



T.C.

SELÇUK ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

SÜRÜ ZEKÂSI KULLANARAK RENKLİ GÖRÜNTÜ SEGMENTASYON TEKNİKLERİNİN GELİŞTİRİLMESİ

Tahir SAĞ

DOKTORA TEZİ

Bilgisayar Mühendisliği Anabilim Dalı

Haziran-2015
KONYA
Her Hakkı Saklıdır

ÖZET

DOKTORA TEZİ

SÜRÜ ZEKÂSI KULLANARAK RENKLİ GÖRÜNTÜ SEGMENTASYON TEKNİKLERİNİN GELİŞTİRİLMESİ

Tahir SAĞ

Selçuk Üniversitesi Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Doç. Dr. Mehmet ÇUNKAŞ

2015, 160 Sayfa

Jüri
Danışman: Doç. Dr. Mehmet ÇUNKAŞ
Prof. Dr. Bekir KARLIK
Doç. Dr. Harun UĞUZ
Yrd. Doç. Dr. Ömer Kaan BAYKAN
Yrd. Doç. Dr. Mehmet HACIBEYOĞLU

Bu çalışmada sürü zekâsı teknikleri kullanılarak görüntü segmentasyonu yapılmıştır. İlk aşamada; tek amaçlı temel Yapay Arı Kolonisi Optimizasyon Algoritması kümeleme yoluyla görüntü segmentasyonuna uyarlanmıştır. Uygunluk fonksiyonu olarak niceleme hatası kullanılmıştır. Elde edilen sonuçlar üzerinden kümelemenin başarısını anlamak amacıyla d_{min} ve d_{max} olarak adlandırılan iki ölçüt değeri hesaplanmıştır. Sonuçlar K-Ortalamalar, Bulanık C-Ortalamalar algoritmaları ve Parçacık Sürü Optimizasyonu temelli bir kümeleme algoritması ile karşılaştırılmıştır. Önerilen yaklaşımın tüm gösterge değerleri bakımından daha başarılı sonuçlar elde ettiği görülmüştür.

İkinci aşamada; çok amaçlı optimizasyon problemlerinin çözümü için Yapay Arı Kolonisi üzerinde durularak yeni hibrid bir algoritma geliştirilmiştir. Önerilen algoritma Çok Amaçlı Optimizasyon için İyileştirilmiş Arı Kolonisi Algoritması (IBMO) olarak isimlendirilmiştir. IBMO'nun performansı literatürdeki en iyi dört sezgisel algoritma ile karşılaştırılmıştır. Bunlar sırasıyla: Bastırılmamışlık Sıralamalı Genetik Algoritma II (NSGA2), Kuvvetli Evrimsel Pareto Algoritması-2 (SPEA2), Optimal Çok-Amaçlı Parçacık Sürü Optimizasyonu (OMOPSO) ve Çok-Amaçlı Optimizasyon için Arşiv temelli Hibrid Dağılma Araması (AbYSS) algoritmalarıdır. Çalışmada karşılaştırmaların doğruluğunu ortaya koymak açısından 26 farklı test problemi kullanılmıştır. Performans değerlendirmesi amacıyla sonuçlar dört farklı performans göstergesi (Genel Uzaklık, Yayılma, Maksimum Yayılma, Hiper-hacim) kullanılmıştır. IBMO, test problemlerinin büyük çoğunluğunda istatistiksel güven aralığında kalarak en iyi değerlere ulaşmayı başarmıştır.

Son aşamada çok-amaçlı optimizasyona dayanan yeni bir renkli görüntü segmentasyon algoritması önerilmiştir. Segmentasyon işlemi, renkli görüntüden çıkarılan özniteliklerin gruplanması yoluyla gerçekleştirilen çok-amaçlı bir kümeleme problemi olarak modellenmiştir. Önerilen yöntem çok-amaçlı optimizasyon algoritması IBMO ve çekirdekli bölge genişletme tekniğine dayanmaktadır. Çekirdek noktalarının optimal koordinatlarını atanması, optimal benzerlik eşik değerlerini belirlenmesi ve segmentasyon kalitesini birden fazla küme doğrulama göstergesi ile değerlendirilmesini sağlamak amacıyla IBMO renkli görüntü segmentasyonuna uyarlanmıştır. Önerilen algoritma Berkeley görüntü segmentasyon veri kümesindeki test görüntüleri üzerinde uygulanmıştır. Elde edilen segmentasyon sonuçlarının bölge-tabanlı tutarlılık göstergeleri hesaplanmıştır. Ayrıca sonuçlar NSGA-II, Bastırılmamışlık Sıralamalı Parçacık Sürü Optimizasyonu (NSPSO) ve Bulanık C-Ortalamalar (FCM) algoritmalarından aynı şartlar altında alınan segmentasyon sonuçlarıyla karşılaştırılmıştır. Elde edilen başarılı sonuçlar önerilen yöntemin genel amaçlı renkli görüntü segmentasyonu için tutarlı bir doğruluğa sahip olduğunu göstermiştir.

Anahtar Kelimeler: Çok-amaçlı optimizasyon, renkli görüntü segmentasyonu, sürü zekâsı, Yapay Arı Kolonisi

ABSTRACT

Ph.D THESIS

DEVELOPMENT OF COLOR IMAGE SEGMENTATION TECHNIQUES USING SWARM INTELLIGENCE

Tahir SAĞ

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

Advisor: Assoc. Prof. Dr. Mehmet ÇUNKAŞ

2015, 160 Pages

Jury
Advisor: Assoc. Prof. Dr. Mehmet ÇUNKAŞ
Prof. Dr. Bekir KARLIK
Assoc. Prof. Dr. Harun UĞUZ
Assist. Prof. Dr. Ömer Kaan BAYKAN
Assist. Prof. Dr. Mehmet HACIBEYOĞLU

In this dissertation, image segmentation is done by using swarm intelligence techniques. First, Original single-objective Artificial Bee Colony Optimization Algorithm is applied to image segmentation through clustering process. A quantization error is used as the objective function. In addition to this indicator, two other metrics are calculated for the obtained values in order to ensure the quality of results. The results are compared with K-Means, Fuzzy C-Means and a Particle Swarm Optimization based clustering algorithm. It can be reported that the proposed ABC-based algorithm has better successful results than the other three algorithms.

