunluklu ortalama gri seviye değerleri, Denklem 3.4 ve 3.5'ten uyarlanarak Denklem 3.16 ve 3.17'deki gibi ifade edilir (Eichmann ve Lüssi, 2005).

$$u_k = \sum_{i \in C_k} i.p_i/\omega_k \tag{3.16}$$

$$u_k(t_{k-1}, t_k) = \sum_{i=(t_{k-1}+1)}^{t_k} i.p_i\,/\omega_k(t_{k-1}, t_k) \tag{3.17}$$

Sınıflara ait değişintiler Denklem 3.6 ve 3.7'den uyarlanarak Denklem 3.18 ve 3.19'a göre hesaplanır (Eichmann ve Lüssi, 2005).

$$\sigma^2{}_k = \sum_{i \in C_k} p_i . (i - u_k)^2 / \omega_k \tag{3.18}$$

$$\sigma^2{}_k(t_{k-1}, t_k) = \sum_{i=(t_{k-1}+1)}^{t_k} p_i . (i - u_k(t_{k-1}, t_k))^2 / \omega_k(t_{k-1}, t_k) \tag{3.19}$$

Sınıf içi ve sınıflar arası değişintiler sırasıyla Denklem 3.8 ve 3.10 göz önüne alınarak bulunur (Eichmann ve Lüssi, 2005).

$$\sigma^2{}_{sınıfici}(T) = \sum_{k=1}^{K} \omega_k . \sigma^2{}_k \tag{3.20}$$

$$\sigma^2{}_{sınıfara}(T) = \sum_{k=1}^{K} \omega_k . (u_k - u_T)^2 \tag{3.21}$$

Çoklu imge eşikleme problemini çözmek amacıyla en uygun eşik değerlerini bulmak için ya sınıf içi değişintiyi minimize eden ya da sınıflar arası değişintiyi maksimize eden çözüm bulunmalıdır.

$$[t_1^*, t_2^*, \ldots, t_{K-1}^*] = Arg\ Min\ \sigma^2{}_{sınıfici}(t) = Arg\ Max\ \sigma^2{}_{sınıfara}(t) \tag{3.22}$$

3.3. Gri Ton Düzgünlüğü ve Düzensizliği Ölçütü

Eşikleme yöntemlerinin çoğu sadece imge histogramı üzerindeki bilgileri kullandığından eşikleme sonucunda elde edilen bölgelerin tekdüzeliğine ilişkin değerlendirme Levine ve Nazif (1985) tarafından önerilen ve Tsai (1995) tarafından geliştirilen Gri Ton Düzgünlüğü (GTD) (Uniformity) ölçütü kullanılarak gerçekleştirilebilir (Tsai, 1995). Bu ölçütün tersi olan Gri Ton Düzensizliği'de (GTDS) gerçekleştirilen eşiklemenin başarımını değerlendirmek için kullanılabilir ve bu ölçütler aşağıdaki gibi ifade edilmektedir (Sezgin, 2002):

$$GTD = 1 - \frac{\sum_{k=1}^{K} \omega_k . \sigma^2{}_k}{\sigma^2{}_T} \tag{3.23}$$

$$GTDS = \frac{\sum_{k=1}^{K} \omega_k . \sigma^2{}_k}{\sigma^2{}_T} \tag{3.24}$$

Denklem 3.20'de ifade edilen $\sigma^2{}_{sınıfici}$ Denklem 3.23 ve 3.24'de yerine koyulursa Denklem 3.25 ve 3.26'daki ifadeler elde edilir (Liang ve Yin, 2011).

$$GTD = 1 - \frac{\sigma^2{}_{sınıfici}}{\sigma^2{}_T} = \frac{\sigma^2{}_{sınıfara}}{\sigma^2{}_T} \tag{3.25}$$

$$GTDS = \frac{\sigma^2{}_{sınıfici}}{\sigma^2{}_T} \tag{3.26}$$

Birbirine yakın grilik değerine sahip bölgeleri aynı sınıf olarak bulma açısından iyi bölütlenmiş bir imgenin GTD ölçütü 1 değerine, GTDS ölçütü ise 0 değerine daha yakın olacaktır (Sezgin, 2002).

Otsu'nun yöntemi kullanılarak gerçekleştirilen çoklu imge eşiklemede amaç fonksiyonu GTD veya GTDS ölçütü olarak normalize edilip ölçüt sırasıyla maksimize veya minimize edilerek en uygun eşik değeri kümesine ulaşılabilir (Liang ve Yin, 2011).

3.4. Etraflı (Tam) Arama

Çoklu imge eşikleme, bir imgeyi K adet sınıfa ayırmak amacıyla amaç fonksiyonunu maksimize veya minimize eden K-1 adet eşik değerini belirleme işlemidir. Bu problem için üretilecek en basit ve belirli çözüm uygun amaç fonksiyonu değerini elde etmek amacıyla tüm olası eşik değerleri yerleşimlerini hesaplamak ve en uygun amaç fonksiyonunu veren eşik değerleri kümesini seçmektir. Bu yaklaşım etraflı arama olarak adlandırılmaktadır ve kombinasyon problemi olarak ele aldığı problem için tüm olasılıkları dener (Eichmann ve Lüssi, 2005).

Örnek olarak L=6 adet gri seviyeye sahip bir imge K=4 ayrı sınıfa ayrılması istenirse K-1=3 adet eşik değeri belirlemek gerekir. Son gri seviye değerinin eşikleme sonrası son sınıf olan C_4 sınıfı içerisinde bulunması gerektiği de değerlendirilmelidir. Etraflı aramanın uygulanması sonucunda amaç fonksiyonunun kaç kez hesaplandığı, Denklem 3.27 ile belirlenir (Eichmann ve Lüssi, 2005).

$$\binom{n}{k} = \frac{n!}{k!(n-k)!} \tag{3.27}$$

$$\binom{L-1}{K-1} = \binom{5}{3} = \frac{5.4.3.2.1}{3.2.1.2.1} = 10$$

Eşitlik 3.27'de görüldüğü üzere kombinasyon sonrası 10 kez amaç fonksiyonu hesabı gerçekleştirilmektedir. Gerçek zamanlı bir örnek vermek gerekirse L=256 gri seviyeli bir imgeyi K=5 farklı sınıfa ayırmak istenildiği taktirde en uygun amaç fonksiyonunu veren $\{t_1, t_{2,} t_3, t_4\}$ eşik değerleri kümesini bulmak amacıyla $\binom{255}{4} = 172061505$ kez amaç fonksiyonu hesabı gerekmektedir (Eichmann ve Lüssi, 2005). Bu nedenle en uygun amaç fonksiyonu değerini her bir olası eşik değeri kombinasyonlarını kontrol etmeden keşfeden metasezgisel algoritmaları kullanmak hesaplama karmaşıklığı ve işlem zamanı açısından büyük bir avantaj sağlayabilir.

4. METASEZGİSEL ALGORİTMALAR

Teknik yazında ilk defa Glover (1986) tarafından ifade edilen metasezgisel terimi, "daha ilerisi, üst seviye" anlamına gelen Yunanca "meta" kelimesi ile sezgisel anlamına gelen İngilizce "heuristic" kelimesinin birleşiminden meydana gelmektedir.

Metasezgisel yaklaşım çözüm uzayında stokastik ancak bilinçli bir şekilde arama gerçekleştiren yöntemlerden oluşur (Blum ve Roli, 2003). Bu yöntemler her iterasyonda çözüm kümesinden yola çıkarak yeni çözümler üretirler. Böylece, arama uzayının en uyguna yakın olan noktalarında arama gerçekleştirerek ve yerel en iyi noktalardan kurtularak, en uygun çözüme yakın sonuçlar elde etmeyi sağlarlar.

Blum ve Roli (2003)'ye göre, metasezgisel yöntemler aşağıdaki şekilde karakterize edilmektedir:

• Metasezgisel yöntemler arama işlemine yön veren stratejilerdir.

• Genel amaç, arama uzayını etkili bir şekilde keşfederek en iyi veya en iyiye yakın sonuçlar elde etmektir.

•Metasezgisel yöntemler yerel arama tekniklerinden, karmaşık öğrenme işlemlerine kadar yayılım gösterir.

•Metasezgisel algoritmalar yaklaşık ve genelde belirleyici olmayan yöntemlerdir.

•Arama uzayında yerel en iyi konumlara takılıp kalmayı engelleyecek mekanizmaları içerirler.

•Metasezgisel teknikler belirli bir probleme özgü değildir.

Metasezgisel, alt sezgiselin işleyişini, verimli bir şekilde yüksek kaliteli çözümler üretmesi amacıyla değiştiren ve yönlendiren tekrarlamalı bir üst süreçtir. Bu alt sezgisel, düşük veya yüksek seviyeli bir işlem, basit bir yerel arama, yapıcı bir yöntem olabilir (Blum ve Roli, 2003). Bir metasezgisel algoritmanın iyi sonuçlar üretmesi için yapılması gereken ilk işlem, yöntemin temel kavramlarının probleme iyi bir şekilde adapte edilmesidir (Hertz ve Widmer, 2003).

Metasezgisel algoritmalar aşağıdaki gibi sınıflandırılmaktadır (Tarantilis ve ark., 2005):

•Doğadan esinlenerek geliştirilenler – Doğadan esinlenmeden geliştirilenler

•Popülasyon tabanlı algoritmalar – Tek nokta (yerel arama) algoritmalar
•Dinamik amaç fonksiyonlu – Statik amaç fonksiyonlu algoritmalar
•Tek komşu yapılı – Çok komşu yapılı algoritmalar
•Hafıza kullanan – Hafıza kullanmayan algoritmalar

Metasezgisel algoritmalar, klasik sezgisel algoritmaların doğadan esinlenerek geliştirilmiş hali olarak görülebilir (Laporte ve ark., 2000). Bu yöntemler sosyal, biyoloji, zooloji, fizik, bilgisayar vb. bilimler temel alınarak geliştirilmiştir. Bu yöntemlerden PSO (Kennedy ve Eberhart, 1995), GOA (Rajabioun, 2011) ve AA (Yang, 2008) sırasıyla kuşların besin arayışı, guguk kuşlarının yumurtlama ve göç davranışı, ateş böceklerinin birbirleriyle olan etkileşimleri gibi doğanın temelinde var olan davranışlar temel alınarak geliştirilmiştir.

4.1. Guguk Kuşu Optimizasyonu Algoritması

Guguk kuşları, Gugukgiller ailesine mensup hafif eğri gagalı, uzun ve sivri kanatlı, uzun kuyruklu, iyi uçucu kuşlardır. Eski Dünya ve Yeni Dünya ormanlarında yaşamaktadırlar ve serçe iriliğinden karga büyüklüğüne kadar 130 çeşit türü olduğu bilinmektedir (Anonim, 2012). Bu kuşları diğer kuş türlerinden ayıran en etkileyici özellikleri atılgan çoğalma stratejileridir.

Eski Dünya türlerinden 56'sı ve Yeni Dünya türlerinden 3'ü kuluçka asalağı olup yumurtalarını yabancı kuşların yuvalarına bırakır (Payne ve ark., 2005). Dişi guguk kuşu, yumurtasını yerleştirmek istediği yuvanın sahibi olan yabancı kuşun yumurtalarından birini alarak kendi yumurtasını yuvaya yumurtlamaktadır ve bu süreç sadece 10 saniyesini almaktadır (Rajabioun, 2011). Örneğin "Ani" ve "Guira" guguk kuşu türleri yavrularının yumurtadan çıkma olasılığını artırmak amacıyla bu şekilde bir strateji uygulamaktadır (Payne ve ark., 2005). Dişi guguk kuşları belirlediği bir kuş türü üzerinde yoğunlaşır ve yumurtalarını şekil ve renk açısından, yuva sahibi olan bu kuş türüne ait yumurtalara benzetir (Rajabioun, 2011). Örneğin "Tapera" türü guguk kuşları birçok yabancı kuş türünün yumurtalarını şekil ve renk açısından taklit ederek yumurtalarının yabancı kuş türüne ait yuvada fark edilme olasılığını azaltmaktadır (Payne ve ark., 2005).

Çoğu yabancı kuş türü yuvasına bırakılan guguk kuşu yumurtalarını fark edebilme yetisine sahiptir. Böyle bir durumda yuvasına konulmuş guguk kuşu

yumurtasını yuvadan atmakta veya yuvayı yeniden yapmak üzere dağıtmaktadır. Üvey anne tarafından fark edilmeyen yumurtalar, kuluçka dönemi sırasında üvey kardeşlerinden önce yumurtadan çıkarlar. İlk 4 gün içerisinde gözleri henüz açılmamış olmasına rağmen akrobatik hareketlerle üvey kardeşlerini yuvadan dışarı atmaya başlar (Anonim, 2012). Eğer yuvadaki guguk kuşu yumurtasından daha önce yumurtadan çıkan üvey annenin yavruları mevcutsa, kendilerinden sonra yumurtadan çıkan guguk kuşu yavrusunun yeme içgüdüsünden dolayı hayatta kalamayacaklardır çünkü daha büyük bir bünyeye sahip olan yavru guguk kuşu, üvey annenin getirdiği yiyeceklerin çoğunu tüketmektedir (Rajabioun, 2011).

4.1.1. Popülasyonu başlatma

Bir optimizasyon problemini çözmek amacıyla problem değişkenlerini bir dizide tutmak gerekir ve bu dizi GA ve PSO terminolojilerinde sırasıyla "Kromozom" ve "Parçacık Konumu" olarak adlandırılmaktadır. Guguk kuşu Optimizasyon Algoritması'nda (GOA) bu dizi, kuşların yetiştiği doğal yaşam ortamı olarak ifade edilen "ortam" kelimesiyle tanımlanmaktadır (Rajabioun, 2011). N_b boyutlu bir problem için ortam ve f_O ile ifade edilen amaç fonksiyonunun maliyeti Denklem 4.1 ve 4.2'ye göre ifade edilmektedir (Rajabioun, 2011).

$$ortam = [x_1, x_2, \dots, x_{N_b}] \tag{4.1}$$

$$maliyet = -f_O\ (ortam) = -f_O\ (x_1, x_2, \dots, x_{N_b}) \tag{4.2}$$

GOA, maliyeti maksimize eden bir optimizasyon yöntemidir ve minimizasyon problemlerine uygulanmak istendiği taktirde maliyeti Denklem 4.3'deki gibi tanımlanabilir:

$$maliyet = f_O\ (ortam) = f_O\ (x_1, x_2, \dots, x_{N_b}) \tag{4.3}$$

Optimizasyon algoritmasını başlatmak amacıyla N_b problem boyutuna ve N_{pop} adet popülasyon sayısına (ortam sayısı) bağlı olarak N_{pop}xN_b matrisi oluşturulur. Ardından her bir ortam için ortamda gerçekleşecek olan yumurtlama işlemi sırasında oluşacak yumurtaların sayıları, y_{min} alt sınır ve y_{max} üst sınırı arasında rastgele bir

değer olarak belirlenir. Her bir guguk kuşunun 5 ile 20 arasında yumurta yumurtladığı kabul edilmektedir. (Rajabioun, 2011).

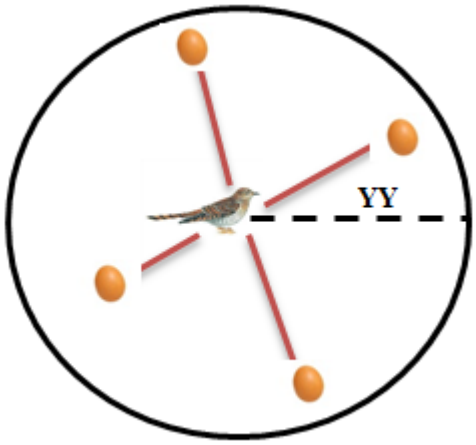
Gerçek guguk kuşlarının diğer bir davranışı da doğal ortamlarından belirli bir maksimum uzaklık içerisinde yumurtalarını bırakmasıdır. Bu maksimum uzaklık değeri, algoritmada Yumurtlama Yarıçapı (YY) olarak tanımlanmaktadır. Algoritmada her bir guguk kuşunun YY değeri, kendi yumurta sayısı, o anki toplam yumurta sayısına ve yumurta sayısı limitlerine bağlı olarak Denklem 4.4'deki gibi belirlenmektedir (Rajabioun, 2011).

$$YY = \alpha \, x \, \frac{Guguk\ kuşunun\ yumurta\ sayısı}{O\ anki\ toplam\ yumurta\ sayısı} \, x \, (y_{max} - y_{min}) \tag{4.4}$$

α parametresi, YY'nin maksimum değerini ayarlamak amacıyla kullanılan tamsayı bir değerdir (Rajabioun, 2011).

4.1.2. Yumurtlama süreci

Şekil 4.1'de görüldüğü gibi bir guguk kuşu, merkezinde bulunduğu, YY değerinde yarıçap genişliğinde alana sahip ortam içerisine rastgele yumurta yaymaya başlar.



Şekil 4.1. YY ile kısıtlanmış ortam ve rastgele yumurta yayma işlemi

Tüm guguk kuşu yumurtalarının, yabancı kuşların yuvalarına bırakılması işlemi sonrasında yabancı kuşun yumurtalarına daha az benzeyen bazı guguk kuşu yumurtaları yabancı kuş tarafından tespit edilir ve yuvadan atılır. Burada o anki mevcut yumurtalar

arasından daha düşük maliyete sahip %p kadar (genelde %10) yumurta yok edilmiş olacaktır (Rajabioun, 2011).

Algoritmada mevcut olan N_{max} parametresi ortamda yaşamını sürdüren guguk kuşu sayısını kısıtlar ve kontrol eder. Bu sebeple her bir iterasyonda en iyi maliyet değerlerine sahip N_{max} sayısı kadar guguk kuşu hayatta kalacaktır (Rajabioun, 2011).

4.1.3. Guguk kuşlarının göç süreci

Guguk kuşu yavruları büyüdüğünde ve yumurtlama yaklaşımı zamanı geldiğinde mevcut bulunduğu ortamdan, yumurtalarını yabancı kuşların yumurtalarına daha çok benzetebileceği türlere sahip ve yavruları için daha çok yiyecek kaynağına sahip yeni ve daha iyi bir ortama göç ederler (Rajabioun, 2011).

Guguk kuşlarının farklı alanlarda grup oluşturmasının ardından en iyi maliyete sahip olan topluluk diğer guguk kuşlarının göç etmesi için hedef nokta olarak seçilmektedir. Guguk kuşu gruplarını belirlemek amacıyla K-Ortalama Kümeleme Algoritması (KKA) kullanılmaktadır (Rajabioun, 2011).

KKA, MacQueen (1967) tarafından geliştirilen en eski kümeleme algoritmalarından biridir. KKA'nın atama mekanizması, her verinin sadece bir kümeye ait olabilmesine izin verir. Merkez noktanın kümeyi temsil etmesi ana fikrine dayalı bir metottur (Han ve Kamber, 2001). Eşit büyüklükte küresel kümeleri bulmaya eğilimlidir. Kümelemede en yoğun olarak kullanılan ölçüt, ilgili kümenin merkezi ile veri setindeki her konum arasındaki karesel uzaklıklarının toplamını bulmak amacıyla kullanılan Kümeleme Hata Kriteri'dir (KHK) (Likas ve ark., 2003) ve Denklem 4.5'de m_i merkezli k adet C_i kümesinin KHK'sı ifade edilmektedir (Pang-Ning ve ark., 2006).

$$KHK = \sum_{i=1}^{k} \sum_{x \epsilon C_i} uzaklık^2(m_i, x) \tag{4.5}$$

Bu kriteri minimize eden en popüler kümeleme yöntemi KKA'dır. KKA'nın n adet guguk konumunu k adet kümeye bölmek amacıyla uygulanacak işlem basamakları şu şekilde tanımlanmaktadır:

1. adım: Kümelerin merkez noktaları tanımlanır. Guguk konumları arasından k adet rastgele nokta seçilerek veya tüm guguk konumlarının ortalaması temel alınarak belirlenir.

2. adım: Guguk konumlarının merkez noktalara olan uzaklığı hesaplanır. Elde edilen sonuçlara göre k adet kümeden kendilerine en yakın olan kümeye yerleştirilir.

3. adım: Oluşan kümelerin yeni merkez noktaları o kümedeki tüm guguk konumlarının ortalama değeri ile değiştirilir.

4. adım: Merkez noktalar değişmeyene dek 2. ve 3. adımlar tekrarlanır.

Guguk kuşlarını kümelemede, Rajabioun (2011) tarafından gerçekleştirilen benzetimler sonucunda k değerinin 3-5 arasında seçilmesi uygun görülmektedir. Kümelerin oluşturulmasının ardından Denklem 4.6'da görüldüğü üzere her bir kümenin ortalama maliyet değerleri hesaplanarak en iyi maliyet değerine sahip olan küme Denklem 4.7'ye göre hedef küme olarak seçilir (Rajabioun, 2011).

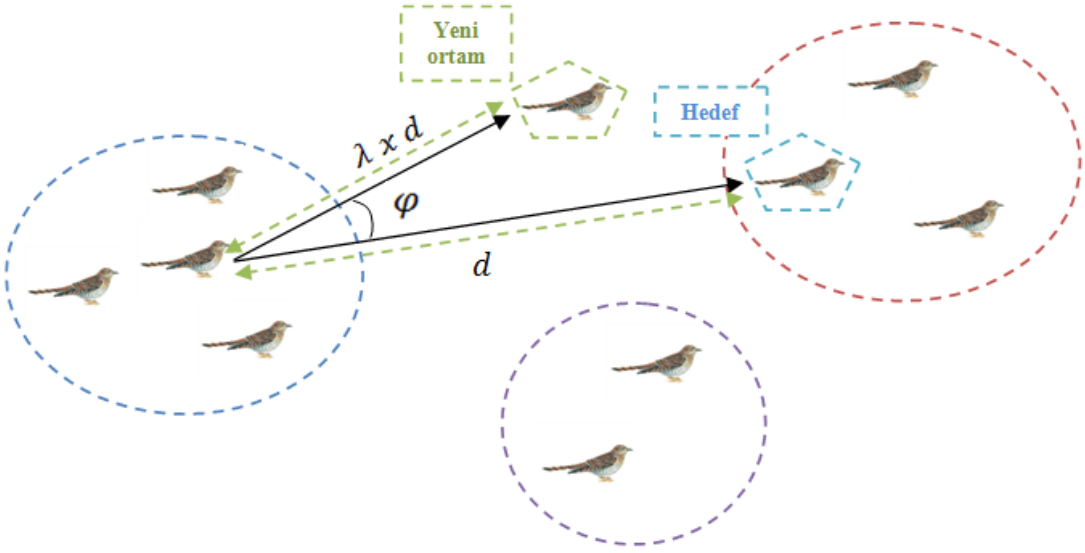
$$ort(C_i) = \frac{\sum_{x \epsilon C_i} f_O(x)}{n_i} \tag{4.6}$$

$$hedef = Arg\,\mathrm{Max}_{i \epsilon \{1,\ldots,k\}}\, ort(C_i) \tag{4.7}$$

$$h_ortam = Arg\,\mathrm{Max}_{x \epsilon C_{hedef}}\, f_O(x) \tag{4.8}$$

Denklem 4.8'de h_ortam olarak tanımlanan guguk kuşu konumu, seçilen hedef kümeye ait guguk kuşları arasında en iyi maliyete sahip ortam olarak değerlendirilir ve göç edilecek olan hedef ortam konumunu alır.

Hedef ortama doğru göç eden guguk kuşları aradaki mesafeyi tamamıyla katetmezler, hedefe bir miktar sapma açısıyla beraber belirli bir mesafe yaklaşırlar.



Şekil 4.2. Bir guguk kuşunun hedef ortama doğru göç örneği

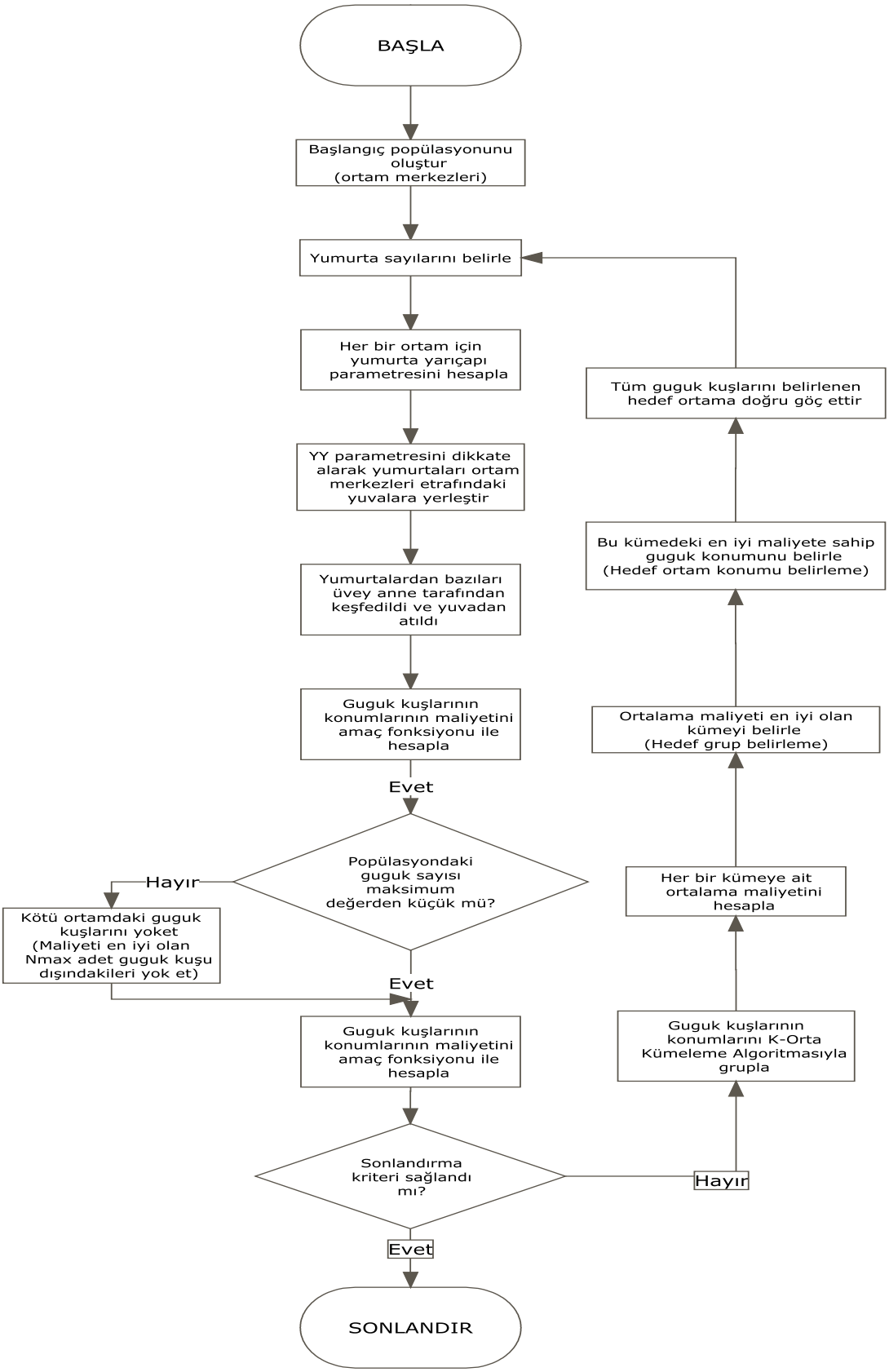
Şekil 4.2'de görüldüğü gibi her bir guguk kuşu çözümü, hedef ortama olan uzaklığının λ katı kadar hedefe yaklaşır ve φ radyan kadar bir sapma eğilimi gösterir. λ ve φ parametreleri Denklem 4.9'a uygun olarak tanımlanmaktadır (Rajabioun, 2011):

$$\lambda \sim U(0,1),\ \varphi \sim U(-w, w) \tag{4.9}$$

Denklem 4.9'da ifade edilen w parametresi φ'nun değerini belirleyen aralığın sınır parametresidir ve en iyi maliyete yakınsamak için $\pi/6$ olarak seçmek yeterlidir (Rajabioun, 2011).

GOA algoritmasının akış şeması Şekil 4.3'de verilmiştir.

4.1.4. Akış şeması



Şekil 4.3. Guguk kuşu algoritmasının akış şeması

4.2. Ateş Böceği Algoritması

Ateş böcekleri, Ateş böceğigiller (Lampyridae) familyasını oluşturan, geceleri uçarken yanıp sönen ışıkları ile bilinen, özellikle sıcak ve ılıman bölgelerde yaşayan, yaklaşık olarak 1900 farklı türe sahip böceklerdir (Lukasik ve Zak, 2009). Bu böcekler, saydam bir kütikül tabakasının arkasında ve vücut yüzeyine yakın bulunan karın bölgesindeki fotojenik hücreler sayesinde ışık üretme yeteneğine sahiptirler (Babu ve Kannan, 2002). Ateş böceklerinin oluşturduğu ışık sinyalleri karşı cinse kur yapma, avını cezp etme, sosyal ortama uyum sağlama ve avcıya karşı bir uyarı işareti olarak işlev gördüğü bilinmektedir (Fraga, 2008).

AA, Xin-She Yang tarafından ateş böceklerinin ışıldama davranışı temel alarak geliştirilmiştir (Yang, 2008). Algoritmada temel alınan ışıldama karakteristiği şu üç kuralla birlikte özetlenmektedir (Yang, 2009):

- Ateş böcekleri için cinsiyet farkı gözetilmemektedir. Bu nedenle bir ateş böceği diğer ateş böceklerini, cinsiyete bakılmaksızın etkileyebilecektir.
- Etkileyicilik, ateş böceğinin parlaklık derecesine bağlıdır. Daha az parlaklığa sahip ateş böceği, daha çok parlaklığa sahip olan ateş böceğine doğru hareket edecektir. Eğer kendisinden daha parlak bir ateş böceği yoksa rastgele bir şekilde hareket edecektir.
- Bir ateş böceğinin parlaklığı, probleme ait amaç fonksiyonuna ve problemin türüne göre belirlenir.

4.2.1. Popülasyonu başlatma

Başlangıçta tüm ateş böcekleri S arama uzayında d boyutlu problem için rastgele bir şekilde veya probleme uygun olarak tasarlanmış bir yerleştirme stratejisiyle oluşturulur. Oluşturulan m adet ateş böceği için her bir i ateş böceğine ait x_i çözümünün uygunluk değerini veren optimizasyon problemi fonksiyonu belirlenir.

$$f(x^*) = min_{x \epsilon S} f(x) \tag{4.10}$$

Problemin minimizasyon problemi olarak temel alındığı düşünülürse hedef, $f(x)$ amaç fonksiyonunu minimize eden x^* çözümünü bulmaktır. Daha önce belirtildiği üzere her bir ateş böceği çözümünün amaç fonksiyonuna uygulanması sonucunda elde edilen maliyet değeri o ateş böceğinin parlaklığını ifade eder (Yang, 2008).

4.2.2. Etkileşimi belirleme

Her bir ateş böceği, diğer ateş böceklerini ne derece etkilediğini gösteren kendine özgü ve farklı bir β etkililiğine sahiptir (Lukasik ve Zak, 2009). Ateş böceği i'nin ateş böceği j den etkilenip etkilenmediğini belirlemek için öncelikle parlaklığının j ye göre daha düşük olduğunun belirlenmesi gerekmektedir ve çözümlerin maliyet değerleri kıyaslanır. Parlaklığının daha düşük olduğunun belirlenmesi durumu, x_i çözümünün x_j çözümünden etkilenen konumda olduğunu gösterir ve β sini belirlemek amacıyla i'nin j'ye olan $r_{i,j} = d(x_i, x_j)$ uzaklığının etkisiyle monoton bir şekilde azalan fonksiyon (üstel fonksiyon) seçilmelidir (Horng ve Jiang, 2010).

$$r_{i,j} = \|x_i - x_j\| = \sqrt{\sum_{k=1}^{d}(x_{i,k} - x_{j,k})^2} \tag{4.11}$$

$$\beta = \beta_0 . e^{-\gamma . r_{i,j}} \tag{4.12}$$

Denklem 4.12'de görülen β_0 parametresi $r_{i,j} = 0$ olması durumundaki etkililik parametresidir ve Denklem 4.11'de görülen γ parametresi, ışık soğurma katsayısıdır (Yang, 2008). Genelde β_0 parametresi [0 1] aralığında bir değer olarak seçilmektedir (Lukasik ve Zak, 2009).

γ=0 olarak seçmek sabit bir etkililik değerine, γ=∞ olarak seçmek sıfıra yakın bir etkililiğe karşılık gelir. γ parametresinin değeri probleme uygun olarak [0 10] aralığında seçilir (Yang, 2008). Optimize edilen arama uzayının karakteristik uzunluğu da parametre içerisine dâhil edilirse γ parametresi Denklem 4.13 ve 4.14'e göre hesaplanır (Lukasik ve Zak, 2009):

$$\gamma = \frac{\gamma_0}{r_{max}} \tag{4.13}$$

$$\gamma = \frac{\gamma_0}{r_{max}^2} \tag{4.14}$$

$$r_{max} = \max d(x_i, x_j) \, , \forall x_i, x_j \in S \tag{4.15}$$

Denklem 4.13 ve 4.14'de görülen γ_0 parametresi, [0 1] aralığında sabit bir katsayıdır.

4.2.3. Konum güncelleme

k boyutlu bir i ateş böceğinin j ateş böceğinden etkilenmesi halinde β parametresinin belirlenmesinin ardından konumu Denklem 4.16' ya göre güncellenir (Horng ve Jiang, 2010):

$$x_{i,k} = (1-\beta).x_{i,k} + \beta.x_{j,k} + u_{i,k} \tag{4.16}$$

Denklem 3.16'da görülen $u_{i,k}$, i ateş böceğinin konumuna rastsallık katar ve [0 1] aralığında rastgele üretilen bir rand1 değeri ve [0 1] aralığında belirlenen α parametresine bağlı olarak ifade edilir:

$$u_{i,k} = \alpha.(rand1 - \tfrac{1}{2}) \tag{4.17}$$

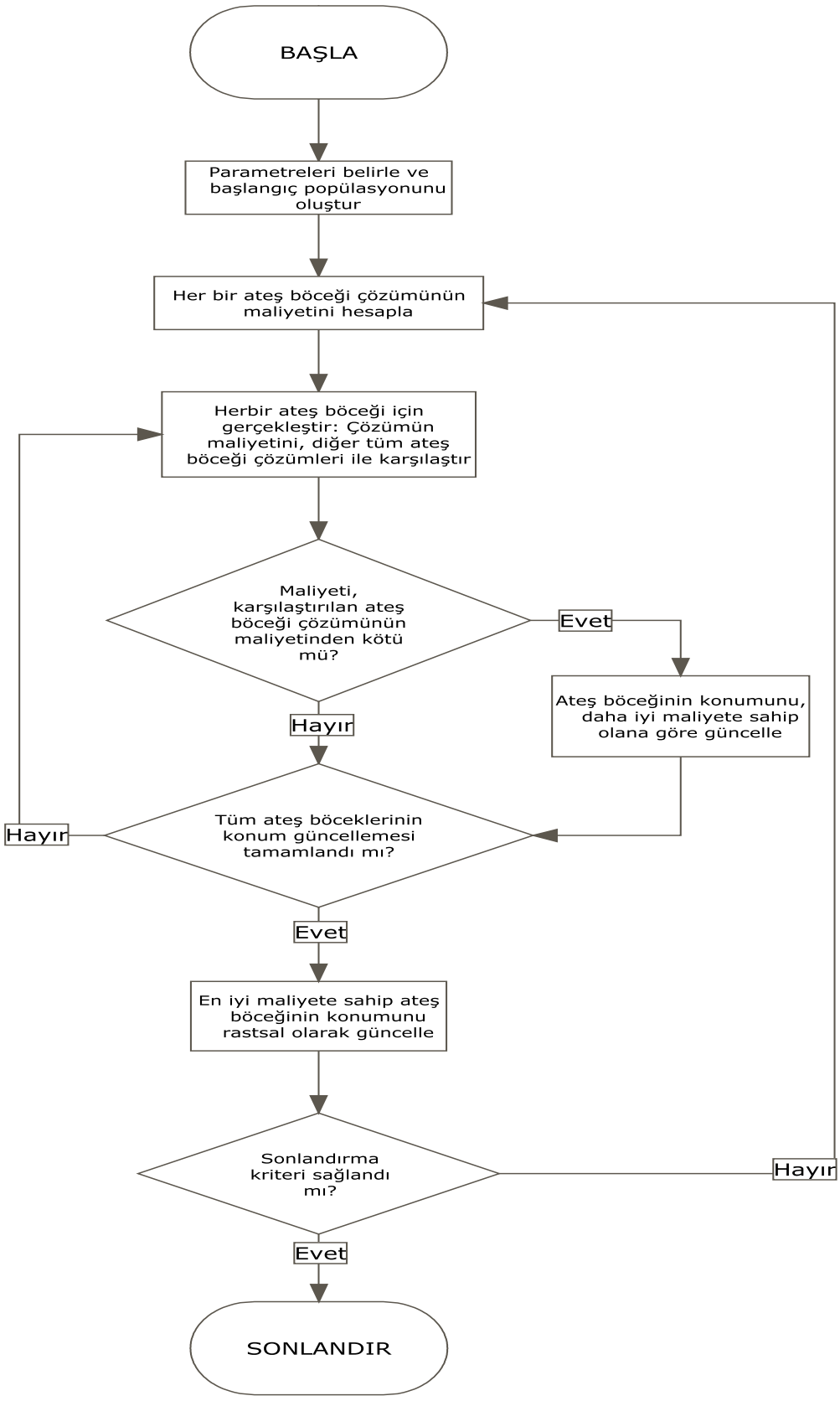
En iyi maliyet değerine sahip olan (en parlak) ateş böceği arama uzayında Denklem 4.18'e göre rastgele hareket edecektir:

$$x_{i^{max},k} = x_{i^{max},k} + u_{i^{max},k} \quad k = 1,2,\dots,d \tag{4.18}$$

$$u_{i^{max},k} = \alpha.(rand1 - \tfrac{1}{2}) \tag{4.19}$$

Ateş böceği algoritmasının akış şeması Şekil 4.4'de verilmiştir.