In the second step of the study, a new hybrid algorithm for multi-objective optimization problems is developed by concentrating on Artificial Bee Colony algorithm. The proposed algorithm is called as Improved Bee Colony Optimization for Multiobjective Optimization (IBMO). The performance of IBMO is compared with the state-of-the-art algorithms: Nondominated Sorting Genetic Algorithm-II (NSGA2), The Strength Pareto Evolutionary Algorithm-2 (SPEA2), e Strength Pareto Evolutionary Algorithm (OMOPSO) and Archive-Based Hybrid Scatter Search to Multiobjective Optimization (AbYSS). In order to demonstrate the accuracy of the obtained results, IBMO and the other algorithms are applied to 26 benchmark functions. Four different quality indicators are used. These are General Distance, Spread, Maximum Spread and Hypervolume. By satisfying the statistical confidence level, IBMO have achieved to the best results for almost all test functions and for all quality indicators.

Finally, a novel image segmentation algorithm based on multi-objective optimization for color images is proposed. Image segmentation is modeled as a multiobjective clustering problem trough grouping features extracted from a color image. The proposed segmentation method is depend on IBMO and seeded region growing technique. IBMO determines the optimal coordinates of the seed points and similarity difference of each region by optimizing a set of cluster validity indices simultaneously in order to improve the quality of segmentation. The proposed method was applied on several natural images obtained from Berkeley segmentation database. The region-based consistency errors of the obtained results are calculated. Furthermore, the results are compared with segmentation results that are obtained from NSGA2, Nondominated Sorting Particle Swarm Optimization (NSPSO) and Fuzzy C-Means algorithms in the same conditions. The robustness of the proposed ideas was proven by comparison of hand-labeled and experimentally obtained segmentation results.

Keywords: Multiobjective optimization, color image segmentation, Artificial Bee Colony optimization, swarm intelligence

ÖNSÖZ

Çalışmamın her aşamasında yardımlarını esirgemeyen ve pozitif eleştirileri ile çalışmaya yön veren değerli danışman hocam Doç. Dr. Mehmet ÇUNKAŞ'a ve fikirleriyle bana yol gösteren tez izleme komitesi üyeleri Prof. Dr. Bekir KARLIK'a ve Yrd. Doç. Dr. Ömer Kaan BAYKAN'a sonsuz teşekkürlerimi sunarım.

Bunun yanında akademik çalışmalarımı ve bu çalışma ile ilgili yurtdışında sunduğum bildirileri ve tez projelerimi destekleyen Selçuk Üniversitesi Bilimsel Araştırma Projeleri Koordinatörlüğüne (BAP) teşekkür ederim.

Doktora tez çalışmam sürecinde TÜBİTAK Bilim İnsanı Destekleme Daire Başkanlığının (BİDEB) yurtiçi lisansüstü burs programı kapsamında eğitimime sağladıkları maddi katkı ve motivasyondan dolayı TÜBİTAK'a teşekkür ederim.

Rahmetli babamın eksikliğini hissettirmemek için ömrünü bizi yetiştirmeye adayan annem Elmas SAĞ başta olmak üzere; bu meşakkatli çalışmanın başından sonuna kadar bana her türlü sabrı, fedakârlığı ve hoşgörüyü gösteren ve her türlü sıkıntıyı benimle paylaşan abim Mehmet Akif SAĞ'a, ablam Neslihan SERDAROĞLU SAĞ'a, kardeşim Fatma SAĞ'a ve sevgili eşim Nagihan BİLİR SAĞ'a en derin şükran duygularımı belirtmek istiyorum.