4.2.4. Akış şeması



Şekil 4.4. Ateş böceği algoritmasının akış şeması

4.3. Parçacık Sürü Optimizasyonu Algoritması

Doğadaki sürülerin etkili biçimde beslenmek amacıyla yiyecek arama davranışlarından esinlenilerek birtakım optimizasyon yöntemleri geliştirilmiştir. Bu optimizasyon yöntemlerinden biri olan Parçacık Sürü Optimizasyonu (PSO), Dr. Eberhart ve Dr. Kennedy (1995) tarafından kuş sürülerinin davranışlarından esinlenerek geliştirilmiş popülasyon tabanlı bir optimizasyon tekniğidir.

PSO senaryo olarak kuşların belirli bir bölge içerisinde yiyecek kaynağını aramalarına benzetilebilir (Kennedy ve Eberhart, 1995). Bölge içerisinde sadece tek bir yiyecek kaynağı bulunmaktadır ve kuşlardan hiçbiri besin kaynağının yerini bilmemektedir ancak kaynaktan ne kadar uzakta olduklarının bilgisi mevcuttur.

4.3.1. Popülasyonu başlatma

PSO tekniğinde, her bir çözüm birer parçacıktır ve uygulanacak problemdeki parametre sayısına göre parçacıklar, belirli bir boyuta sahiptir. Bu parçacıklar mevcut optimum değerleri izleyerek çözüm uzayında gezinirler. Her bir parçacık, uygulanan problemin amaç fonksiyonuna bağlı olarak bir uygunluk değerine ve çözüm uzayında gezinmelerini kontrol eden bir hız vektörüne sahiptir (Kennedy ve Eberhart, 1995).

Bu optimizasyon yönteminin temel basamaklarından birisi başlangıç popülasyonunun oluşturulmasıdır. Genel olarak uygulanan probleme bağlı olan bir aralık içerisinde rastgele konumlara yerleştirilmiş ve rastgele veya sıfır değerli hızlara sahip parçacıklardan oluşur.

d adet boyuttan oluşan *n* adet parçacık için popülasyon matrisi Denklem 4.20'de, popülasyon matrisindeki *i*. parçacık Denklem 4.21'de görülmektedir.

$$X = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1d} \\ x_{21} & x_{22} & \dots & x_{2d} \\ \vdots & & & \\ x_{n1} & x_{n2} & \dots & x_{nd} \end{bmatrix}_{nxd} \tag{4.20}$$

$$x_i = [x_{i1},\ x_{i2}, \dots, x_{id}] \tag{4.21}$$

i' inci parçacığın daha önce bulunmuş olduğu en iyi konum bilgisi p_{best} olarak tanımlanır.

$$p_{best_i} = [p_{i1},\ p_{i2}, \dots, p_{id}] \tag{4.22}$$

En iyi parçacığın konum bilgisi olarak tanımlanan g_{best}, bir iterasyondaki tüm parçacıklar için aynıdır ve Denklem 4.23'teki gibi ifade edilir.

$$g_{best} = [p_1,\ p_2, \dots, p_d] \tag{4.23}$$

PSO yöntemi için kullanılması gereken paremetreler şu şekilde tanımlanmaktadır (Tamer ve Karakuzu, 2006):

Parçacık miktarı (n): Probleme bağlı olarak genellikle 20-40 arasında seçilebilir.

Parçacık boyutu (d): Optimize edilecek problemin boyutuyla aynı değere sahiptir.

Parçacık değer aralığı: Optimize edilecek fonksiyon değişkenlerinin değer aralığını belirtir.

Vmax: Bir parçacığın ulaşabileceği maksimum hız değeridir. Örneğin [-2,2] aralığındaki bir parçacığın Vmax değeri 4' dür.

Öğrenme faktörleri: Denklem 4.25'deki c_1 ve c_2 parametreleridir. c_1, parçacığın p_{best} değerine; c_2, parçacığın g_{best} değerine yaklaşmasını sağlar. Öğrenme faktörleri genel olarak 2 seçilir ancak bu değerler [0 4] aralığında da seçilebilmektedir.

Rastgele sayılar: Denklem 4.25'deki r_1 ve r_2 parametreleri [0 1] aralığında düzgün dağılımlı rastgele sayılardır (Kennedy ve Eberhart, 1995).

Sonlandırma ölçütü: Süre, iterasyon sayısı, belirli bir iterasyon sayısınca g_{best} değerinin değişmemesi, sonlandırma kriterleri olarak seçilebilir.

Eylemsizlik ağırlığı: Denklem 4.27'de geçen eylemsizlik ağırlığı (ω), eski hız vektörünün yeni hız vektörüne etkisini ayarlayan parametredir ve [0 1] aralığında seçilebilir. Ayrıca her bir iterasyonda doğrusal bir şekilde azaltılması en uygun değere yaklaşan parçacıkların değere yaklaştıkça çözüm uzayında küçük adımlarla hareket etmesini sağlar.

4.3.2. Hız ve konum güncelleme

PSO'da mevcut olan her parçacığın ayrı bir hızı vardır ve parçacıklar hızlarını mevcut olan en iyi parçacığa göre yenilerler. Parçacıklar en iyi parçacığa doğru yaklaşma esnasında, kullanılan rastgele parametreler sayesinde bir sonraki iterasyonda daha iyi çözümler üretilebilir.

Her bir parçacığa ait olan konum bilgisi, parçacığın daha önceden bulunmuş olduğu en iyi konum bilgisi olarak tanımlanan p_{best} ve popülasyonda bulunan tüm parçacıklar arasından şimdiye kadar bulunmuş olan en iyi parçacığın konum bilgisi olarak tanımlanan g_{best} değerlerine bakılarak güncellenir.

Parçacıkların hızları v, konumları da x şeklinde tanımlanırsa PSO'nun t. iterasyonu sonunda i. parçacığın yeni hız ve konum bilgileri Denklem 4.25 ve 4.26'ya göre güncellenir. Her bir parçacık için bu işlem tekrarlanır.

$$v_i{}^{t+1} = v_i{}^{t} + c_1.r_1{}^{t}.\left(p_{best_i}{}^{t} - x_i{}^{t}\right) + c_2.r_2{}^{t}.\left(g_{best}{}^{t} - x_i{}^{t}\right) \tag{4.25}$$

$$x_i{}^{t+1} = x_i{}^{t} + v_i{}^{t+1} \tag{4.26}$$

Algoritmanın başarımını arttırmak amacıyla birçok yöntem ortaya atılmıştır. Algoritmayla ilgili yapılan bu çalışmaların ortak amacı, parçacıkların yerel çözümlere takılmadan, optimum çözüme daha hızlı bir şekilde yakınsamasını sağlamaktır.

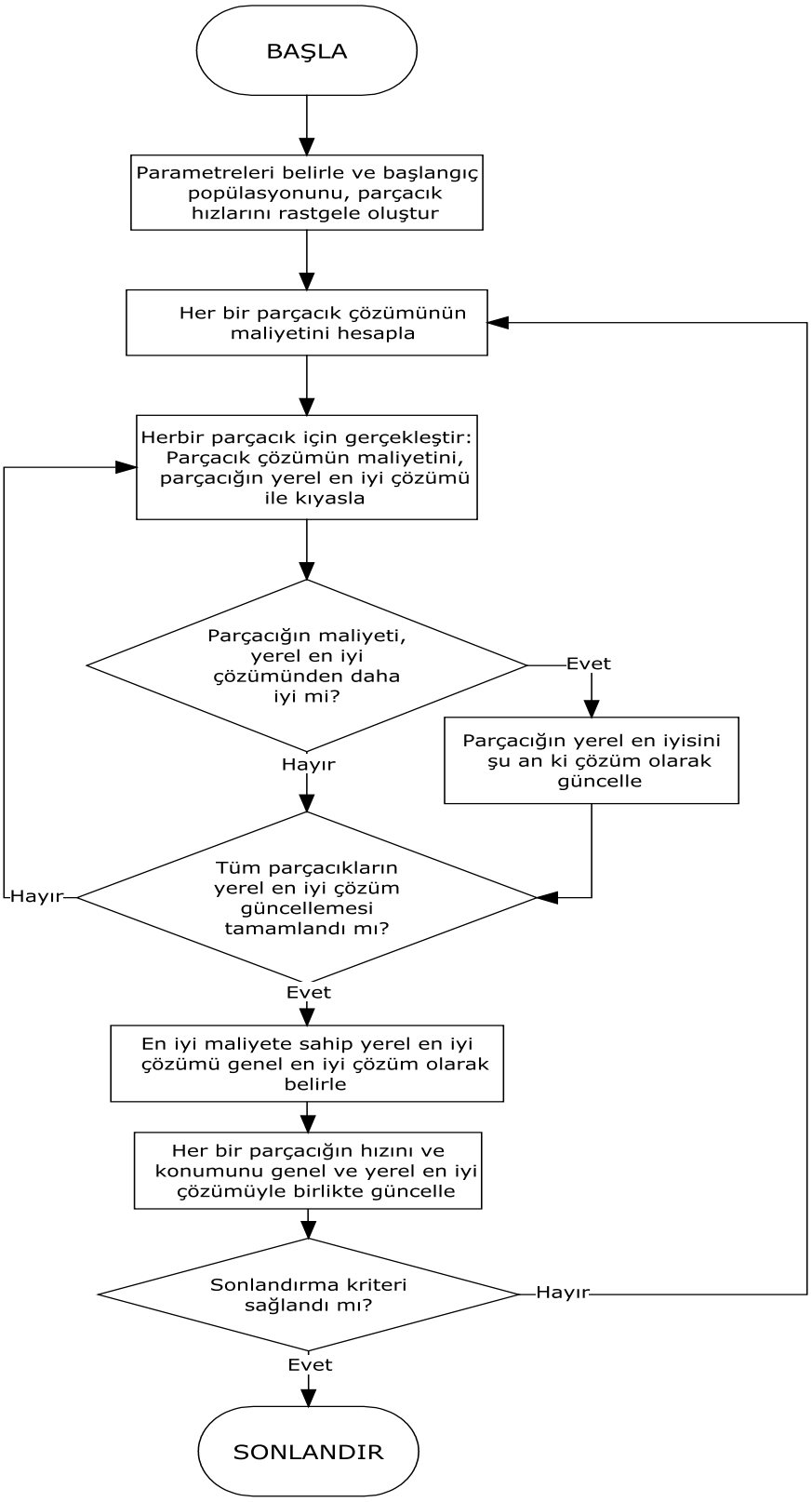
Shi ve Eberhart (1998) eski hız vektörünün yeni hız vektörüne etkisini belirlemek amacıyla w eylemsizlik ağırlığı parametresini hız denklemine eklemiştir. w, [0 1] aralığında değerler alabilir ve değerini belirlemek amacıyla faklı stratejiler geliştirilmiştir.

$$v_i{}^{t+1} = w.v_i{}^{t} + c_1.r_1{}^{t}.\left(p_{best_i}{}^{t} - x_i{}^{t}\right) + c_2.r_2{}^{t}.\left(g_{best}{}^{t} - x_i{}^{t}\right) \tag{4.27}$$

PSO'da eylemsizlik ağırlığı, yerel ve genel (global) arama yeteneği arasındaki dengeyi kurmak amacıyla da kullanılır. Değerin büyük seçilmesi genel aramayı, küçük seçilmesi de yerel aramayı kolaylaştırır. Bu nedenle her bir iterasyonda doğrusal bir şekilde azaltılarak genel arama ve yerel arama arasındaki denge sağlanmış olur. Böylece optimum sonuca daha az iterasyonda ulaşılabilir.

PSO algoritmasının akış şeması Şekil 4.5'de verilmiştir.

4.3.3. Akış şeması



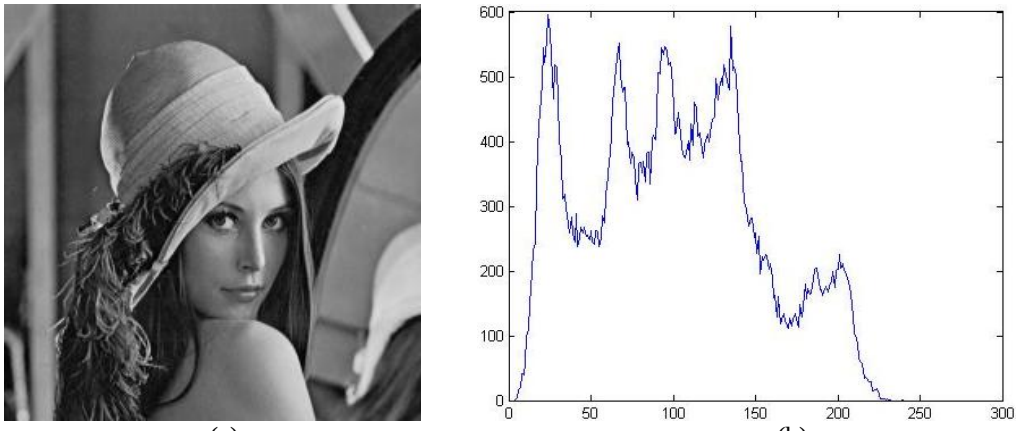
Şekil 4.5. Parçacık sürü optimizasyonu algoritmasının akış şeması

5. DENEYSEL SONUÇLAR VE PERFORMANS ANALİZİ

Bu bölümde metasezgisel algoritmalarla çoklu imge eşikleme gerçekleştirmek üzere Otsu'nun yöntemini amaç fonksiyonu olarak temel alan ve ilki sınıf içi değişintiyi minimize eden, ikincisi de sınıflar arası değişintiyi maksimize eden iki ayrı uygulama gerçekleştirilmiştir. Uygulamalar MATLAB 2009 programında kodlanmış olup Intel Core 2 Duo P8700 2.53 GHz işlemci ve 4 GB RAM kapasitesine sahip bir bilgisayarda gerçekleştirilmiştir.

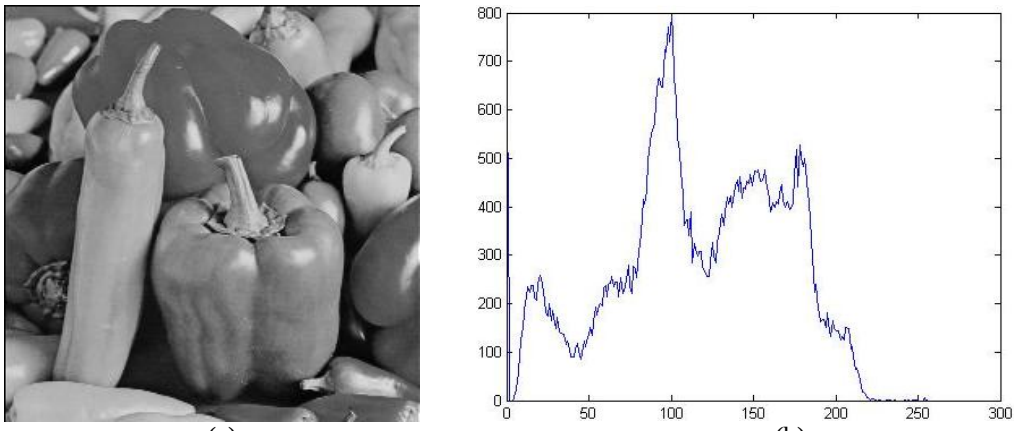
5.1. Sınıf İçi Değişinti ile İmge Eşikleme Uygulaması

Çalışmada çoklu imge eşikleme gerçekleştirmek üzere teknik yazında sıklıkla kullanılan test imgelerinden Lena, Peppers, House, Airplane, Lake ve Boats imgeleri seçilmiştir. Seçilen imgeler, Hammouche ve ark.'nın (2010) gerçekleştirdiği çalışmadan alınmıştır. İmgeler, 256x256 piksel boyutunda, "bmp" formatında olup 256 gri seviyeye sahiptir. İmgeler ve imgelere ait histogramlar Şekil 5.1- 5.6'daki gibidir.



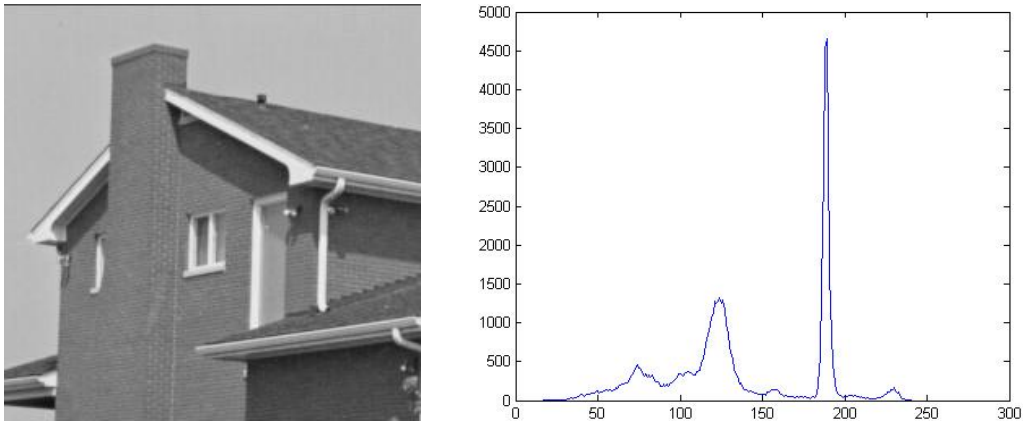
(a) (b)

Şekil 5.1. Kullanılan test imgeleri (a) Lena, (b) Lena imgesine ait histogram



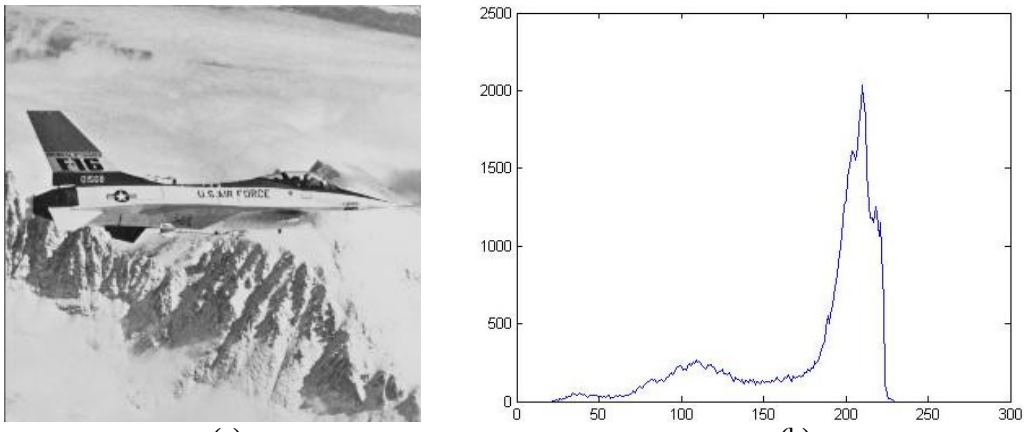
(a) (b)

Şekil 5.2. Kullanılan test imgeleri (a) Peppers, (b) Peppers imgesine ait histogram



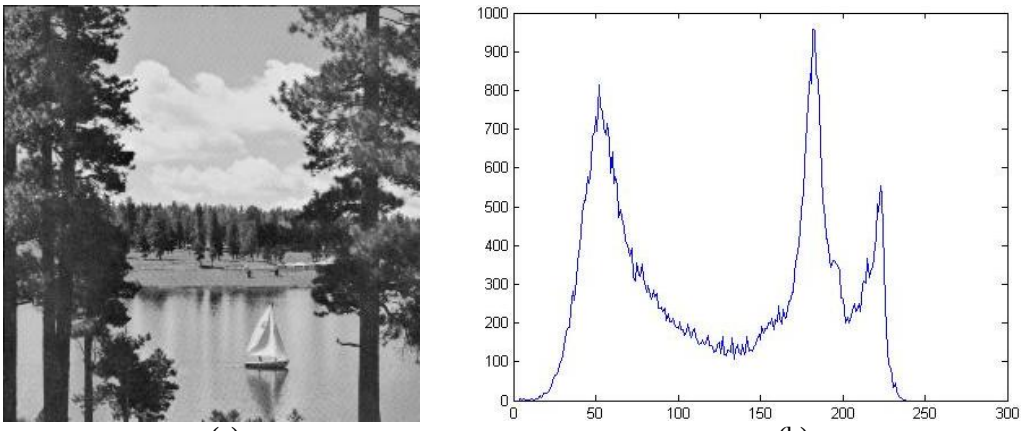
(a) (b)

Şekil 5.3. Kullanılan test imgeleri (a) House, (b) House imgesine ait histogram



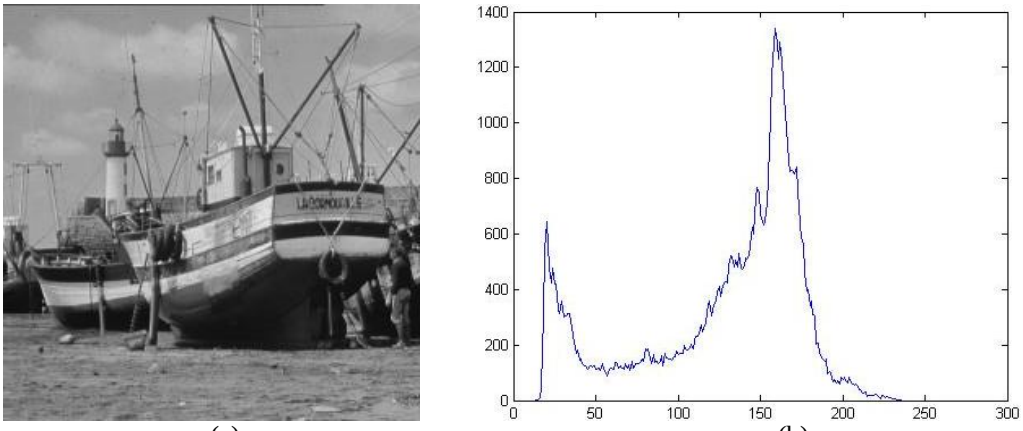
(a) (b)

Şekil 5.4. Kullanılan test imgeleri (a) Airplane, (b) Airplane imgesine ait histogram



(a) (b)

Şekil 5.5. Kullanılan test imgeleri (a) Lake, (b) Lake imgesine ait histogram



(a) (b)

Şekil 5.6. Kullanılan test imgeleri (a) Boats, (b) Boats imgesine ait histogram

Bu imgeler üzerinde çoklu imge eşikleme gerçekleştirmek üzere Otsu'nun (1979) yöntemi kullanılmıştır. Hem etraflı arama, hem de metasezgisel algoritmaların uygunluk değerlerinin belirlenmesinde amaç fonksiyonu olarak; sınıf içi değişintinin, imgenin toplam değişintisine oranı ile elde edilen Gri Ton Düzensizliği Ölçütü kullanılmıştır (Denklem 3.26). Bir imgenin toplam değişintisinin sabit olduğu göz önüne alınırsa en uygun eşik değerleri, Denklem 3.20'de ifade edilen imgenin sınıf içi değişintisi minimize edilerek belirlenmiştir. Buna bağlı olarak GTDS'nin amaç fonksiyonu olarak kullanılmasıyla uygunluk değerleri [0 1] aralığında normalize edilmiş değerlere sahip olacaktır ve 0'a daha yakın uygunluk değerine sahip çözüm daha iyi bir bölütleme gerçekleştiren bir çözümdür.

Çalışmada, Otsu'nun yöntemini kullanan etraflı (detaylı) arama ile imgeler 2'den 5'e kadar sınıfa bölütlenerek eşikleme gerçekleştirilmiş ve sonucunda elde edilen eşik değerleri, f_{opt} amaç fonksiyonu değerleri ve hesaplama süreleri Çizelge 5.1'de

verilmiştir. Etraflı arama sonucunda elde edilen amaç fonksiyonu değerleri, kullanılan yöntemin amaç fonksiyonuna ait genel minimum noktası ve elde edilen eşik değerleri ise ilgili imgeye göre amaç fonksiyonunu minimum yapan en uygun eşik değerleridir.

Çizelge 5.1. Test imgeleri üzerinde Otsu'nun yöntemi ile etraflı arama gerçekleştirilmesi sonucu elde edilen eşik değerleri, amaç fonksiyonu değerleri ve hesaplama süreleri

İmge	*K*	*Otsu*		
		Eşik Değerleri	Amaç fonksiyonu (f_{opt})	Hesaplama süresi (ms)
Lena	2	102	0.3128012303739900	5.55
	3	78-145	0.1438152862397140	375.62
	4	57-106-159	0.0710375037609998	29,271.37
	5	47-84-119-164	0.0428575875114826	1,743,617.17
Peppers	2	117	0.3028234202573920	50.91
	3	67-132	0.1355182600041680	484.41
	4	62-116-160	0.0754486001511938	36,113.53
	5	44-83-121-162	0.0493072659850905	2,421,685.09
House	2	147	0.2171193157402840	6.99
	3	96-155	0.0717776906839701	314.42
	4	82-112-158	0.0523300014485430	23,141.82
	5	82-112-156-205	0.0355574650386083	1,405,918.54
Boats	2	101	0.2132116778201910	6.49
	3	77-140	0.0998682872507520	306.58
	4	59-111-151	0.0605269320356135	23,567.38
	5	56-104-141-168	0.0393339123897511	1,381,348.63
Lake	2	125	0.1452788445464620	6.71
	3	88-155	0.0776876954362840	340.05
	4	80-140-193	0.0445815164548689	28,606.44
	5	69-111-158-197	0.0280979914547629	1,738,891.15
Airplane	2	154	0.1694023170470150	7.48
	3	117-175	0.0974979570016160	249.31
	4	94-146-191	0.0609037629636396	18,353.96
	5	88-132-174-203	0.0392906184771508	1,034,722.09

Ateş böceği Algoritması, Parçacık Sürü Optimizasyonu ve Guguk kuşu Optimizasyonu Algoritması'yla test imgeleri üzerinde çoklu imge eşikleme gerçekleştirmek amacıyla ilk olarak parametreler belirlenmiştir. Algoritmaların performansını büyük ölçüde etkileyen parametreler, teknik yazında önerilen değerlere

uygun olarak seçilmiştir (Hammouche ve ark., 2010; Horng ve Jiang, 2010; Rajabioun, 2011). Performans kıyaslaması yapabilmek adına popülasyon sayısı, maksimum iterasyon sayısı üç metasezgisel algoritma için de aynı seçilmiştir. PSO, AA ve GOA için belirlenen parametre değerleri sırasıyla Çizelge 5.2, 5.3 ve 5.4'de verilmiştir.

Çizelge 5.2. Seçilen PSO parametreleri

PSO		
	Popülasyon sayısı (N)	100
	Maksimum iterasyon sayısı	100
	Ağırlık (w)	0.5
	Maksimum hız sınırı (Vmax)	5/255
	Minimum hız sınırı (Vmin)	-5/255
	Öğrenme faktörleri (c_1 ve c_2)	2

Çizelge 5.3. Seçilen AA parametreleri

AA		
	Popülasyon sayısı (N)	100
	Maksimum iterasyon sayısı	100
	Etkililik parametresi (β_0)	1
	Işık soğurma katsayısı (γ)	1
	Rastsallık katsayısı (α)	0.01

Çizelge 5.4. Seçilen GOA parametreleri

GOA		
	Başlangıç popülasyonu sayısı (N)	20
	Maksimum popülasyon sayısı (N_{max})	100
	Maksimum iterasyon sayısı	100
	Grup sayısı (k)	2
	Maksimum yumurta sayısı (y_{max})	4
	Minimum yumurta sayısı (y_{min})	2
	Yumurtlama yarıçapı katsayısı (α)	10
	Yaklaşım parametresi (λ)	0.5

Deneysel çalışmalarda gri seviye imgelere ait [0 255] arasındaki piksel değerleri [0 1] aralığına normalize edilmiş ve her üç metasezgisel algoritma için başlangıç popülasyonu [0 255] arasında tamsayı değerler yerine [0 1] aralığında rasyonel değerler olarak rastgele başlatılmıştır. Dolayısıyla arama uzayı [0 1] aralığıyla kısıtlı olup her bir hesaplama işlemi esnasında çözümler 255 değeri ile çarpılıp arama uzayı tekrar [0 255] aralığına genişletilerek değişinti hesapları gerçekleştirilmiştir.

Deneysel çalışmalarda kullanılan tüm algortimalar için Gri seviye imgelerin piksel değerleri [0 1] aralığına normalize edilmiştir. Arama uzayının [0 1] aralığına indirgenmesi sebebiyle Vmax ve Vmin parametreleri sırasıyla 5/255 ve -5/255 olarak belirlenmiştir.

GOA'nın yumurta yayma aşamasında; ortam merkezinin yumurtlama yarıçapı içerisindeki dairesel bölge, üretilecek yumurta sayısı kadar lineer artan radyan açı dizisinin etkisiyle parçalara ayrılmış ve her bir parçaya birer yumurta gelecek şekilde rastgele olarak dağıtılmıştır. Böylece algoritma, bir ortamın etrafındaki tüm alana hakim olacak şekilde geliştirilmiştir.

Bu çalışmada imgeler üzerinde gerçekleştirilen çoklu eşikleme uygulamasında Bölüm 4'te GOA'da tanımlanan φ açısı ile sapma işlemi algoritma sonuçlarında iyileştirici bir etki göstermediğinden dolayı kullanılmamıştır.

Çoklu imge eşikleme gerçekleştirmek üzere her bir metasezgisel algoritma, 50 kez (tur) çalıştırılmıştır (koşturulmuştur) ve her bir i. tur aşamasında elde edilen en iyi f_i^* çözümlerinin uyguluk değeri ortalamaları olan f_{ort} hesaplanmıştır. f_i^* ortalamalarına bağlı olarak 50 tur sonunda elde edilen ortalama uygunluk değeri Denklem 5.1'deki gibi hesaplanmıştır.

$$f_{ort} = \frac{\sum_{i=1}^{50} f_i^*}{50} \tag{5.1}$$

f_i^* değerlerinin hesaplanmasının ardından ortalama uygunluk değerlerinin standart sapması S Denklem 5.2'de görüldüğü gibi hesaplanmıştır.

$$S = \sqrt{\frac{\sum_{i=1}^{50} (f_i^* - f_{ort})^2}{50}} \tag{5.2}$$

Çoklu imge eşikleme problemi üzerinde; metasezgisel algoritmalarla elde edilen çözümlerin kalitesini kıyaslamak amacıyla çoklu eşikleme sonunda elde edilen f_{ort} uygunluk değerleri kıyas kriteri olarak kullanılmıştır ve daha küçük f_{ort} uygunluk değerine sahip çözüm üreten algoritma, daha iyi sonuç veren algoritma olarak kabul edilmiştir.

Metasezgisel algoritmaların problemin çözümüne yakınsama sürelerini kıyaslayabilmek adına her bir i. tur, o turda elde edilen f_i^* en iyi çözümün, amaç

fonksiyonunun genel en iyi değeri olan f_{opt} değerine belirli bir ε hata miktarından küçük veya eşit bir değerde yaklaştığı an sonlandırılmıştır. Deneyde kullanılan ve her bir turu sonlandıran sonlandırma kriteri, Denklem 5.3'de ifade edilmektedir.

$$\left|f_i^* - f_{opt}\right| \leq \varepsilon = 10^{-9} \tag{5.3}$$

GOA, PSO ve AA algoritmalarının çoklu imge eşikleme problemine uyarlanmasını ifade eden sözde kodlar sırasıyla Şekil 5.7-5.9'da verilmiştir.

Rastgele seçilen eşik değerleriyle $x=[x_1, x_2, \ldots, x_d]$ *ortam merkezleri üret*
While $it \leq it_{max}$ *veya* $\left|g_{best} - f_{opt}\right| \leq \varepsilon$
    For i=1 den N e kadar gerçekleştir
        $y_i = U(y_{min}, y_{max})$
    End
    For i=1 den N e kadar gerçekleştir
        $YY_i = \alpha \; x \; \frac{y_i}{y_{top}} \; x \; (y_{max} - y_{min})$
    For j=1 den y_i *ye kadar gerçekleştir*
        For k=1 den d ye kadar gerçekleştir
            ${x_{i,k}}^j = x_{i,k} + U(0,1) * YY_i$
        End
    End
End
$N=N+y_{top}$
x_i *çözümlerinin* $f(x_i)$ *maliyetlerini hesapla*
x_i *çözümlerini maliyete göre sırala ve en kötü %p kadarını yok et*
    If $N_{max} \leq N$
        x_i *çözümlerini maliyete göre en iyi* N_{max} *adet çözümü al, gerisini yok et*
        $N=N_{max}$
    End
Çözüm uzayındaki çözüm konumlarını $C_{1,\ldots,K}$ *gruplarına ayır*
    For i=1 den K ye kadar gerçekleştir
        $ort(C_i) = \frac{\sum_{x \epsilon C_i} f(x)}{sınıftaki\ çözüm\ sayısı}$
    End
$hedef = Arg\ Min_{i \epsilon \{1,\ldots,K\}}\ ort(C_i)$ *[Minimizasyon problemi]*
$h_ortam = Arg\ Min_{x \epsilon C_{hedef}}\ f(x)$ *[Minimizasyon problemi]*
    For i=1 den N e kadar gerçekleştir
        For k=1 den d ye kadar gerçekleştir
            $x_{i,k} = h_ortam_k + \lambda * (x_{i,k} - h_ortam_k)$
        End
    End
End

Şekil 5.7. Guguk kuşu optimizasyonu algoritmasının sözde kodu

Rastgele seçilen eşik değerleriyle $x_{i\epsilon 1,..,N}=[x_{i,1}, x_{i,2}, \ldots, x_{i,d}]$ *parçacık çözümlerini üret*
While $it \leq it_{max}$ *veya* $\left|g_{best} - f_{opt}\right| \leq \varepsilon$
    For i=1 den N e kadar gerçekleştir
        x_i *çözümünün f(*x_i*) maliyetini hesapla*
        If ($f(p_{best_i}) \geq f(x_i^{it})$ *) [Minimizasyon problemi]*
            $p_{best_i} = x_i^{it}$
        End
    End
    For i=1 den N e kadar gerçekleştir
        If ($f(g_{best}) \geq f(p_{best_i})$*) [Minimizasyon problemi]*
            $g_{best} = p_{best_i}$
        End
    End
    For i=1 den N e kadar gerçekleştir
        For k=1 den d ye kadar gerçekleştir
        $v_{i,k}^{t+1} = w.v_{i,k}^{t} + c_1.r_1^{t}.\left(p_{best_{i,k}}^{t} - x_{i,k}^{t}\right) + c_2.r_2^{t}.(g_{best_k}^{t} - x_{i,k}^{t})$
        $x_{i,k}^{t+1} = x_{i,k}^{t} + v_{i,k}^{t+1}$
        End
    End
End

Şekil 5.8. Parçacık sürü optimizasyonu algoritmasının sözde kodu

Rastgele seçilen eşik değerleriyle $x_{i\epsilon 1,..,N}=[x_{i,1}, x_{i,2}, \ldots, x_{i,d}]$ *ateş böceği çözümleri üret*
Aday çözümlerin f(x) maliyetlerini (parlaklıklarını) hesapla
$x_{en\ iyi} = Arg\ Min_{n\epsilon 1,\ldots,N}\, f(x_n)$ *[Minimizasyon problemi]*
While $it \leq it_{max}$ *veya* $\left|f(x_{max}) - f_{opt}\right| \leq \varepsilon$
    For i=1 den N e kadar gerçekleştir
        For j=i den N e kadar gerçekleştir
            If f(x_i*)≥ f(*x_j*) [Minimizasyon problemi]*
                $r_{i,j} = d(x_i, x_j)$ *ve* $\beta = \beta_0.e^{-\gamma.r_{i,j}}$
                For k=1 den d ye kadar gerçekleştir
                $u_{i,k} = \alpha.(rand1 - \frac{1}{2})$
                $x_{i,k} = (1-\beta).x_{i,k} + \beta.x_{j,k} + u_{i,k}$
                End
            End
        End
    End
    For k=1 den d ye kadar gerçekleştir
        $u_{max,k} = \alpha.(rand1 - \frac{1}{2})$
        $x_{max,k} = x_{max,k} + u_{i,k}$
    End
End

Şekil 5.9. Ateş böceği algoritmasının sözde kodu

Sözde kodlarda belirtilen it, o anki iterasyonu; it_{max}, maksimum iterasyon sayısını; d, çözümlerin boyutunu ve $x_{en\ iyi}$ ise en iyi maliyete sahip olan x çözümünü ifade etmektedir.

Çizelge 5.5'de AA, GOA, PSO ile çoklu eşikleme sonucu elde edilen f_{ort} değerleri ile S standart sapmaları verilmiştir.

Çizelge 5.5. 50 tur sonunda elde edilen f_{ort} uygunluk değerleri ve S standart sapmaları (GOA, AA, PSO)

İmgeler	**K**	**GOA**		**AA**		**PSO**	
		f_{ort}	S	f_{ort}	S	f_{ort}	S
Lena	2	0.3128012304	2.78E-16	0.3128012304	2.78E-16	0.3128012304	2.78E-16
	3	0.1438152862	2.78E-17	0.1438152862	2.78E-17	0.1438152862	2.78E-17
	4	0.0710375038	6.94E-17	0.0710375038	6.94E-17	0.0710375038	6.94E-17
	5	0.0428575875	3.47E-17	0.0428575875	3.47E-17	0.0428575875	3.47E-17
Peppers	2	0.3028234203	1.67E-16	0.3028234203	1.67E-16	0.3028234203	1.67E-16
	3	0.1355182600	5.55E-17	0.1355182600	5.55E-17	0.1355182600	5.55E-17
	4	0.0754486002	0	0.0754486002	0	0.0754486002	0
	5	0.0493072660	6.25E-17	0.0493121171	3.40E-05	0.0493072660	6.25E-17
House	2	0.2171193157	2.22E-16	0.2171193157	2.22E-16	0.2171193157	2.22E-16
	3	0.0717776907	6.94E-17	0.0717776907	6.94E-17	0.0717776907	6.94E-17
	4	0.0523300014	4.16E-17	0.0523326353	1.84E-05	0.0527338641	8.62E-04
	5	0.0355574650	2.08E-17	0.0356448039	6.11E-04	0.0355574650	2.08E-17
Boats	2	0.2132116778	8.33E-17	0.2132116778	8.33E-17	0.2132116778	8.33E-17
	3	0.0998682873	4.16E-17	0.0998682873	4.16E-17	0.0998682873	4.16E-17
	4	0.0605269320	4.16E-17	0.0605269320	4.16E-17	0.0605269320	4.16E-17
	5	0.0393339124	6.94E-18	0.0393339124	6.94E-18	0.0393339124	6.94E-18
Lake	2	0.1452788445	5.55E-17	0.1452788445	5.55E-17	0.1452788445	5.55E-17
	3	0.0776876954	9.71E-17	0.0776876954	9.71E-17	0.0776876954	9.71E-17
	4	0.0445815165	0	0.0445815165	0	0.0445815165	0
	5	0.0280979915	2.78E-17	0.0280979915	2.78E-17	0.0280979915	2.78E-17
Airplane	2	0.1694023170	2.78E-17	0.1694023170	2.78E-17	0.1694023170	2.78E-17
	3	0.0974979570	2.78E-17	0.0974979570	2.78E-17	0.0974979570	2.78E-17
	4	0.0609037630	3.47E-17	0.0609037630	3.47E-17	0.0609037630	3.47E-17
	5	0.0392906185	4.86E-17	0.0392906185	4.86E-17	0.0392906180	4.86E-17

Metasezgisel tekniklerin etraflı aramaya göre hesaplama zamanına olan etkisini araştırmak üzere bu tekniklerle çoklu eşikleme esnasında her bir i. turda harcanan t_i hesaplama süreleri ve gerçekleştirilen n_i iterasyon sayıları tutulmuştur. Çizelge 5.6'da

GOA, AA ve PSO ile çoklu eşikleme gerçekleştirmek üzere algoritmaların 50 kez çalıştırılması sonrasında elde edilen t_i ve n_i değerlerinin ortalamaları olan t_{ort} ve n_{ort}verilmiştir.