Tahir SAĞ
KONYA-2015

İÇİNDEKİLER

TEZ BİLDİRİMİ ..... **iii**

ÖZET ..... **iv**

ABSTRACT ..... **v**

ÖNSÖZ ..... **vi**

İÇİNDEKİLER ..... **vii**

ŞEKİL LİSTESİ ..... **x**

ÇİZELGE LİSTESİ ..... **xiv**

SİMGELER VE KISALTMALAR ..... **xviii**

1. GİRİŞ **1**

1.1. Çalışmanın Amacı ve Önemi 3

1.2. Kaynak Araştırması 5

1.3. Tezin Organizasyonu 10

2. SÜRÜ ZEKÂSI **12**

2.1. Yapay Arı Kolonisi 14

2.2. Parçacık Sürü Optimizasyonu 18

2.3. Bölüm Değerlendirmesi 21

3. GÖRÜNTÜ SEGMENTASYONU **22**

3.1. Renk Uzayları 23

3.1.1. RGB Renk Uzayı 24

3.1.2. CIE Renk Uzayı Ailesi 25

3.1.3. HSI Renk Uzayı Ailesi 26

3.1.4. YUV Renk Uzayı Ailesi 27

3.2. Segmentasyon Yöntemlerinin Sınıflandırılması 28

3.2.1. Eşikleme-Tabanlı Segmentasyon 29

3.2.2. Kenar-Tabanlı Segmentasyon 30

3.2.3. Bölge-Tabanlı Segmentasyon 31

3.2.4. Kümeleme-Tabanlı Segmentasyon 35

3.2.5. Diğer Segmentasyon Yaklaşımları 37

4. ABC VE PSO İLE KÜMELEME TABANLI GÖRÜNTÜ SEGMENTASYONU 39

4.1. PSO Temelli Kümeleme Algoritması 39

4.2. ABC Temelli Kümeleme Algoritması 39

4.3. Değerlendirme Ölçütleri 41

4.4. Deneysel Sonuçlar 42

5. ÇOK AMAÇLI OPTİMİZASYON VE SEGMENTASYON 46

5.1. Pareto-Optimal Kavramı 48

5.2. Çok Amaçlı Optimizasyon Problemleri 49

5.3. Bastırılmamışlık Sıralamalı Genetik Algoritma II (NSGA-II) 50

5.3.1. Hızlı bastırılmamış sınıflandırma yaklaşımı 51

5.3.2. Çeşitliliğin sağlanması 52

5.3.3. Ana döngü 54

5.4. Bastırılmamışlık Sıralamalı Parçacık Sürü Optimizasyonu (NSPSO) 55

5.5. Bölüm Değerlendirmesi 56

6. ABC TEMELLİ ÇOK AMAÇLI OPTİMİZASYON ALGORİTMASI 59

6.1. İyileştirilmiş Arı Kolonisi Optimizasyon Algoritması (IBMO) 59

6.2. Deneysel Sonuçlar 67

6.2.1. Test Problemleri 67

6.2.2. Performans Göstergeleri 68

6.3. Parametre Ayarları 71

6.4. Gösterge Sonuçları ve Tartışma 72

6.5. Bölüm Değerlendirmesi 74

7. ÇOK AMAÇLI OPTİMİZASYON İLE RENKLİ GÖRÜNTÜ SEGMENTASYONU 85

7.1. Önerilen Renkli Segmentasyon Yöntemi 85

7.2. Öznitelik Çıkarımı 90

7.2.1. Kullanılan Renk Uzayları 91

7.2.2. Gabor Filtresi 91

7.2.3. Yerel Homojenlik 93

7.3. Değerlendirme Ölçütleri ve Deneysel Sonuçlar 94

7.3.1. Değerlendirme Ölçütleri 95
7.3.2. Parametre Seçimi 97
7.3.3. Deneysel Sonuçlar 100
7.4. Bölüm Değerlendirmesi 103
8. SONUÇLAR 107
KAYNAKLAR 110
EKLER 120
EK-1 Çok-Amaçlı Test Problemlerinin Tanımlamaları 120
EK-2 50.000 Fonksiyon Değerlendirmesi için Hesaplanan Performans Göstergeleri 125
EK-3 Geliştirilen Renkli Segmentasyon Yöntemi ile Test Görüntüleri için Alınan Ön-Çalıştırma Sonuçları 130
EK-4 Algoritmaların 500 Fonksiyon Değerlendirmesi için 10 Bağımsız Çalıştırma Sonunda Aldıkları Segmentasyon Sonuçları 134
ÖZGEÇMİŞ 158

ŞEKİL LİSTESİ

Şekil 2.1. ABC algoritmasının temel adımları 14

Şekil 2.2. PSO algoritmasının sözde-kodu 20

Şekil 3.1. 3-boyutlu bir küp içinde RGB renk uzayı gösterimi (Cheng ve ark. 2001).... 24

Şekil 3.2. HSI renk uzayı (Cheng ve ark., 2001)..... 26

Şekil 3.3. Gri-seviye histogramı 30

Şekil 3.4. Dörtlü ağaç yapısı (Gonzales ve Woods, 2002)..... 34

Şekil 4.1. Omran'ın PSO temelli kümeleme algoritması 40

Şekil 4.2. (a) Sentetik, (b) MRI insan beyni ve (c) Uydu görüntüsü..... 42

Şekil 4.3. Segmentasyon sonuçları 44

Şekil 4.4. Sentetik test görüntüsü için ortalama Je değerleri 44

Şekil 4.5. MRI test görüntüsü için ortalama Je değerleri..... 44

Şekil 4.6. Uydu test görüntüsü için ortalama Je değerleri..... 45

Şekil 4.7. Sentetik test görüntüsü için ortalama Je değerinden sapmaları 45

Şekil 4.8. MRI test görüntüsü için ortalama Je değerinden sapmaları..... 45

Şekil 4.9. Uydu test görüntüsü için ortalama Je değerinden sapmaları..... 45

Şekil 5.1. Görüntü segmentasyonunun girişten çıkışa kadar çok kriterli yapısı..... 47

Şekil 5.2. Bastırılmamış çözümler (kapalı çemberler) ve bastırılmış çözümler (açık çemberler)..... 50

Şekil 5.3. Hızlı bastırılmamış çözüm sıralama metodu için sözde kod (Deb, 2002).... 51

Şekil 5.4. Kalabalıklık mesafe hesaplaması 53

Şekil 5.5. Kalabalıklık karşılaştırma mesafesi için sözde kod (Deb ve ark., 2002) 53

Şekil 5.6. NSGA-II bastırılmamışlık sıralaması (Deb, 2002)..... 54

Şekil 5.7. NSGA-II ana prosedürü (Deb, 2002) 55

Şekil 5.8. NSPSO algoritmasının sözde kodu 57

Şekil 6.1. IBMO algoritmasının sözde-kodu..... 60

Şekil 6.2. IBMO algoritmasının akış diyagramı 61

Şekil 6.3. Çeşitlendirme metodu 61

Şekil 6.4. Değiştirilecek parametrenin belirlenmesinin kaba kodu 63

Şekil 6.5. İyileştirme metodunun sözde kodu 64

Şekil 6.6. 3-boyutlu bir hypervolume 70

Şekil 6.7. 2-amaçlı test problemlerinin IBMO ile bulunan Pareto yüzeyleri 80

Şekil 6.8. DTLZ1 problemi için IBMO, AbySS, OMOPSO, NSGAII, SPEA2 ile elde edilmiş bastırılmamış çözümler ve doğru Pareto-Optimal yüzey .. 81