Çizelge 5.6. 50 tur sonunda ortalama t_{ort} hesaplama süreleri ve n_{ort} iterasyon sayıları (GOA, AA, PSO)

İmgeler	K	GOA		AA		PSO	
		t_{ort} (ms)	n_{ort}	t_{ort} (ms)	n_{ort}	t_{ort} (ms)	n_{ort}
Lena	2	8.75	1.30	16.12	1.06	3.38	2.08
	3	26.42	3.00	41.29	3.02	14.00	6.82
	4	59.16	5.76	80.79	6.32	23.97	10.78
	5	111.83	10.16	136.33	10.04	34.13	14.42
Peppers	2	8.25	1.14	15.22	1.00	3.11	1.94
	3	31.51	3.34	47.38	3.58	14.13	7.32
	4	69.57	6.58	75.47	6.22	24.92	11.48
	5	128.64	11.48	134.15	11.66	40.90	17.00
House	2	7.58	1.18	15.70	1.06	3.17	2.10
	3	30.43	3.16	35.40	2.78	12.37	6.62
	4	65.76	6.52	101.40	9.36	57.73	27.20
	5	126.94	11.78	162.23	13.28	29.78	13.90
Boats	2	9.33	1.28	15.75	1.04	2.80	1.90
	3	26.25	3.02	42.07	3.30	12.73	6.74
	4	62.59	6.12	87.72	6.72	23.29	11.20
	5	146.83	13.42	126.14	9.82	30.62	13.90
Lake	2	8.06	1.24	14.61	1.02	3.01	1.90
	3	31.74	3.36	39.10	3.12	12.74	6.56
	4	70.25	6.60	78.78	6.18	21.99	10.56
	5	126.72	10.94	144.68	10.38	32.37	14.06
Airplane	2	7.65	1.18	15.10	1.04	2.92	1.84
	3	26.68	3.06	36.19	2.80	12.76	7.00
	4	61.55	6.04	76.90	5.60	23.28	11.28
	5	99.38	9.22	160.12	12.94	36.02	15.34

Çizelge 5.5'de görülen f_{ort} değerleri, algoritmaların test imgeleri üzerinde daha iyi bir çözüm üretip üretmediği yani f_{opt} değerine yakınlığı (çözüm kalitesi); S değerleri, algoritmanın stabilitesi (gürbüzlüğü) hakkında (her bir turda elde edilen çözümler arasındaki benzerlik) değerlendirme yapmaya imkan vermektedir. Çizelge 5.5'de görüldüğü üzere her bir algoritma, seçilen eşik değeri (K-1) sayısının küçük

olması durumunda oldukça kaliteli çözümler üretmiş ve stabil bir davranış göstermiştir. Tüm test imgeleri üzerinde 1 ve 2 adet eşik değeri çözümlerinin aranması ile eşikleme gerçekleştirildiği taktirde algoritmaların her biri için ilgili test imgesi üzerinde aynı f_{ort} ve S değerlerine sahip olduğu görülmüştür ve bu f_{ort} değerleri Çizelge 5.1'de belirtilen Otsu'nun yöntemi ile gerçekleştirilmiş etraflı arama sonucu elde edilen f_{opt} değerleriyle aynı olduğu görülmüştür.

Peppers ve House imgeleri dışında diğer test imgeleri üzerinde 3 ve 4 adet eşik değeri çözümleri ile eşikleme gerçekleştirildiği taktirde algoritmaların yine ilgili test imgesi üzerinde aynı f_{ort} ve S değerlerine sahip olduğu görülmüştür ve bu f_{ort} değerleri Çizelge 5.1'de ifade edilen Otsu'nun yöntemi ile gerçekleştirilmiş etraflı arama sonucu elde edilen f_{opt} değerleriyle aynıdır.

Peppers test imgesi üzerinde 4 adet eşik değeri seçilerek gerçekleştirilen imge eşikleme işlemlerinde f_{ort} ve S değerlerinde farklılıklar saptandığı görülmüştür. GOA ve PSO ile elde edilen f_{ort} ve S değerleri aynı iken AA ile elde edilen değerler diğer iki algoritmaya göre daha büyük çıkmıştır. Aynı durum House imgesi üzerinde 4 adet eşik değeri seçilerek gerçekleştirilen eşikleme sonucunda da görülmektedir. PSO ve GOA'nın, AA'ya göre daha kaliteli ve daha stabil bir şekilde çözümler ürettiği görülmüştür.

House imgesinin üzerinde 3 eşik değeri seçilerek gerçekleştirilen eşikleme işlemi sonucunda GOA'nın PSO ve AA'dan daha kaliteli ve stabil çözümler ürettiği görülmüştür. AA'nın f_{ort} ve S değerleri, PSO'ya göre daha küçüktür ve dolayısıyla AA'nın PSO'ya göre daha kaliteli ve stabil bir çözüm ürettiği söylenebilir.

Genel bir değerlendirme yapmak gerekirse GOA, diğer iki algoritmaya göre daha stabil, kaliteli çözümler ve iyi sonuçlar üretmiştir. Tüm imgeler üzerinde PSO ve AA'ya göre ya aynı ya da daha küçük standart sapma değerlerine ve Çizelge 5.1'de verilen f_{opt} değerleri ile aynı değerlere sahip çözümler üretmiştir.

Sonlandırma kriteri olarak kullanılan ε değerine bağlı olarak her bir algoritmanın bir turda devam ettirdiği iterasyon sayısı farklılık göstermektedir. Eğer algoritma, ε hatayla Çizelge 5.1'de verilen f_{opt} değerine 100 iterasyon içinde yaklaşamıyorsa ilgili tur sonlanır. Çizelge 5.6'da görüldüğü üzere her 3 algoritmanın hesaplama zamanı, tek eşik değeri seçimi dışında Otsu'nun yöntemi ile gerçekleştirilen etraflı aramanın hesaplama zamanına göre çok daha küçük olduğu görülmüştür. Ayrıca Çizelge 5.6 göz önüne alınarak PSO'nun, GOA ve AA'ya göre daha kısa bir sürede

uygun çözüme yakınsadığı görülmüştür. GOA'nın ise AA'ya göre daha kısa bir sürede uygun çözüme yakınsadığı söylenebilir.

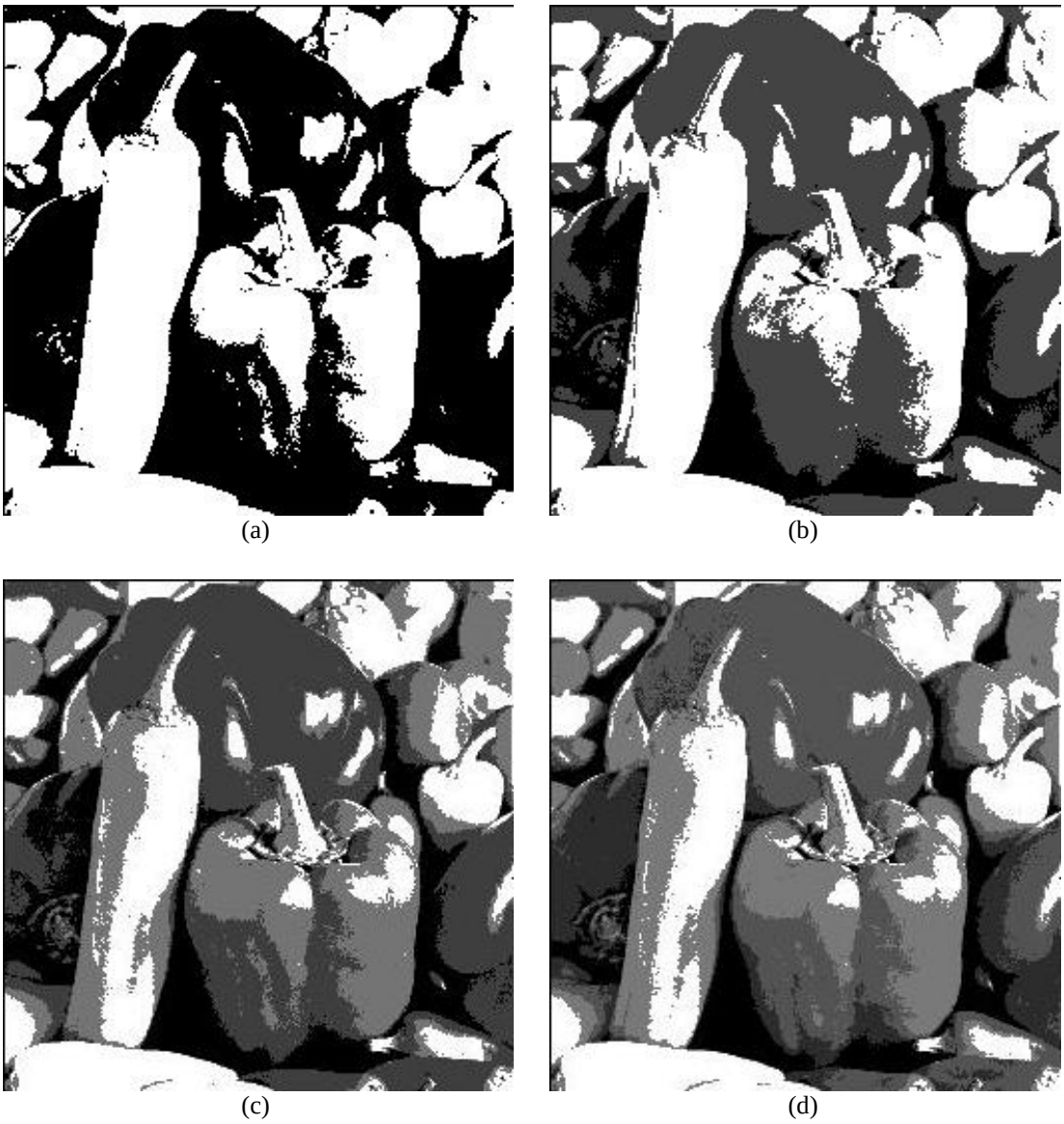
Şekil 5.10-5.15'te Otsu'nun yöntemi ile test imgeleri üzerinde gerçekleştirilen etraflı (detaylı) arama sonucunda 1'den 4 adete kadar belirlenen eşik değerlerini kullanarak imgelerin bölütlenmesi sonucunda oluşan sınıflar, farklı gri seviye değerleriyle temsil edilerek verilmiştir.



(a) (b)

(c) (d)

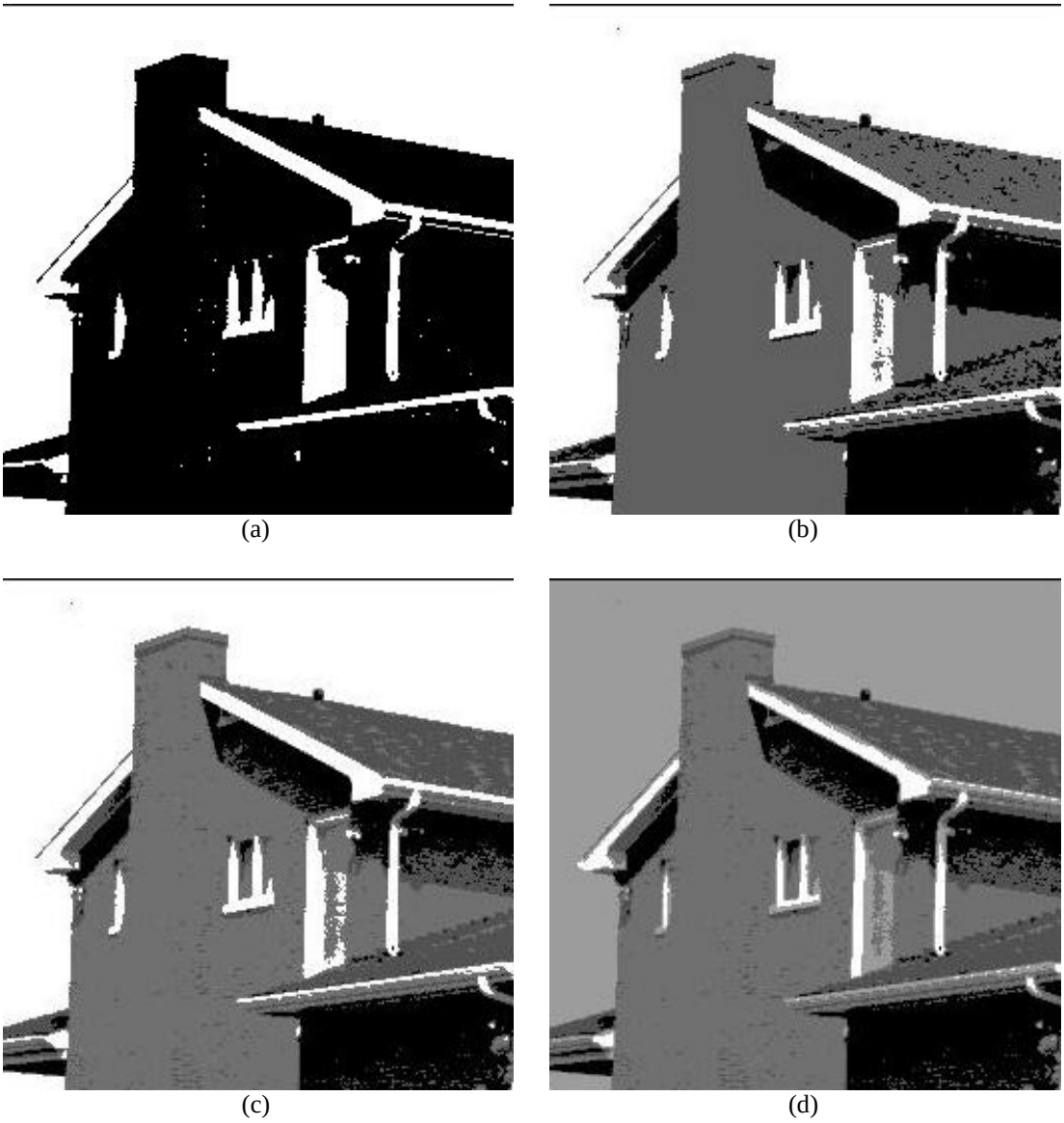
Şekil 5.10. Lena imgesinin eşiklenmiş hali: (a) 2 gri seviyeli imge (b) 3 gri seviyeli imge (c) 4 gri seviyeli imge (d) 5 gri seviyeli imge



(a) (b)

(c) (d)

Şekil 5.11. Peppers imgesinin eşiklenmiş hali: (a) 2 gri seviyeli imge (b) 3 gri seviyeli imge (c) 4 gri seviyeli imge (d) 5 gri seviyeli imge



(a) (b)

(c) (d)

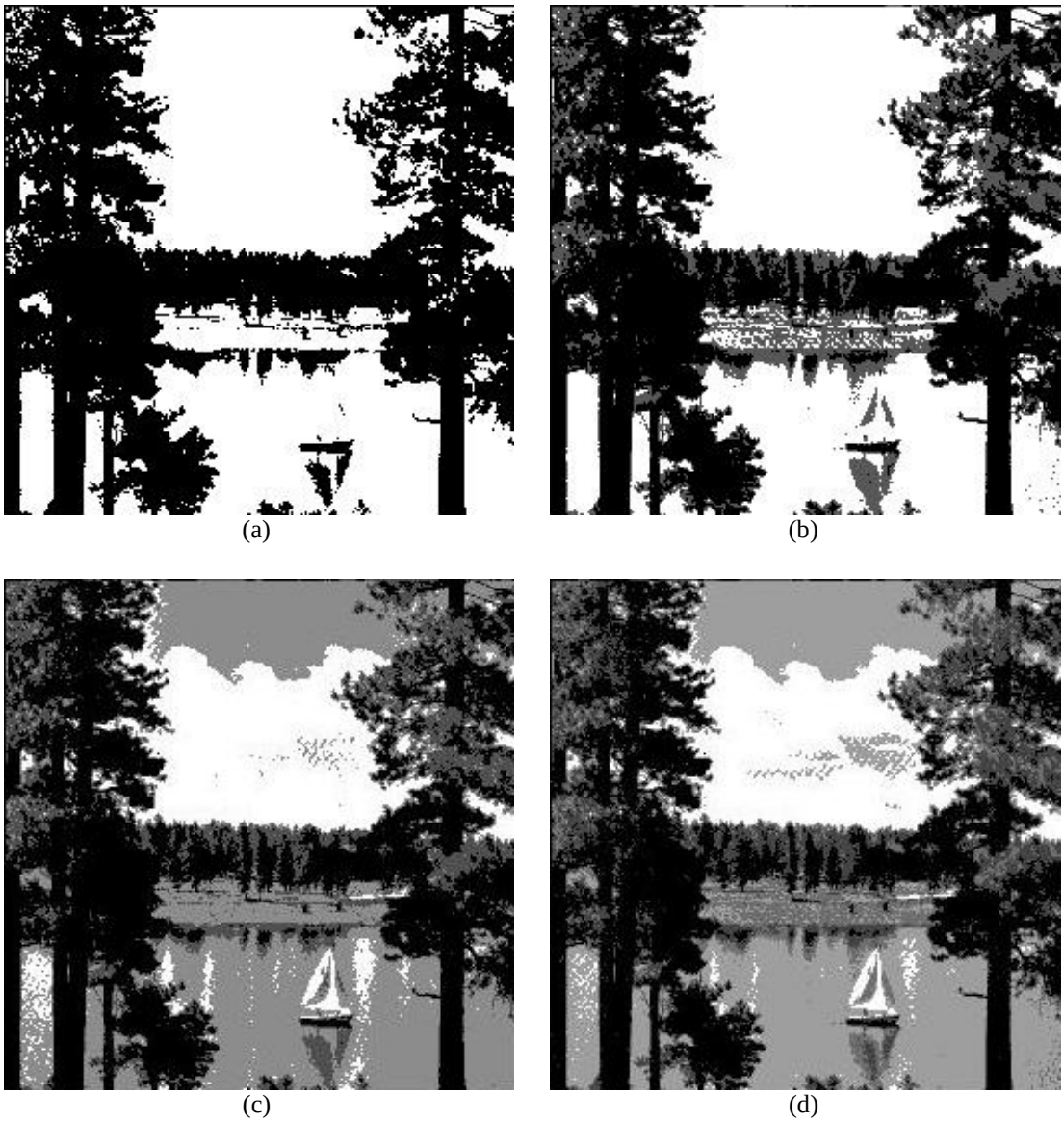
Şekil 5.12. House imgesinin eşiklenmiş hali: (a) 2 gri seviyeli imge (b) 3 gri seviyeli imge (c) 4 gri seviyeli imge (d) 5 gri seviyeli imge