Şekil 6.9. DTLZ3 problemi için IBMO, AbySS, OMOPSO, NSGAII, SPEA2 ile elde edilmiş bastırılmamış çözümler ve doğru Pareto-Optimal yüzey .. 82

Şekil 6.10. DTLZ4 problemi için IBMO, AbySS, OMOPSO, NSGAII, SPEA2 ile elde edilmiş bastırılmamış çözümler ve doğru Pareto-Optimal yüzey .. 83

Şekil 6.11. DTLZ7 problemi için IBMO, AbySS, OMOPSO, NSGAII, SPEA2 ile elde edilmiş bastırılmamış çözümler ve doğru Pareto-Optimal yüzey .. 84

Şekil 7.1. Önerilen segmentasyon metodunun çalışmasını açıklayan ana hatları .. 86

Şekil 7.2. IBMO algoritmasında oluşturulan bir aday çözüm gösterimi 88

Şekil 7.3. i, j pozisyonundaki bir piksel için yerel görüntü penceresi 93

Şekil 7.4. S1 ve S2 segmentasyonları için hata matrisi 96

Şekil 7.5. Tüm pikseller için LRE hesaplamasını gösteren sözde-kod - $LRES1, S2$ 96

Şekil 7.6. Önerilen segmentasyon metodunun akış diyagramı 98

Şekil 7.7. Toplam tutarlılık hatalarının 10 çalışma için kutu diyagramları 104

Şekil 7.8. Test görüntüleri ve denekler tarafından oluşturulan el-çizimi segmentasyon sonuçları 105

Şekil 7.9. Algoritmaların bulduğu segmentasyon sonuçları 106

Şekil Ek-4.1. 3096 numaralı test görüntüsü için 500 FEs sonunda IBMO ile alınan 10 çalıştırma sonucu 144

Şekil Ek-4.2. 3096 numaralı test görüntüsü için 500 FEs sonunda NSPSO ile alınan 10 çalıştırma sonucu 144

Şekil Ek-4.3. 3096 numaralı test görüntüsü için 500 FEs sonunda NSGA2 ile alınan 10 çalıştırma sonucu 145

Şekil Ek-4.4. 3096 numaralı test görüntüsü için 500 FEs sonunda FCM ile alınan 10 çalıştırma sonucu 145

Şekil Ek-4.5. 42049 numaralı test görüntüsü için 500 FEs sonunda IBMO ile alınan 10 çalıştırma sonucu 146

Şekil Ek-4.6. 42049 numaralı test görüntüsü için 500 FEs sonunda NSPSO ile alınan 10 çalıştırma sonucu 146

Şekil Ek-4.7. 42049 numaralı test görüntüsü için 500 FEs sonunda NSGA2 ile alınan 10 çalıştırma sonucu 147

Şekil Ek-4.8. 42049 numaralı test görüntüsü için 500 FEs sonunda FCM ile alınan 10 çalıştırma sonucu 147

Şekil Ek-4.9. 62096 numaralı test görüntüsü için 500 FEs sonunda IBMO ile alınan 10 çalıştırma sonucu 148

Şekil Ek-4.9. 62096 numaralı test görüntüsü için 500 FEs sonunda NSPSO ile alınan 10 çalıştırma sonucu 148

Şekil Ek-4.10. 62096 numaralı test görüntüsü için 500 FEs sonunda NSGA2 ile alınan 10 çalıştırma sonucu 149

Şekil Ek-4.11. 62096 numaralı test görüntüsü için 500 FEs sonunda FCM ile alınan 10 çalıştırma sonucu 149

Şekil Ek-4.12. 86016 numaralı test görüntüsü için 500 FEs sonunda IBMO ile alınan 10 çalıştırma sonucu 150

Şekil Ek-4.13. 86016 numaralı test görüntüsü için 500 FEs sonunda NSPSO ile alınan 10 çalıştırma sonucu 150

Şekil Ek-4.14. 86016 numaralı test görüntüsü için 500 FEs sonunda NSGA2 ile alınan 10 çalıştırma sonucu 151

Şekil Ek-4.15. 86016 numaralı test görüntüsü için 500 FEs sonunda FCM ile alınan 10 çalıştırma sonucu 151

Şekil Ek-4.16. 113016 numaralı test görüntüsü için 500 FEs sonunda IBMO ile alınan 10 çalıştırma sonucu 152

Şekil Ek-4.17. 113016 numaralı test görüntüsü için 500 FEs sonunda NSPSO ile alınan 10 çalıştırma sonucu 152

Şekil Ek-4.18. 113016 numaralı test görüntüsü için 500 FEs sonunda NSGA2 ile alınan 10 çalıştırma sonucu 153

Şekil Ek-4.19. 113016 numaralı test görüntüsü için 500 FEs sonunda FCM ile alınan 10 çalıştırma sonucu 153

Şekil Ek-4.20. 135069 numaralı test görüntüsü için 500 FEs sonunda IBMO ile alınan 10 çalıştırma sonucu 154

Şekil Ek-4.21. 135069 numaralı test görüntüsü için 500 FEs sonunda NSPSO ile alınan 10 çalıştırma sonucu 154

Şekil Ek-4.22. 135069 numaralı test görüntüsü için 500 FEs sonunda NSGA2 ile alınan 10 çalıştırma sonucu 155