(a) (b)

(c) (d)

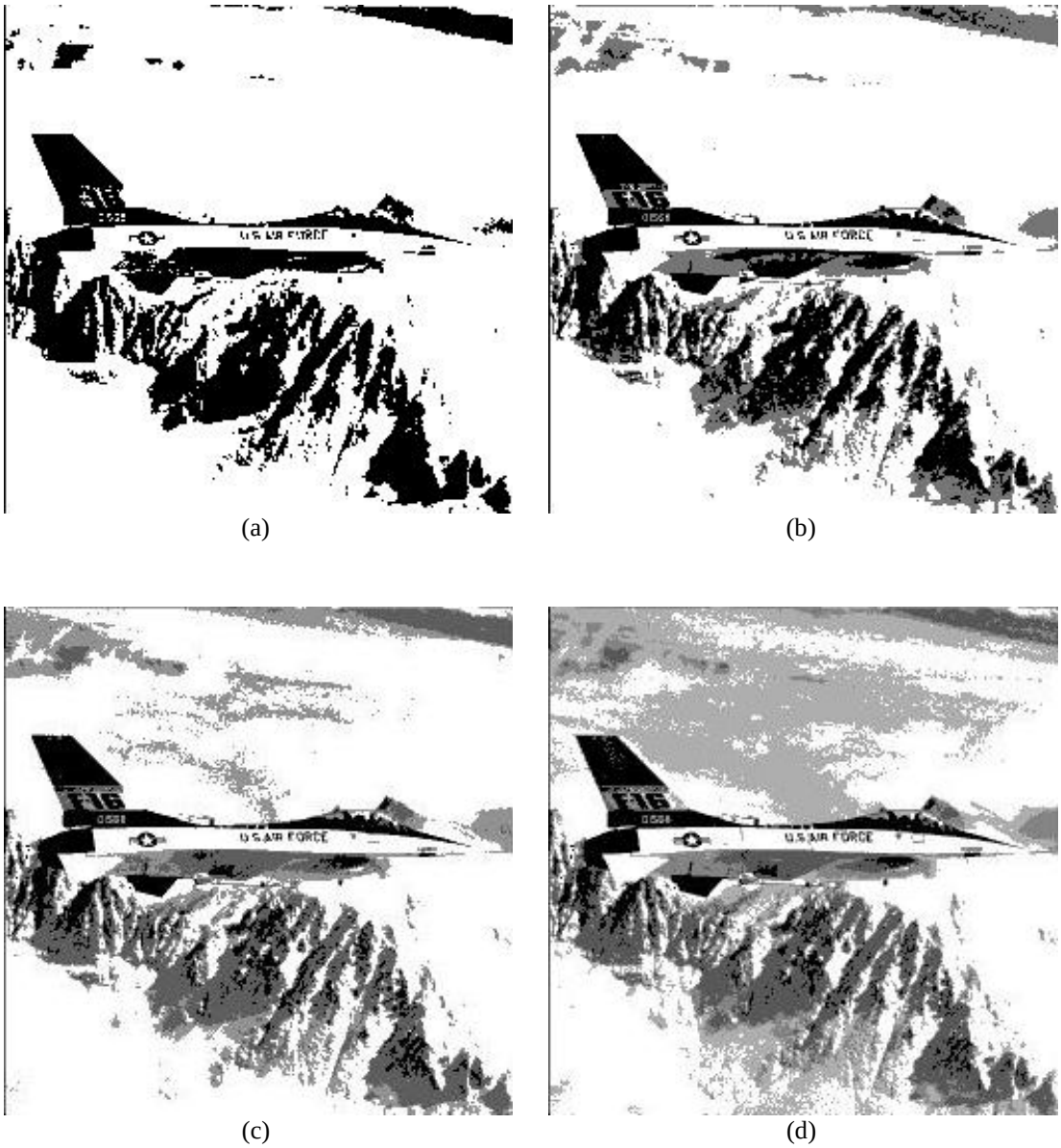
Şekil 5.13. Boats imgesinin eşiklenmiş hali: (a) 2 gri seviyeli imge (b) 3 gri seviyeli imge (c) 4 gri seviyeli imge (d) 5 gri seviyeli imge



(a) (b)

(c) (d)

Şekil 5.14. Lake imgesinin eşiklenmiş hali: (a) 2 gri seviyeli imge (b) 3 gri seviyeli imge (c) 4 gri seviyeli imge (d) 5 gri seviyeli imge



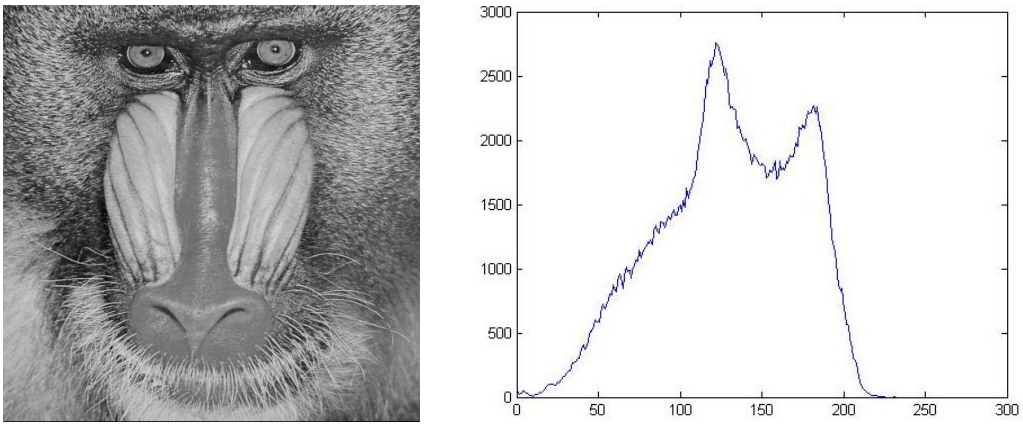
(a) (b)

(c) (d)

Şekil 5.15. Airplane imgesinin eşiklenmiş hali: (a) 2 gri seviyeli imge (b) 3 gri seviyeli imge (c) 4 gri seviyeli imge (d) 5 gri seviyeli imge

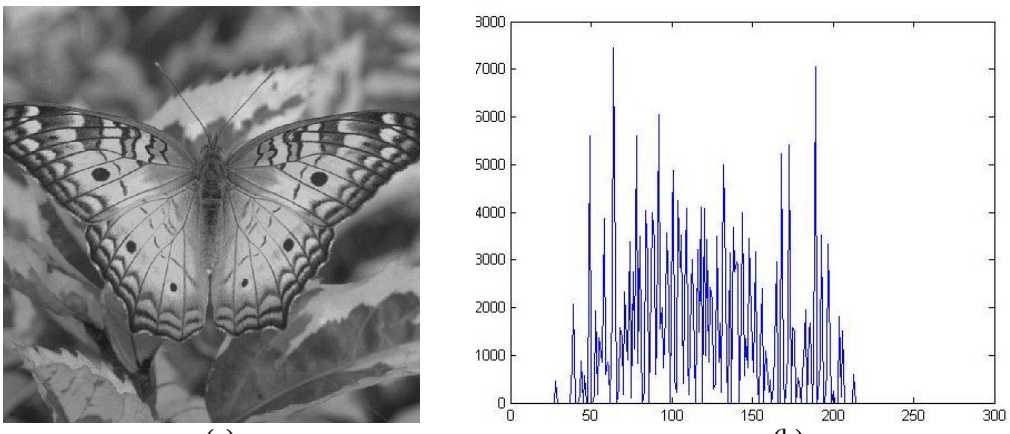
5.2. Sınıflar Arası Değişinti ile İmge Eşikleme Uygulaması

Sınıflar arası değişintinin maksimizasyonu ile imge eşikleme gerçekleştirmek üzere teknik yazında sıklıkla kullanılan test imgelerinden Baboon, Butterfly, Hunter, Livingroom ve Road imgeleri kullanılmıştır. Kullanılan imgeler, 512x512 boyutuna sahip "gif" formatında 256 gri seviyeli imgelerdir. Bu imgeler Granada Üniversitesi Bilgisayar Bilimleri ve Yapay Zeka Bölümü'nün gri seviyeli standart test imgeleri veri setinden alınmıştır (http://decsai.ugr.es/cvg/CG/base.htm). Şekil 5.16- 5.20'de seçilen test imgeleri ve imgelere ait gri seviye dağılımlarını gösteren histogramlar sunulmuştur.



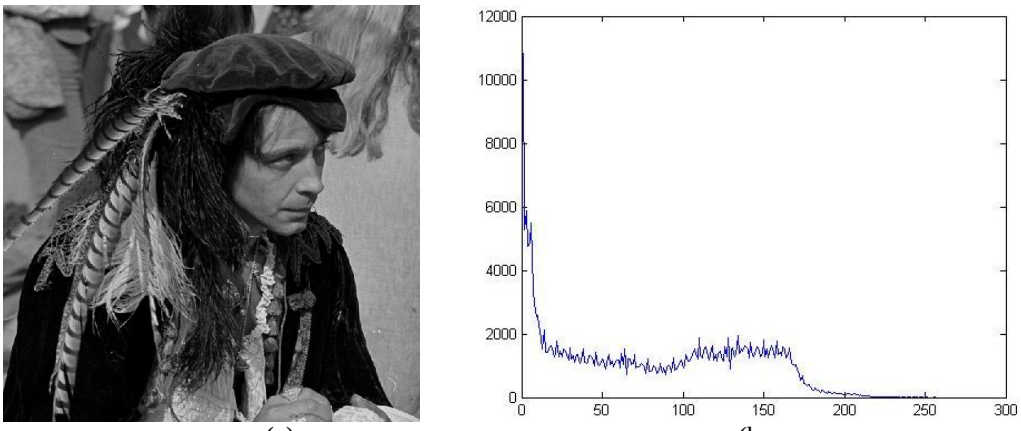
(a) (b)

Şekil 5.16. Kullanılan test imgeleri (a) Baboon, (b) Baboon imgesine ait histogram



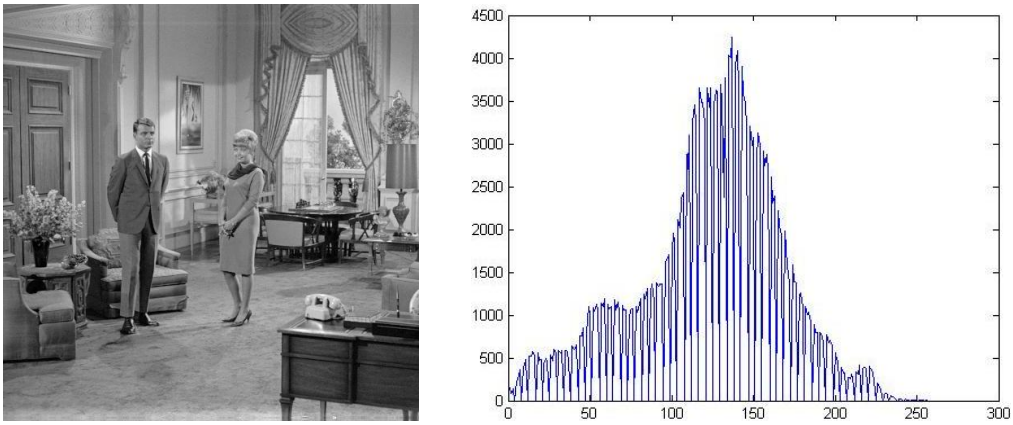
(a) (b)

Şekil 5.17. Kullanılan test imgeleri (a) Butterfly, (b) Butterfly imgesine ait histogram



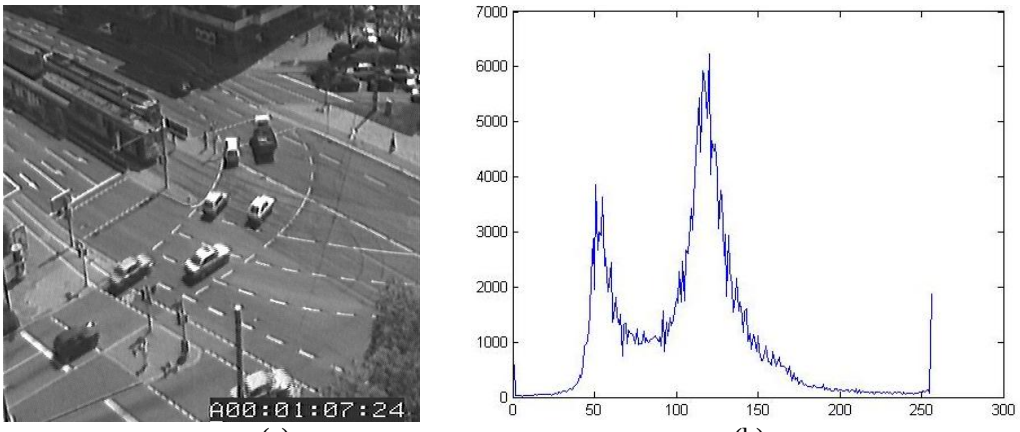
(a) (b

Şekil 5.18. Kullanılan test imgeleri (a) Hunter, (b) Hunter imgesine ait histogram



(a) (b)

Şekil 5.19. Kullanılan test imgeleri (a) Livingroom, (b) Livingroom imgesine ait histogram



(a) (b)

Şekil 5.20. Kullanılan test imgeleri (a) Road, (b) Road imgesine ait histogram

Bu imgeler üzerinde çoklu imge eşikleme gerçekleştirmek üzere Otsu'nun (1979) yöntemi kullanılmıştır. Uygunluk değerlerini belirlemek amacıyla hem etraflı arama, hem de metasezgisel algoritmalar için amaç fonksiyonu olarak Denklem 3.21'de

ifade edilen sınıflar arası değişinti kullanılmıştır. Denklem 3.22'de ifade edildiği gibi imgenin sınıflar arası değişintisi maksimize edilerek en uygun eşik değerleri belirlenmiştir.

Çalışmada, Otsu'nun yönteminin kullanıldığı etraflı (detaylı) arama ile imgeler 2'den 5'e kadar sınıfa bölütlenerek eşikleme gerçekleştirilmiş ve sonucunda elde edilen eşik değerleri, f_{opt} amaç fonksiyonu değerleri ve hesaplama süreleri Çizelge 5.7'de verilmiştir. Etraflı arama sonucunda elde edilen amaç fonksiyonu değerleri, kullanılan yöntemin amaç fonksiyonuna ait genel maksimum noktası; elde edilen eşik değerleri ise ilgili imgeye göre amaç fonksiyonunu maksimum yapan en uygun eşik değerleridir.

Çizelge 5.7. Test imgeleri üzerinde Otsu'nun yöntemi ile etraflı arama gerçekleştirilmesi sonucu elde edilen eşik değerleri, amaç fonksiyonu değerleri ve hesaplama süreleri

İmge	*K*	*Otsu*		
		Eşik Değerleri	F_{opt}	Süre (ms)
Baboon	2	128	1,219.050430	8.69
	3	97-149	1,548.140808	434.18
	4	85-125-161	1,638.320455	29,905.07
	5	72-106-137-168	1,692.149082	1,794,552.94
Butterfly	2	122	1,272.271778	4.06
	3	98-151	1,553.073405	256.90
	4	81-118-160	1,669.278327	14,275.86
	5	71-98-126-161	1,711.219320	703,576.72
Hunter	2	80	2,700.004083	5.57
	3	51-116	3,064.211202	551.01
	4	36-86-135	3,213.445979	41,822.80
	5	30-72-111-146	3,269.515980	2,841,715.66
Livingroom	2	107	1,249.745793	10.38
	3	88-146	1,626.644908	561.18
	4	76-124-163	1,758.852447	41,423.56
	5	57-98-133-169	1,827.458651	2,835,094.59
Road	2	97	966.3984012	8.40
	3	90-154	1,321.373927	530.75
	4	86-130-185	1,433.984405	41,270.26
	5	73-106-137-191	1,490.369804	2,866,724.13

Ateş böceği Algoritması, Parçacık Sürü Optimizasyonu ve Guguk kuşu Optimizasyonu Algoritması'yla test imgeleri üzerinde çoklu imge eşikleme gerçekleştirmek amacıyla PSO, AA ve GOA için sırasıyla Çizelge 5.2, 5.3 ve 5.4'de belirtilen parametreler kullanılmıştır. Performans kıyaslaması yapabilmek adına popülasyon sayısı, maksimum iterasyon sayısı üç metasezgisel algoritma için de aynı seçilmiştir.

Çoklu imge eşikleme gerçekleştirmek üzere her bir metasezgisel algoritma, 50 kez (tur) çalıştırılmıştır ve f_{ort} değerleri Denklem 5.1'de ve standart sapması S Denklem 5.2'de görüldüğü gibi hesaplanmıştır. Deneysel çalışmalarda gri seviye imgelere ait [0 255] arasındaki piksel değerleri [0 1] aralığına normalize edilmiştir.

Çizelge 5.8'de AA, GOA, PSO ile çoklu eşikleme sonucu elde edilen f_{ort} değerleri ile S standart sapmaları verilmiştir.

Çizelge 5.8. 50 tur sonunda elde edilen f_{ort} uygunluk değerleri ve S standart sapmaları (GOA, AA, PSO)

İmgeler	K	GOA		AA		PSO	
		f_{ort}	S	f_{ort}	S	f_{ort}	S
Baboon	2	1,219.050430	6.82E-13	1,219.050430	6.82E-13	1,219.050430	6.82E-13
	3	1,548.140808	1.82E-12	1,548.140808	1.82E-12	1,548.140808	1.82E-12
	4	1,638.320455	1.82E-12	1,638.320455	1.82E-12	1,638.320455	1.82E-12
	5	1,692.149082	2.27E-13	1,692.149082	2.27E-13	1,692.149082	2.27E-13
Butterfly	2	1,272.271778	9.09E-13	1,272.271778	9.09E-13	1,272.271778	9.09E-13
	3	1,553.073405	1.36E-12	1,553.054318	5.73E-02	1,553.073405	1.36E-12
	4	1,669.278327	1.14E-12	1,669.278327	1.14E-12	1,669.278327	1.14E-12
	5	1,711.219320	1.59E-12	1,711.219320	1.59E-12	1,711.219320	1.59E-12
Hunter	2	2,700.004083	1.36E-12	2,700.004083	1.36E-12	2,700.004083	1.36E-12
	3	3,064.211202	9.09E-13	3,064.211202	9.09E-13	3,064.211202	9.09E-13
	4	3,213.445979	2.73E-12	3,213.445979	2.73E-12	3,213.445979	2.73E-12
	5	3,269.511569	2.89E-03	3,269.513514	4.73E-03	3,269.511750	4.65E-03
Livingroom	2	1,249.745793	2.27E-13	1,249.745793	2.27E-13	1,249.745793	2.27E-13
	3	1,626.644908	4.55E-13	1,626.644908	4.55E-13	1,626.644908	4.55E-13
	4	1,758.852447	1.82E-12	1,758.852447	1.82E-12	1,758.852447	1.82E-12
	5	1,827.458651	1.59E-12	1,827.389352	1.90E-01	1,827.458651	1.59E-12
Road	2	966.3984012	6.82E-13	966.398401	6.82E-13	966.398401	6.82E-13
	3	1,321.373927	9.09E-13	1,321.373927	9.09E-13	1,321.373927	9.09E-13
	4	1,433.984405	6.82E-13	1,433.984405	6.82E-13	1,433.984405	6.82E-13
	5	1,490.369804	6.82E-13	1,490.369804	6.82E-13	1,490.369804	6.82E-13

Metasezgisel algoritmaların problemin çözümüne yakınsama sürelerini kıyaslayabilmek için her bir i. tur, o turda elde edilen f_i^* en iyi çözümün, amaç fonksiyonunun genel en iyi değeri olan f_{opt} değerine belirli bir ε hata miktarından küçük veya eşit bir değerde yaklaştığı an sonlandırılmıştır. Deneyde kullanılan ve her bir turu sonlandıran sonlandırma kriteri, Denklem 5.4'de ifade edilmektedir.

$$\left|f_i^* - f_{opt}\right| \leq \varepsilon = 10^{-3} \tag{5.4}$$

Çizelge 5.9'da GOA, AA ve PSO ile çoklu eşikleme gerçekleştirmek üzere algoritmaların 50 kez çalıştırılması sonrasında elde edilen t_i ve n_i değerlerinin ortalamaları verilmiştir.