Şekil Ek-4.23. 135069 numaralı test görüntüsü için 500 FEs sonunda FCM ile alınan 10 çalıştırma sonucu 155

Şekil Ek-4.24. 299091 numaralı test görüntüsü için 500 FEs sonunda IBMO ile alınan 10 çalıştırma sonucu 156

Şekil Ek-4.25. 299091 numaralı test görüntüsü için 500 FEs sonunda NSPSO ile alınan 10 çalıştırma sonucu 156

Şekil Ek-4.26. 299091 numaralı test görüntüsü için 500 FEs sonunda NSGA2 ile alınan 10 çalıştırma sonucu 157

Şekil Ek-4.26. 299091 numaralı test görüntüsü için 500 FEs sonunda FCM ile alınan 10 çalıştırma sonucu 157

ÇİZELGE LİSTESİ

Çizelge 4.1. Gösterge sonuçları 43

Çizelge 6.1. IBMO'nun kontrol parametreleri 71

Çizelge 6.2. GD göstergesi için ortanca ve çeyrekler açıklığı değerleri (25000 fonksiyon değerlendirmesi) 76

Çizelge 6.3. SP göstergesi için ortanca ve çeyrekler açıklığı değerleri (25000 fonksiyon değerlendirmesi) 77

Çizelge 6.4. MS göstergesi için ortanca ve çeyrekler açıklığı değerleri (25000 fonksiyon değerlendirmesi) 78

Çizelge 6.5. HV göstergesi için ortanca ve çeyrekler açıklığı değerleri (25000 fonksiyon değerlendirmesi) 79

Çizelge 7.1. Tüm algoritmalar için ortak kullanılan kontrol parametreleri 97

Çizelge 7.2. 3096 numaralı test görüntüsü için alınan ön-çalıştırma sonuçları 99

Çizelge 7.3. Test görüntüleri için seçilen kontrol parametrelerinin değerleri 99

Çizelge 7.4. Önerilen segmentasyon yöntemi IBMO'nun 3096 numaralı test görüntüsüne 500 fonksiyon değerlendirmesi için uygulanması ile alınan sonuçlar 99

Çizelge 7.5. Önerilen yöntem ve diğer kıyaslama algoritmaları için hesaplanan değerlendirme ölçütlerinin sonuçları 102

Çizelge Ek-1.1. Kısıtsız iki-amaçlı test problemleri 121

Çizelge Ek-1.2. Kısıtlı iki-amaçlı test problemleri 122

Çizelge Ek-1.3. DTLZ test problemleri 123

Çizelge Ek-1.4. Viennet test ailesi ve Tamaki test problemleri 124

Çizelge Ek-2.1. GD göstergesi için ortanca ($\tilde{x}$) ve çeyrekler açıklığı (IQR) değerleri (50000 fonksiyon değerlendirmesi) 126

Çizelge Ek-2.2. SP göstergesi için ortanca ($\tilde{x}$) ve çeyrekler açıklığı (IQR) değerleri (50000 fonksiyon değerlendirmesi) 127

Çizelge Ek-2.3. MS göstergesi için ortanca ($\tilde{x}$) ve çeyrekler açıklığı (*IQR*) değerleri (50000 fonksiyon değerlendirmesi)..... 128

Çizelge Ek-2.4. HV göstergesi için ortanca ($\tilde{x}$) ve çeyrekler açıklığı (*IQR*) değerleri (50000 fonksiyon değerlendirmesi)..... 129

Çizelge EK-3.1. 42049 numaralı test görüntüsü için alınan ön-çalıştırma sonuçları 131

Çizelge EK-3.2. 62096 numaralı test görüntüsü için alınan ön-çalıştırma sonuçları 131

Çizelge EK-3.3. 86016 numaralı test görüntüsü için alınan ön-çalıştırma sonuçları 132

Çizelge EK-3.4. 113016 numaralı test görüntüsü için alınan ön-çalıştırma sonuçları 132

Çizelge EK-3.5. 135069 numaralı test görüntüsü için alınan ön-çalıştırma sonuçları..... 133

Çizelge EK-3.6. 299091 numaralı test görüntüsü için alınan ön-çalıştırma sonuçları..... 133

Çizelge EK-4.1. IBMO'nun 42049 numaralı test görüntüsü için aldığı çalıştırma sonuçları..... 135

Çizelge EK-4.2. IBMO'nun 62069 numaralı test görüntüsü için aldığı çalıştırma sonuçları..... 135

Çizelge EK-4.3. IBMO'nun 86016 numaralı test görüntüsü için aldığı çalıştırma sonuçları..... 135

Çizelge EK-4.4. IBMO'nun 113016 numaralı test görüntüsü için aldığı çalıştırma sonuçları..... 136

Çizelge EK-4.5. IBMO'nun 135069 numaralı test görüntüsü için aldığı çalıştırma sonuçları..... 136

Çizelge EK-4.6. IBMO'nun 299091 numaralı test görüntüsü için aldığı çalıştırma sonuçları..... 136

Çizelge EK-4.7. NSPSO'nun 3096 numaralı test görüntüsü için aldığı çalıştırma sonuçları..... 137

Çizelge EK-4.8. NSPSO'nun 42049 numaralı test görüntüsü için aldığı çalıştırma sonuçları..... 137

Çizelge EK-4.9. NSPSO'nun 62096 numaralı test görüntüsü için aldığı çalıştırma sonuçları..... 137

Çizelge EK-4.10. NSPSO'nun 86016 numaralı test görüntüsü için aldığı çalıştırma sonuçları..... 138

Çizelge EK-4.11. NSPSO'nun 113016 numaralı test görüntüsü için aldığı çalıştırma sonuçları..... 138