Çizelge 5.9. 50 tur sonunda ortalama t_{ort} hesaplama süreleri ve n_{ort} iterasyon sayıları (GOA, AA, PSO)

İmgeler	**K**	**GOA**		**AA**		**PSO**	
		t_{ort} (ms)	n_{ort}	t_{ort} (ms)	n_{ort}	t_{ort} (ms)	n_{ort}
Baboon	2	8.22	1.20	16.37	1.02	3.38	2.12
	3	26.68	3.06	38.47	3.22	14.30	7.14
	4	60.51	5.88	104.41	7.92	24.80	12.10
	5	144.94	13.12	130.72	10.36	32.01	14.90
Butterfly	2	7.35	1.18	14.07	1.00	2.53	1.78
	3	21.10	2.54	54.81	12.52	9.41	5.36
	4	38.74	3.74	62.83	5.12	14.87	7.78
	5	80.90	7.40	138.68	10.02	21.42	10.60
Hunter	2	8.38	1.18	16.29	1.06	4.12	2.22
	3	30.04	3.28	47.41	3.80	17.16	7.44
	4	61.37	5.74	107.22	7.82	27.46	12.16
	5	823.69	74.30	384.15	42.00	161.32	66.54
Livingroom	2	7.61	1.10	15.21	1.00	2.60	1.70
	3	28.19	2.90	40.57	3.26	12.74	6.64
	4	50.31	4.80	77.06	6.12	19.72	9.62
	5	129.63	11.36	184.76	22.30	31.49	14.46
Road	2	8.77	1.26	15.87	1.02	3.19	2.06
	3	36.10	3.78	37.12	2.86	13.95	7.38
	4	81.86	7.62	84.89	5.98	23.52	11.48
	5	163.28	14.02	163.40	12.34	34.91	16.12

Çizelge 5.8'de görüldüğü üzere Hunter dışındaki tüm test imgelerinde PSO ve GOA'nın, ilgili test imgesi üzerinde aynı f_{ort} ve S değerlerine sahip olduğu görülmüştür ve bu f_{ort} değerleri Çizelge 5.7'de ifade edilen Otsu'nun yöntemi ile gerçekleştirilmiş etraflı arama sonucu elde edilen f_{opt} değerleriyle aynı olduğu görülmüştür.

Butterfly imgesi üzerinde 2 adet eşik değeri ile imge eşikleme ve Livingroom imgesi üzerinde 4 adet eşik değeri ile imge eşikleme gerçekleştirildiği anda PSO ve GOA'nın AA'ya göre daha iyi çözümlere sahip bir sonuç verdiği, daha stabil olduğu görülmüştür.

Hunter imgesi üzerinde 4 adet eşik değeri ile imge eşikleme gerçekleştirilmesi durumunda f_{opt} sonucuna en çok yaklaşan f_{ort} değerine sahip algoritmanın AA, daha sonrasında PSO olduğu görülmüştür. Ancak GOA'nın her bir turda birbirine daha yakın çözümler keşfettiği belirlenmiş ve diğer algoritmalara göre daha düşük bir f_{ort} elde etmesine rağmen stabilitesinden ödün vermediği görülmüştür.

Çizelge 5.9'da görüldüğü üzere en kısa sürede ε hatayla f_{opt} değerine ulaşan algoritma PSO ve ardından GOA'dır. AA'nın Hunter imgesi üzerinde 4 adet eşik değeri ile imge eşikleme gerçekleştirilmesi dışında GOA daha az iterasyonda ε hatayla f_{opt} değerine ulaşmıştır ancak PSO'nun bir iterasyonda harcadığı hesaplama zamanı daha kısa olmasından dolayı en kısa sürede yaklaşmaktadır. Hunter imgesi üzerinde 4 adet eşik değeri ile imge eşikleme gerçekleştirilmesi durumunda bazı turlarda algoritmalar ε hatayla f_{opt} değerine yaklaşamamışlar ve dolayısıyla 100 iterasyon sonunda turlarını sonlandırmışlardır. Bu sebeple iterasyon sayısı ve algoritma süresi artmıştır.

Bu uygulamanın son aşaması olarak seçilen test imgeleri üzerinde GOA, AA ve PSO ile 6 ve 7 adet eşik değeri seçilerek eşikleme gerçekleştirilmiş ve sonuçları Çizelge 5.10'da verilmiştir. Uygulamanın bu aşamasında sonlandırma kriteri olarak sadece 100 iterasyon seçilmiştir. Otsu'nun yöntemi ile 5 ve 6 adet eşik değeri seçilerek etraflı arama gerçekleştirmek çok uzun bir zaman almaktadır (Gao ve ark., 2010). Dolayısıyla uygulamada 5 ve 6 adet eşik değeri çözümleri etraflı arama ile hesaplanamamıştır ve f_{opt} değeri belirlenemediğinden belirli bir hatayla bu değere yaklaşmak mümkün olamamaktadır.

Çizelge 5.10. 50 tur sonunda ortalama f_{ort} hesaplama süreleri ve S standart sapmaları (GOA, AA, PSO)

İmgeler	K	GOA		AA		PSO	
		f_{ort}	S	f_{ort}	S	f_{ort}	S
Baboon	6	1717.890987	2.27E-13	1717.886420	1.69E-02	1717.890768	7.41E-04
	7	1735.634714	8.78E-03	1735.611150	1.08E-01	1735.630580	1.36E-02
Butterfly	6	1736.657190	4.55E-13	1736.505521	7.94E-01	1736.657190	4.55E-13
	7	1753.427493	2.27E-12	1753.232995	3.77E-01	1753.204412	4.30E-01
Hunter	6	3308.142295	2.27E-12	3308.008393	6.38E-01	3308.142295	2.27E-12
	7	3327.25082	1.82E-01	3327.112863	2.15E-01	3327.271614	1.76E-01
Livingroom	6	1870.577424	1.82E-12	1870.547343	1.52E-01	1870.577424	1.82E-12
	7	1896.582725	2.43E-02	1896.501712	1.76E-01	1896.566775	5.91E-02
Road	6	1520.828818	2.27E-13	1520.816704	4.28E-02	1520.828818	2.27E-13
	7	1538.374978	7.25E-01	1537.867367	1.26E+00	1538.170137	1.23E+00

Çizelge 5.10'da görüldüğü üzere 100 iterasyon sonunda AA'nın hiçbir aşamada, diğer algoritmalardan daha büyük bir f_{ort} ve/veya daha küçük bir S değeri elde edemediği görülmüştür. Bu durum, AA'nın diğer iki algoritmaya göre daha düşük bir stabilite ile daha az kaliteli çözümler üreten bir algoritma olduğunu göstermiştir.

Hunter imgesiyle 6 eşik değeri seçilerek gerçekleştirilen imge eşiklemede PSO, GOA ve AA'ya göre daha iyi f_{ort} değerine ve daha düşük S değerlerine sahiptir. Ancak Butterfly, Hunter, Livingroom, Road imgelerinin 5 adet eşik değeri ile eşikleme gerçekleştirilmiş uygulamalarında PSO ve GOA aynı f_{ort} değerlerine aynı S standart sapmalarla ulaşmıştır. Baboon, Butterfly, Livingroom ve Road imgelerinin 6 adet eşik değeri ile eşikleme gerçekleştirilmiş uygulamalarında ise GOA, PSO ve AA'ya göre daha büyük bir f_{ort} değerine daha küçük bir S standart sapmayla ulaşmıştır. Bu da GOA'nın diğer iki algoritmaya göre çoklu imge eşikleme üzerinde daha iyi bir sonuca stabil bir şekilde ulaştığının göstergesidir.

Bu çalışmada gerçekleştirilen iki uygulama sonucunda çoklu imge eşikleme probleminin çözümünde GOA'nın PSO ve AA'ya göre daha stabil ve daha kaliteli çözümlere sahip sonuçlar elde edildiği söylenebilir. Ancak PSO'nun yakınsama süresinin GOA ve AA'ya göre daha kısa sürmesi ile PSO'nun diğerlerine göre daha hızlı bir algoritma olduğu açıkça görülmüştür.

Çizelge 5.11'de, test imgeleri üzerinde 5 ve 6 adet eşik değeri seçilmesi ile gerçekleştirilen uygulama sonucu her bir algoritmanın her bir i. tur aşamasında elde

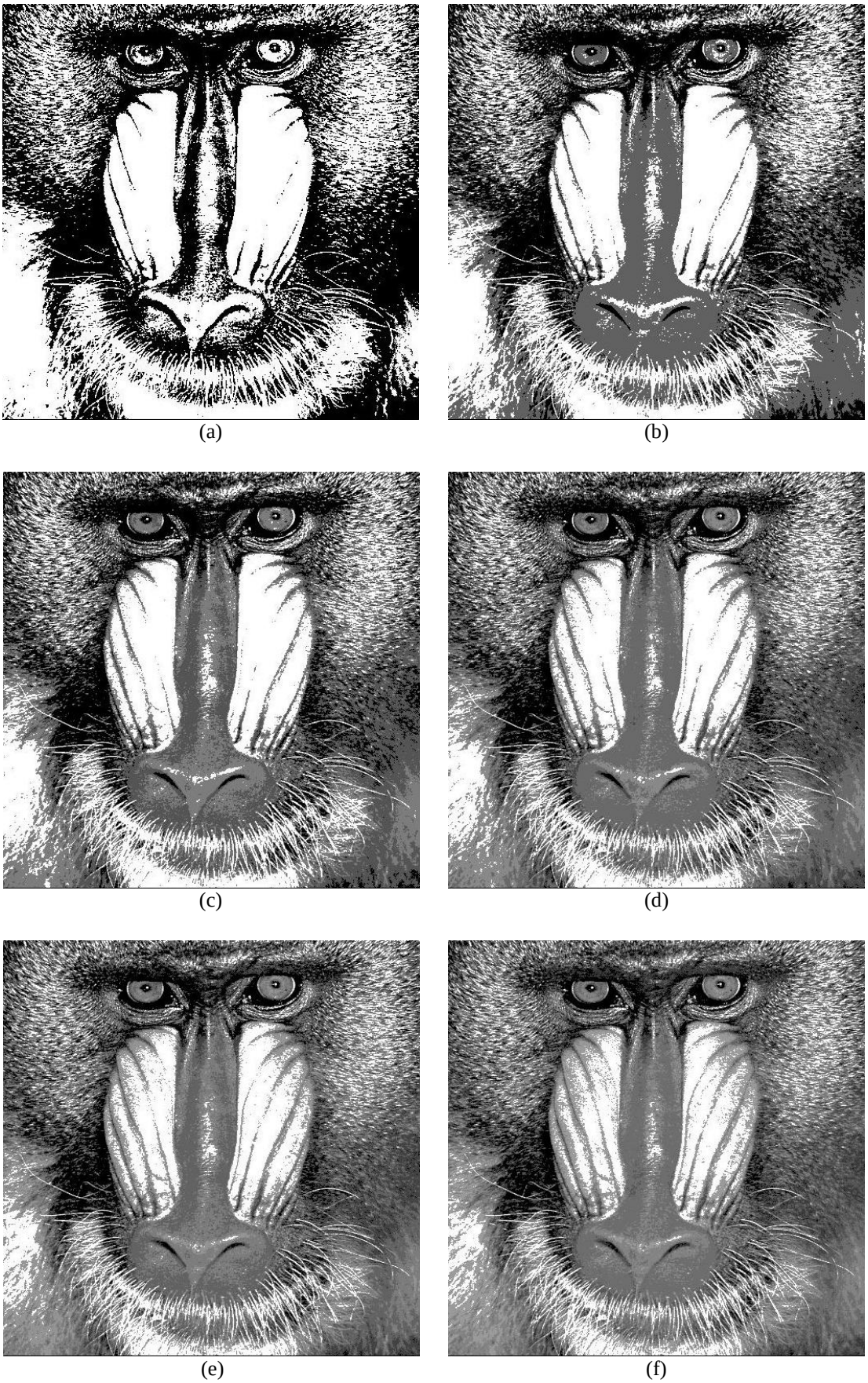
edilen en iyi f_i^* çözümlerinin uyguluk değeri maksimum olan f_{max}^* tur çözümünün belirlediği T_{max}^* eşik değerleri verilmiştir.

Çizelge 5.11. 50 tur arasındaki f_{max} çözümü ve belirlenen eşik değerleri (GOA, AA, PSO)

İmgeler	K	GOA		AA		PSO	
		f_{max}^*	T_{max}^*	f_{max}^*	T_{max}^*	f_{max}^*	T_{max}^*
Baboon	6	1717,890987	67-99-125-149-174	1717,890987	67-99-125-149-174	1717,890987	67-99-125-149-174
	7	1735,636932	59-86-110-132-155-178	1735,636932	59-86-110-132-155-178	1735,636932	59-86-110-132-155-178
Butterfly	6	1736,65719	71-98-125-153-180	1736,65719	71-99-125-153-180	1736,65719	71-99-125-154-179
	7	1753,427493	68-90-110-130-153-179	1753,427493	68-90-110-130-153-179	1753,427493	68-90-110-130-154-180
Hunter	6	3308,142295	22-53-88-122-152	3308,142295	22-53-88-122-152	3308,142295	22-53-88-122-152
	7	3327,403642	20-47-76-104-130-157	3327,403642	20-47-76-104-130-157	3327,403642	20-47-76-104-130-157
Livingroom	6	1870,577424	50-89-120-146-179	1870,577424	51-89-121-147-179	1870,577424	50-89-120-146-179
	7	1896,593514	42-76-105-129-151-181	1896,593514	42-76-105-128-151-181	1896,593514	43-77-105-129-151-180
Road	6	1520,828818	68-98-122-148-199	1520,828818	68-98-122-148-199	1520,828818	68-98-122-148-199
	7	1538,481883	63-89-111-130-156-204	1538,481883	63-89-111-130-156-204	1538,481883	63-89-111-130-156-204

Çizelge 5.11'de görüldüğü üzere her bir algoritma, aynı f_{max}^* değerine bir tur sonucunda muhakkak ulaşmıştır. Aynı f_{max}^* değeri karşısında elde edilen eşik değerleri çözümleri imgeyi aynı sınıflara bölütleyecek ve eşiklenmiş imge çıktıları aynı olacaktır. Butterfly ve Livingroom imgelerinde aynı f_{max}^* değeri sonucunda elde edilen eşik değerleri çözümlerinin bazıları arasında 1 gri seviye değeri kadar bir fark gözükmektedir. Ancak bu değerlerin farklılığı, seçilen gri seviye eşik değerinin imge içerisinde bulunmamasından kaynaklanmaktadır. Dolayısıyla her iki eşikleme çözümü de imgeyi aynı sınıflara ayırır.

Şekil 5.21-5.25'te Otsu'nun yöntemi ile gerçekleştirilen etraflı (detaylı) arama sonucunda 1'den 4 adete kadar belirlenen eşik değerlerini ve f_{max}^* değerleri sonucunda elde edilen 5 ve 6 adete kadar belirlenen eşik değeri çözümlerini kullanarak imgelerin bölütlenmesi sonucunda oluşan sınıflar, farklı gri seviye değerleriyle temsil edilmiştir.



(a) (b)

(c) (d)

(e) (f)

Şekil 5.21. Baboon imgesinin eşiklenmiş hali: (a) 2 gri seviyeli imge (b) 3 gri seviyeli imge (c) 4 gri seviyeli imge (d) 5 gri seviyeli imge (e) 6 gri seviyeli imge (f) 7 gri seviyeli imge



(a) (b)

(c) (d)

(e) (f)

Şekil 5.22. Butterfly imgesinin eşiklenmiş hali: (a) 2 gri seviyeli imge (b) 3 gri seviyeli imge (c) 4 gri seviyeli imge (d) 5 gri seviyeli imge (e) 6 gri seviyeli imge (f) 7 gri seviyeli imge



(a) (b)

(c) (d)

(e) (f)

Şekil 5.23. Hunter imgesinin eşiklenmiş hali: (a) 2 gri seviyeli imge (b) 3 gri seviyeli imge (c) 4 gri seviyeli imge (d) 5 gri seviyeli imge (e) 6 gri seviyeli imge (f) 7 gri seviyeli imge

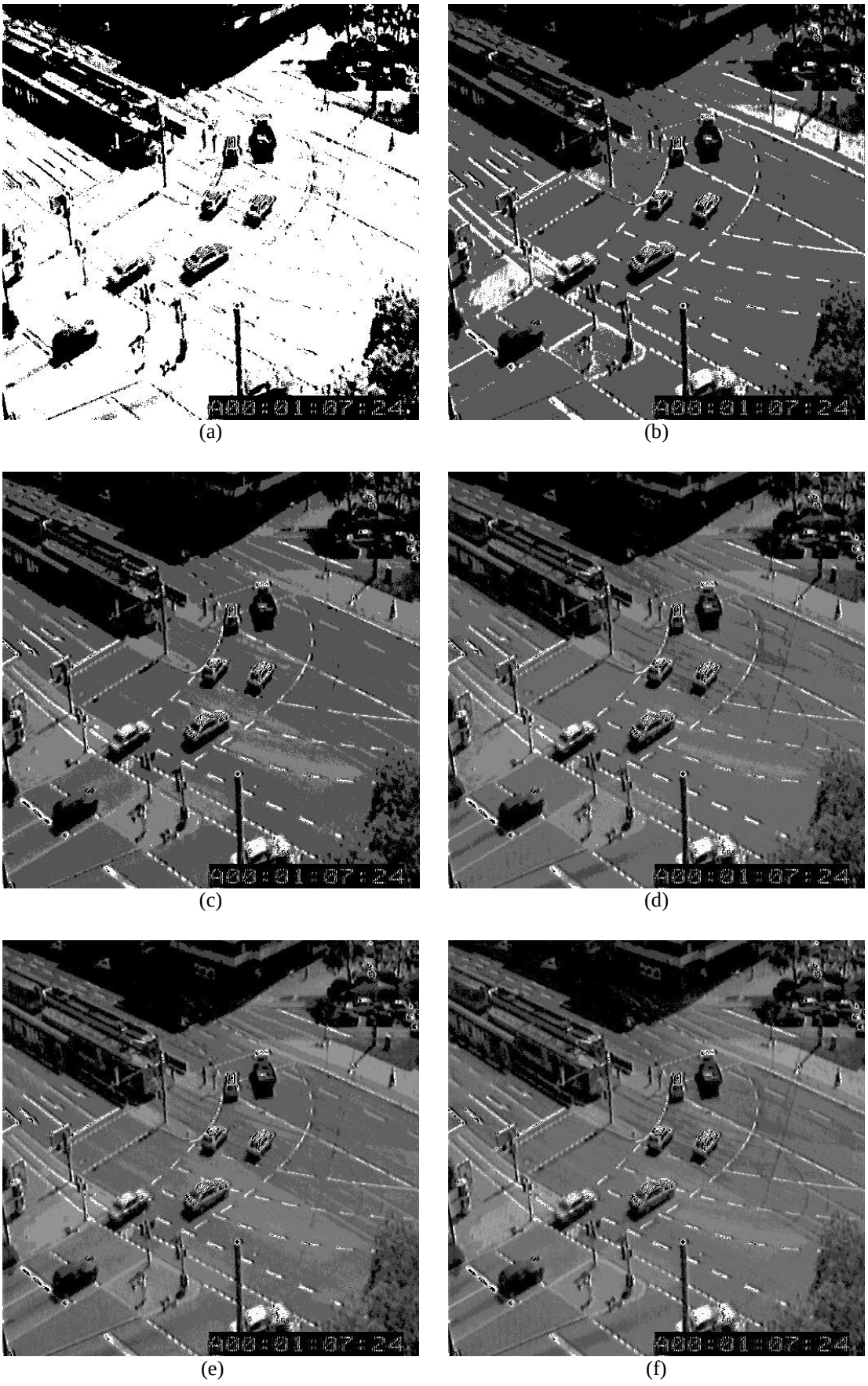


(a) (b)

(c) (d)

(e) (f)

Şekil 5.24. Livingroom imgesinin eşiklenmiş hali: (a) 2 gri seviyeli imge (b) 3 gri seviyeli imge (c) 4 gri seviyeli imge (d) 5 gri seviyeli imge (e) 6 gri seviyeli imge (f) 7 gri seviyeli imge



Şekil 5.25. Road imgesinin eşiklenmiş hali: (a) 2 gri seviyeli imge (b) 3 gri seviyeli imge (c) 4 gri seviyeli imge (d) 5 gri seviyeli imge (e) 6 gri seviyeli imge (f) 7 gri seviyeli imge

6. SONUÇLAR VE ÖNERİLER

6.1. Sonuçlar

İmge bölütleme, bir imgeyi birbiriyle örtüşmeyen alt imge gruplarına ayırma işlemidir. İmge eşikleme işlemi, imge bölütlemenin temel adımlardan biridir. Literatürde imge eşikleme amacıyla önerilmiş çok sayıda yöntem mevcuttur.

Tez çalışmasında, imge eşiklemede kullanılan yöntemlerden Otsu'nun yöntemi incelenmiş ve kullanılmıştır. Biri Otsu'nun amaç fonksiyonunu minimize (sınıf içi değişinti) eden ve diğeri de maksimize (sınıflar arası değişinti) eden iki farklı uygulama gerçekleştirilmiştir. Her iki uygulamada da test imgeleri üzerinde öncelikle Etraflı Arama tekniği kullanılmış ve optimum amaç fonksiyonu değerlerini veren eşik değerleri tespit edilmiştir. Test imgeleri üzerinde etraflı arama ile çoklu eşikleme gerçekleştirmede eşik sayısı arttıkça hesaplama zamanının hızla arttığı gözlenmiştir (Çizelge 5.1, Çizelge 5.7). Çalışmada etraflı arama ile 4 adet eşik değerinden sonra hesaplama zamanının aşırı artması ($\geq$2800 sn'den büyük bir değer) sebebiyle eşik değerleri hesaplanmamıştır. Bu durum, etraflı arama ile gerçek zamanlı uygulamalarda imge bölütleme gerçekleştirmenin mümkün olamayacağını göstermektedir.

Her iki uygulamada da, Otsu'nun yöntemi ile en iyi çözüme yakınsama aşamasında PSO, AA ve GOA olmak üzere 3 farklı metasezgisel teknik kullanılmıştır ve bu tekniklerin performans kıyaslaması çözüm kalitesi, stabilite ve yakınsama süresi bakımından gerçekleştirilmiştir. Her bir teknik için turlar, belirli bir hatayla etraflı arama ile elde edilen optimum uygunluk değerine yaklaştığı an sonlandırılmıştır. Optimal uygunluk değerlerine yaklaşılmadığı takdirde 100 iterasyon sonunda tur sonlandırılmıştır. Uygulama sonuçlarında da görüldüğü üzere metasezgisel algoritmalarla çoklu imge eşikleme gerçekleştirildiği takdirde hesaplama süresi etraflı aramaya göre büyük oranda düşmektedir (Örn. Lena imgesi için 4 eşik değeri seçiminde 1,743,617.17 ms'lik etraflı arama süresinin PSO ile 34.13 ms'ye düşmesi). PSO'nun AA ve GOA'ya göre daha kısa sürede en uygun değere yakınsadığı görülmüştür. GOA'nın en iyi çözüme yakınsama sürecinde harcadığı hesaplama süresinin, AA'ya göre daha kısa olduğu görülmüştür. Bu durum, metasezgisel algoritmaların; çoklu imge eşiklemenin gerçek zamanlı uygulamalarda kullanılabilirliğini, önemli derecede arttırmaktadır.

Deneysel sonuçlara göre, GOA genel olarak diğer iki algoritmaya kıyasla ya eşit ya da daha düşük standart sapmalara ve daha iyi uygunluk değerlerine ulaşmıştır. Bu da GOA'nın PSO ve AA'ya göre daha stabil bir algoritma olduğunu göstermektedir. AA, hem hesaplama süresi hem de en iyi sonuca yaklaşım açısından diğer iki algoritmaya göre daha düşük bir performans göstermiştir.

En uygun çözümün ne olduğu hakkında bilgi sahibi olmadan belirli bir iterasyon sayısında en iyi çözümü en düşük standart sapmayla üreten algoritmanın öncelikle GOA daha sonra PSO olduğu görülmüştür.

Tez çalışmasında kullanılan metasezgisel algoritmların çoklu imge eşikleme probleminde çözüm kalitesi, stabilite ve yakınsama süresi bakımından kullanılabilir olduğu sonucuna ulaşılmıştır.

6.2. Öneriler

İleriye yönelik yapılacak çalışmalarda AA'nın ve GOA'nın hesaplama süresini azaltmaya yönelik modifikasyonlar geliştirilerek algoritmanın etkinliği ve performansı arttırılabilir. Bu üç metasezgisel algoritma, teknik yazında önerilen diğer imge eşikleme yöntemleriyle de birleştirilerek imge eşikleme problemine uygulanabilir.

GOA'nın çözüm uzayına dağılmış olan çözümleri gruplandırma aşamasında K-Ortalama Kümeleme Algoritması yerine alternatif kümeleme algoritmaları kullanılarak GOA'nın performası incelenebilir.

Otsu yöntemi imgenin gri seviye değerlerini ölçüt alarak en uygun eşik değeri çözümlerini belirlemektedir. Teknik yazında Otsu'nun, hem imgenin gri seviye değerlerini hem de komşulukların ortalama gri seviye değerlerini temel alan iki boyuta genişletilmiş versiyonları mevcuttur ve gürültülü imgelerde daha iyi sonuçlar üretmektedir. İlerde gerçekleştirilebilecek çalışmalarda iki boyutlu Otsu'nun yöntemi, çoklu imge eşikleme problemine uygulanabilir.

KAYNAKLAR

Anonim, *Guguk kuşu hakkında ansiklopedik bilgi* [online], Türkiye, http://www.turkcebilgi.com/ansiklopedi/guguk_kuşu [Ziyaret Tarihi: 15 Ocak 2012].

Babu, B.G., Kannan, M., 2002, Lightning bugs, *Resonance*, 7 (9), 49-55.

Blum, C., Roli, A., 2003, Metaheuristics in combinatorial optimization: Overview and conceptual comparison, *ACM Computing Surveys-CSUR*, 35 (3), 268-308.

Chu, C.P., Lee, D.J., Tay, J.H., 2004, Bilevel thresholding of floc images, *Journal of Colloid and Interface Science*, 273 (2), 483–489.

Dong, L., Yu, G., Ogunbona, P., Li, W., 2008, An efficient iterative optimization algorithm for image thresholding, *Pattern Recognition Letters*, 29 (9), 1311–1316.

Eichmann, M., Lüssi, M., 2005, Efficient multilevel image thresholding, Dipl. Ing. FH Thesis, *Hochschule für Technik Rapperswil*, Switzerland, 1-53.

Fraga, H., 2008, Firefly luminescence: A historical perspective and recent developments, *Journal of Photochemical and Photobiological Sciences*, 7, 146-158.

Glover, F., 1986, Future paths for integer programming and links to artificial intelligence, *Computers and Operations Research*, 13 (5), 533–549.

Gonzales-Baron, U., Butler, F., 2006, A comparison of seven thresholding techniques with the k-means clustering algorithm for measurement of bread-crumb features by digital image analysis, *Journal of Food Engineering* 74 (2), 268–278.

Hammouche, K., Diaf, M., Siarry, P., 2010, A comparative study of various meta-heuristic techniques applied to the multilevel thresholding problem, *Engineering Applications of Artificial Intelligence*, 23 (5), 676-688.

Han, J., Kamber, M., 2006, Data mining concepts and techniques, *Morgan Kauffmann Publishers*, 500 Sansome St., S. 400, San Francisco, CA.

Hertz, A., Widmer, M., 2003, Guidelines for the use of meta-heuristics in combinatorial optimization, *European Journal of Operational Research*, 151 (2), 247-252.

Horng, M. H., Jiang, T. W., 2010, Multilevel image thresholding selection based on the firefly algorithm, *Symposia and Workshops on Ubiquitous, Autonomic and Trusted Computing*, 58-63.

Kapur, J.N., Sahoo P.K., Wong, A.K.C., 1985, A new method for gray-level picture thresholding using the entropy of the histogram, *Signal Processing*, 29 (3), 273-285.

Kennedy, J., Eberhart, R., 1995, Particle swarm optimization, *Proceedings of IEEE International Conference on Neural Networks*, Piscataway, NJ, 1942-1948.

Kittler, J., Illingworth, J., 1986, Minimum error thresholding, *Pattern Recognition*, 19 (1), 41–47.

Laporte, G., Gendreau, M., Potvin, J. Y., Semet, F., 2000, Classical and modern heuristics for the vehicle routing problem, *International Transactions in Operation Research*, 7, 285-300.

Levine, M. D., Nazif, A. M., 1985, Dynamic measurement of computer generated image segmentations, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7 (2), 155-164.

Li, X., Zhao, Z., Cheng, H.D., 1995, Fuzzy entropy threshold approach to breast cancer detection, *Information Sciences*, 4 (1), 49–56.

Liang, Y. C., Yin, Y. C., 2011, Optimal multilevel thresholding using a hybrid ant colony system, *Journal of the Chinese Institute of Industrial Engineers*, 28 (1), 20-33.

Liao, P.S., Chen, T.S., Chung, P.C., 2001, A fast algorithm for multilevel thresholding, *Journal of Information Science and Engineering*, 17, 713–727.

Likas, A., Vlassis, N., Verbeek, J., 2003, The global k-means clustering algorithm, *Pattern Recognition*, 36, 451 – 461.

Lin, K.C., 2001, Fast thresholding computation by searching for zero derivates of images between-class variance, *27th Annual Conference on IEEE Industrial Electronics Society-IECON 2001*, Denver, Colorado, USA, 393–397.

Liu, D., Yu, J., 2009, Otsu method and k-means, *9th International Conference on Hybrid Intelligent Systems – HIS 2009*, Shenyang, Liaoning, China 344-349.

Lukasik, S., Zak, S., 2009, Firefly algorithm for continuous constrained optimization tasks, *1th International Conference on Computational Collective Intelligence, Semantic Web, Social Networks and Multiagent Systems*, Wrodaw, 97-106.

Luo, X., Tian, J., 2000, Multi-level thresholding: maximum entropy approach using ICM, *Proceedings of the 15th International Conference on Pattern Recognition*, 3, Barcelona, Spain, 778–781.

MacQueen, J.B., 1967, Some methods for classification and analysis of multivariate observations, *5th Berkeley Symposium on Mathematical Statistics and Probability*, University of California, Berkeley, CA, 281–297.

Maitra, M., Chatterjee, A., 2008, A hybrid cooperative-comprehensive learning based PSO algorithm for image segmentation using multilevel thresholding, *Expert Systems with Applications*, 34 (2), 1341–1350.

Otsu, N., 1979, A threshold selection method from gray-level histograms, *IEEE Transactions on Systems, Man and Cybernetics*, 9 (1), 62-66.

Pal, N.R., Pal, S.K., 1993, A review on image segmentation techniques, *Pattern Recognition*, 26 (9), 1277-1294.

Pang-Ning Tan, P. N., Steinbach, M., and Kumar, V., (2006), Introduction to data mining, *Addison-Wesley Longman Publishing*, Boston, MA, USA.

Payne, R.B. Robert, Sorenson, M.D., 2005, The cuckoos, *Oxford University Press*, Oxford, UK.

Pun, T., 1980, A new method for gray-level picture thresholding using the entropy of the histogram, *Signal Processing*, 2 (3), 223-237.

Rajabioun, R., 2011, Cuckoo optimization algorithm, *Applied Soft Computing*, 11 (8), 5508-5518.

Rueda, L., 2008, An efficient algorithm for optimal multilevel thresholding of irregularly sampled histograms, *International Workshops on Structural and Syntactic Pattern Recognition and Statistical Techniques in Pattern Recognition*, Orlando, Florida, USA, 602-611.

Sathya, P.D., Kayalvizhi, R., 2010, A New multilevel thresholding method using adaptive particle swarm optimization for image segmentation, *International Journal of Electronics and Communication Engineering*, 3 (3), 113-124.

Sathya, P.D., Kayalvizhi, R., 2011, Modified bacterial foraging algorithm based multilevel thresholding for image segmentation, *Engineering Applications of Artificial Intelligence*, 24 (4), 595-615.

Sezgin, M., 2002, İmge eşikleme yöntemlerinin başarım değerlendirmesi ve tahribatsız muayenede kullanımı, Doktora Tezi, *İTÜ Fen Bilimleri Enstitüsü*, İstanbul, 8-108.

Sezgin, M., Sankur, B., 2004, Survey over image thresholding techniques and quantitative performance evaluation, *Journal of Electronic Imaging*, 13 (1), 146-165.

Sezgin, M., Tasaltin, R., 2000, A new dichotomization technique to multilevel thresholding devoted to inspection applications, *Pattern Recognition Letters*, 21 (2), 151–161.

Shi, Y. H., Eberhart, R. C., 1998, A modified particle swarm optimizer, *IEEE International Conference on Evolutionary Computation*, Anchorage, Alaska, 69-73.

Şengür, A., Türkoğlu, İ., İnce, M., 2005, İki boyutlu entropi ile görüntü eşikleme uygulamaları, *IEEE 13. Sinyal İşleme ve İletişim Uygulamaları Kurultayı*, 131 (2), Kayseri, Türkiye, 275-278.

Tamer, S., Karakuzu, C., 2006, Parçacık sürü optimizasyon algoritması ve benzetim örnekleri, *Elektrik-Elektronik ve Bilgisayar Sempozyumu-ELECO 2006, Elektronik Bildirileri Kitabı*, Bursa, 302-306.

Tarantilis, C. D., Ioannou, G., Prastacos, G., 2005, Advanced vehicle routing algorithms for complex operations management problems, *Journal of Food Engineering*, 70 (3), 455-471.

Tsai, D.M., 1995, A fast thresholding selection procedure for multimodal and unimodal histograms, *Pattern Recognition Letters*, 16 (6), 653-666.

Ünver, Ö., Gamgam, H., Altunkaynak, B., 2011, Temel istatistiksel yöntemler, *Seçkin Yayınları*, Ankara, Türkiye, 38-40.

Wei, C., Kangling, F., 2008, Multilevel thresholding algorithm based on particle swarm optimization for image segmentation, *27th Chinese Control Conference*, Kunming, Yunnan, China 348-351.

Yang, X.S., 2008, Nature-inspired metaheuristic algorithms, *Luniver Press*, Frome, BA11 6TT, United Kingdom.

Yang, X.S., 2009, Firefly algorithms for multimodal optimization, *Proceedings of the 5th International Conference on Stochastic Algorithms: Foundations and Applications-SAGA 2009*,Sapporo, Japan, 169-178.

Yang, X.S., 2010, Firefly algorithm stochastic test functions and design optimization, *International Journal of Bio-Inspired Computation*, 2 (2), 78-84.

Yang, Z. H., Pu, Z. B., Qi, Z. Q., 2003, Relative entropy multilevel thresholding method based on genetic algorithm, *IEEE International Conference on Neural Networks and Signal Processing*, 1, Nanjing, China, 583-586.

Yen, J.C., Chang, F.J., Chang, S., 1995, A new criterion for automatic multilevel thresholding, *IEEE Transactions on Image Processing*, 4 (3), 370–378.

Yin, P. Y., Chen, L. H., 1997, A fast iterative scheme for multilevel thresholding methods, *Signal Processing*, 60 (3), 305-313

Zhiwei, Y., Zhaobao, Z., Xin, Y., Xiaogang, N., 2006, Automatic threshold selection based on ant colony optimization algorithm. *Proceedings of the International Conference on Neural Networks and Brain*, Beijing, China, 728–732.

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı	**:**	Gökhan PEKDEMİR
Uyruğu	**:**	T.C.
Doğum Yeri ve Tarihi	**:**	Antalya – 16.11.1987
Telefon	**:**	05058039731
Faks	**:**	
e-mail	**:**	gpekdemir@sgk.gov.tr

EĞİTİM

Derece		**Adı, İlçe, İl**	**Bitirme Yılı**
Lise	:	**Adem Tolunay Anadolu Lisesi, Merkez, Antalya**	**2005**
Üniversite	:	**Selçuk Üniversitesi, M.M.F., Bilgisayar Müh., Selçuklu, Konya**	**2009**
Yüksek Lisans	:	**Selçuk Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar Müh., A.B.D., Selçuklu, Konya**	
Doktora	:		

İŞ DENEYİMLERİ

Yıl	**Kurum**	**Görevi**
2009	**POLMOT MOT. VE MAK. SAN. TİC. LTD. ŞTİ.**	**Bilgi İşlem Sorumlusu**
2010	**KOMBASSAN MERAM MERMER A.Ş.**	**Yazılım Uzmanı**
2011	**T.C. SOSYAL GÜVENLİK KURUMU BİLGİ SİS. VE GÜV. DAİRE BŞK.**	**Programcı (Network Birimi)**

UZMANLIK ALANI

Cisco Ağ Teknolojileri, Network Programlama, Ağ Yönetimi

YABANCI DİLLER

İngilizce **Okuma:** İyi **Yazma:** İyi **Konuşma:** İyi

BELİRTMEK İSTEĞİNİZ DİĞER ÖZELLİKLER

Alınan Eğitimler:

- Cisco CCNA (ICND 1-2)
- Enterasys Switching & Routing
- F5 Load Balancing
- McAfee ePolicy Orchestrator