Çizelge EK-4.12. NSPSO'nun 135069 numaralı test görüntüsü için aldığı çalıştırma sonuçları..... 138

Çizelge EK-4.13. NSPSO'nun 299091 numaralı test görüntüsü için aldığı çalıştırma sonuçları..... 139

Çizelge EK-4.14. NSGA2'nin 3096 numaralı test görüntüsü için aldığı çalıştırma sonuçları..... 139

Çizelge EK-4.15. NSGA2'nin 42049 numaralı test görüntüsü için aldığı çalıştırma sonuçları..... 139

Çizelge EK-4.16. NSGA2'nin 62096 numaralı test görüntüsü için aldığı çalıştırma sonuçları..... 140

Çizelge EK-4.17. NSGA2'nin 86016 numaralı test görüntüsü için aldığı çalıştırma sonuçları..... 140

Çizelge EK-4.18. NSGA2'nin 113016 numaralı test görüntüsü için aldığı çalıştırma sonuçları..... 140

Çizelge EK-4.19. NSGA2'nin 135069 numaralı test görüntüsü için aldığı çalıştırma sonuçları..... 141

Çizelge EK-4.20. NSGA2'nin 299091 numaralı test görüntüsü için aldığı çalıştırma sonuçları..... 141

Çizelge EK-4.21. FCM'nin 3096 numaralı test görüntüsü için aldığı çalıştırma sonuçları..... 141

Çizelge EK-4.22. FCM'nin 42049 numaralı test görüntüsü için aldığı çalıştırma sonuçları..... 142

Çizelge EK-4.23. FCM'nin 62096 numaralı test görüntüsü için aldığı çalıştırma sonuçları..... 142

Çizelge EK-4.24. FCM'nin 86016 numaralı test görüntüsü için aldığı çalıştırma sonuçları..... 142

Çizelge EK-4.25. FCM'nin 113016 numaralı test görüntüsü için aldığı çalıştırma sonuçları..... 143

Çizelge EK-4.26. FCM'nin 135069 numaralı test görüntüsü için aldığı çalıştırma sonuçları..... 143

Çizelge EK-4.27. FCM'nin 299091 numaralı test görüntüsü için aldığı çalıştırma sonuçları..... 143

SİMGELER VE KISALTMALAR

Simgeler

S	:	Arama uzayı / Bölütlenecek görüntü
Ω	:	Kısıt fonksiyonları kümesi
f	:	Amaç fonksiyonu
P	:	Optimizasyon problemi
x_i^j	:	Kolonideki i.çözümün j.parametresi
x_{min}^j	:	j. karar değişkenin alt sınır değeri
x_{max}^j	:	j. karar değişkenin üst sınır değeri
x_k	:	Çözümün komşu çözümü
v_i	:	Yeni üretilen aday çözüm
SN	:	ABC algoritmasında besin kaynaklarının sayısı
D	:	Karar değişkenlerinin sayısı
fit_i	:	ABC algoritmasında kolonideki i.çözümün uygunluk değeri
$delta$	:	Yeni çözümünün bir parametresi için hesaplanan değişim miktarı
$\emptyset$	:	$[-1, 1]$ aralığında rasgele üretilen bir reel sayı
p_i	:	i.çözümün uygunluk değerine göre hesaplanan seçilme olasılığı
FEs	:	Maksimum uygunluk fonksiyonu değerlendirme sayısı
$pbest$	:	PSO algoritmasının bir parçacık için bulunan en iyi çözüm
$gbest$	:	PSO algoritmasının tüm parçacıklar için bulunan en iyi çözüm
c_1 ve c_2	:	PSO algoritmasında öğrenme katsayıları
$P(\ )$	:	Bağlı piksel gruplarını tanımlayan bir homojenlik fonksiyonu
T	:	Global eşik değeri
C_i veya R_i	:	i. küme / bölge
p_i	:	Görüntüdeki i.piksel
δ	:	İki bölge arasındaki farklılık ölçüsü
m_j	:	j. küme merkezi

n_j	:	j. kümedeki toplam piksel sayı
J_m	:	En küçük kareler yöntemi
u_{ij}	:	x_i verisinin j.kümeye olan üyelik derecesi
$\|\| * \|\|$	:	FCM'de verilerle küme merkezleri arasındaki benzerliği ifade eder
U	:	FCM'de üyelik matrisi
J_e	:	Niceleme hatası
d_{min}	:	Farklı kümeler arasındaki minimum mesafe ölçütü
d_{max}	:	Tüm kümeler arasında maksimum küme-içi uzaklık
$\prec$	:	İki nondominated çözüm adayı arasında soldakinin sağdakini bastırıp bastırmadığını ifade eder
R_t	:	NSGA2'de 2N boyutlu popülasyon
EXA	:	Harici arşiv
nEXA	:	Harici arşiv boyutu
CD	:	Kalabalıklık mesafesi
$dist(p_1, p_2)$	:	p_1 ve p_2 pikselleri arasındaki Öklid uzaklığı
TF	:	Doku özniteliği
$scat$	:	Dağılma (*scattering*)
$Dens_{bw(K)}$	:	Yoğunluk (*density*)

Kısaltmalar

ABC	:	Yapay Arı Kolonisi
AbYSS	:	Archive-Based Hybrid Scatter Search to Multiobjective Optimization
BCE	:	İki-yönlü tutarlılık hatası
BBS	:	Bastırılmamış çözüm sayısı
BÇS	:	Bulunan uygun çözüm sayısı
BSDS	:	Berkeley görüntü segmentasyonu veri tabanı
DTLZ	:	Deb ve arkadaşlarının oluşturduğu bir test paketi

FCM	:	Bulanık C-Ortalamalar Algoritması
GCE	:	Global tutarlılık hatası
GD	:	Genel Uzaklık Göstergesi
HV	:	Hypervolume Göstergesi
IBMO	:	Çok Amaçlı Optimizasyon için İyileştirilmiş Yapay Arı Kolonisi Algoritması
IQR	:	Çeyrekler açıklığı (Interquartile range)
K-Means	:	K-Ortalamalar Algoritması
LCE	:	Yerel tutarlılık hatası
Limit	:	İyileştirme metodu için gerekli olan maksimum adım sayısı
LRE	:	Yerel düzeltme hatası
MNC	:	Maksimum çevrim sayısı
MR	:	Modifikasyon Oranı
MS	:	Maksimum Yayılma Göstergesi
NP	:	Koloni boyutu
NSGA2	:	Bastırılmamışlık Sıralamalı Genetik Algoritma 2
NSPSO	:	Bastırılmamışlık Sıralamalı Parçacık Sürü Optimizasyonu
OMOPSO	:	Optimal Multi-Objective Particle Swarm Optimization
PSO	:	Parçacık Sürü Optimizasyonu
S_Dbw	:	Scattering ve Density küme doğruluk göstergesi
SF	:	Ölçekleme Faktörü
SP	:	Yayılma Göstergesi
SPEA2	:	The Strength Pareto Evolutionary Algorithm 2
SRG	:	Çekirdekli Bölge Genişletme
URG	:	Çekirdeksiz Bölge Genişletme
ZDT	:	Zitzler-Deb-Thiele'nin test paketi

1. GİRİŞ

Görüntü segmentasyonu; görüntü işleme, örüntü tanıma ve görüntü analizi için temel teşkil eden kritik bir öneme sahiptir. Segmentasyon ayrık ve homojen bölgelerin belirlenmesi veya görüntüde yer alan objelerin kenarlarının veya sınırlarının tespit edilmesi ile görüntünün parçalara ayrılmasını sağlayan düşük-seviye işlemdir. Temel amacı görüntüyü belirli uygulamalar için anlamlı olan farklı homojen parçalara ayrıştırmaktır. Bu işlem sırasında homojen bölgelerin veya kenarların, görüntülerdeki objeler veya bu objelerin parçaları oldukları varsayılır. Çok sayıda görüntü işleme ve bilgisayar görüşü uygulamasında görüntülere tanıma, semantik yorumlama gibi daha yüksek seviye işlemler uygulanmadan önce, segmentasyon işlemi ilk adım olarak önemli bir rol oynar. Çünkü segmentasyonun kalitesi tanıma ve sorgulama sistemlerinin kalitesini doğrudan etkilemektedir. Bununla birlikte tüm görüntülere başarılı bir şekilde uygulanacak genel bir segmentasyon yöntemi yoktur (Gonzales ve Woods, 2002).

Renkli görüntü boyutu itibariyle gri seviyeli görüntüden daha fazla bilgi içerir. Renk bilgisi görüntülerin daha eksiksiz ve sağlıklı gösterimine imkân tanır. Buna karşın yakın zamana kadar görüntü işleme uygulamalarındaki araştırmalar çoğunlukla gri-seviyeli görüntüler üzerinde yapılmıştır. Çünkü renkli görüntü, segmentasyonun kalitesi ile beraber problemin hesaplama karmaşıklığını da artırır. Renkli görüntüleri işlemek için gereken hesaplama süresi, gri-seviyeli görüntüler için ihtiyaç duyulan zamandan önemli ölçüde fazladır. Ancak günümüzde bilgisayar teknolojisinin hızlı bir gelişme göstermesi ve bunun yanı sıra nieten daha ucuz renkli kameraların üretilmeye başlanması ile görüntü işlemede renkli görüntüler tek renkli bilginin yerini almaya başlamıştır. Artık hesaplama güçlüğü büyük hacimli verilerin işlenmesinde bir sınırlama olmaktan çıkmıştır. Bu nedenle son zamanlarda çalışmalar renkli görüntü segmentasyon algoritmaları ve bununla ilgili çeşitli teknikler üzerinde yoğunlaşmıştır. Bunların birçoğu gri-seviyeli görüntüler için geliştirilmiş tekniklerin (histogram-eşikleme, özellik kümeleme, piksel-tabanlı, kenar-tabanlı, bölge-tabanlı yöntemler, bulanık teknikler, yapay sinir ağları vb.) boyutlarını genişletmeye dayanır. Renkli görüntünün birçok bilgi taşıması bir avantaj olsa da renk bilgilerinin üç boyutlu renk uzaylarıyla temsil edilmesi ve renkler arasındaki farkın tespit edilme zorluğu ile doygunluk, renk geçişi, parlaklık vb. birçok etki renkli görüntülerin segmentasyonunu oldukça güçleştirmektedir (Dass ve ark., 2012; Gonzales ve Woods, 2002).

Bu güçlüğün üstesinden gelmek için görüntüdeki tüm olası bölümlenmeleri bulacak ve değerlendirecek bir kaba kuvvet metodu kullanılabilir. Ancak bu durum son derece geniş bir arama uzayının oluşmasına neden olacağından, bu metodu uygulamak çoğu zaman mümkün değildir (Bhandarkar ve Zhang, 1999). Ayrıca bir segmentasyon yöntemi seçildikten sonra, kaliteli bir segmentasyon yapılabilmesi için ayarlanması gereken çok sayıda parametre vardır. Birçok yöntem için bu parametre değerlerini ayrıntılı bir şekilde araştırmak da mümkün değildir.

Bu bağlamda; segmentasyon işlemi bir optimizasyon problemi olarak formülize edilebilir. Tasarlanacak problem, başarılı bir segmentasyon sonucu için algoritma parametrelerinin optimal değerlenin tespit edilmesi ya da olası segmentasyonlar arasındaki optimal segmentasyonun bulunması olarak düşünülebilir. Her iki durumda da taranması gereken geniş bir arama uzayı vardır (Bong ve Mandava, 2010). Bununla birlikte; renkleri homojen kümelere ayrıştırmak için, sürü zekâsına dayanan optimizasyon algoritmaları kullanılarak sürü davranışlarının modellenmesiyle pikseller birbirlerinden ayrıştırılabilir ve istenen bölgelerin ortaya çıkarılması sağlanabilir. Literatürde bu amaçla kullanılan hem evrimsel hesaplama hem de sürü zekâsına dayalı algoritmaların kullanıldığı çalışmalar vardır. Ancak bu çalışmaların büyük çoğunluğu gri-seviye görüntüler için gerçekleştirilmiştir.

Çalışma kapsamında; renkli görüntüler üzerinde odaklanılmış ve renkli görüntü segmentasyon tekniklerine sürü zekâsı algoritmalarının uyarlanmasıyla yeni segmentasyon algoritmalarının geliştirilmesi amaçlanmıştır. Son birkaç sene içinde sürü zekâsına dayanan zeki optimizasyon algoritmalarından parçacık sürü optimizasyonu ve karınca koloni optimizasyonu algoritmalarının renkli görüntü segmentasyonu üzerinde kullanıldığı başarılı denemeler literatürde yer almaya başlamıştır (Aydın, 2011). Bununla birlikte ABC algoritmasının bu alanda uygulandığı çalışma sayısı yok denecek kadar azdır.

Tez çalışması kapsamında; tek-amaçlı optimizasyon problemlerinin çözümünde başarısı ispatlanmış olan yapay arı kolonisi optimizasyon algoritmasının renkli görüntü segmentasyonuna uyarlanması hedeflenmiştir. Gerçekleştirilen çalışma üç aşamada açıklanmıştır. İlk olarak, temel tek-amaçlı ABC algoritması gri-seviyeli görüntülere kümeleme yoluyla uygulanmıştır. Niceleme hatası ve küme doğruluk göstergelerinden alınan sonuçlar PSO ile üretilen sonuçlarla karşılaştırılmıştır. Görüntü segmentasyonu ele alınış biçimi itibariyle bir kümeleme problemi olarak düşünüldüğünde içerdiği karar değişkenlerine göre çok-amaçlı bir optimizasyon problemi olarak modellenebilir (Bong

ve Mandava, 2010; Shirakawa ve Nagao, 2009). Bu amaçla ikinci aşamada; tek-amaçlı temel ABC algoritması çok-amaçlı optimizasyon problemlerine uyarlanmıştır. Geliştirilen yeni algoritmanın başarısı test problemleri üzerinde denenmiştir. Sonuçlar literatürde önde gelen en başarılı çok-amaçlı optimizasyon algoritmaları ile karşılaştırılmıştır. Son aşamada; geliştirilen ABC-temelli çok-amaçlı optimizasyon algoritması renkli görüntülerin segmentasyonunda kullanılmıştır. Önerilen algoritma Berkeley görüntü segmentasyon veri kümesindeki test görüntüleri üzerinde uygulanmıştır. Elde edilen segmentasyon sonuçlarının bölge-tabanlı tutarlılık göstergeleri hesaplanmıştır. Ayrıca sonuçlar Bastırılmamışlık Sıralamalı Genetik Algoritma II (NSGA-II), Bastırılmamışlık Sıralamalı Parçacık Sürü Optimizasyonu (NSPSO) ve Bulanık C-Ortalamalar (FCM) algoritmalarından aynı şartlar altında alınan segmentasyon sonuçlarıyla karşılaştırılmıştır. Elde edilen başarılı sonuçlar önerilen yöntemin genel amaçlı renkli görüntü segmentasyonu için tutarlı bir doğruluğa sahip olduğunu göstermiştir.

1.1. Çalışmanın Amacı ve Önemi

Akademik alanda ve endüstriyel çevrelerde yapılan uzun çalışmalardan sonra, görüntü işleme ve bilgisayarla görme uygulamaları başlıca endüstriyel faaliyetlerden biri haline gelmiştir. Video oyunları ve filmlerde kullanılan özel efektler gibi uygulamaların yer aldığı eğlence endüstrisinin yanı sıra birçok üretim ortamında otomatikleştirilmiş görsel denetim araçlarının kullanılması üretim zincirinin maliyet ve kalite optimizasyonu için önemlidir. Bununla birlikte, görüntü işleme ve bilgisayarla görme uygulamalarının bu denli kullanımının artmış olması tek başına bu alandaki tüm problemlerin çözümü için her zaman uygulanabilir olduğu anlamına gelmez. Çünkü tasarlanan bir bilgisayar uygulaması, sadece belirli problemlerin çözümüne yöneliktir. Bir problemin tanımı boyutsal olarak büyüdüğünde, problemin normal aralıklarının dışında bir veya daha fazla özelliğinin olması doğaldır. Bu yüzden gerçek dünya uygulamaları genellikle yüksek bir belirsizlik derecesiyle tanımlanan yüksek-boyutlu verilerle uğraşmak zorunda kalır. Bu durum, özellikle görüntü işleme ve bilgisayarla görme uygulamaları için de geçerlidir (Cagnoni ve ark., 2007).

Örüntü tanıma problemleriyle karşı karşıya olan birçok bilgisayarla görme sisteminin düzgün bir şekilde çalışması çok sayıda iyi ayarlanmış parametreye bağlıdır. Örneğin bir trafik izleme uygulamasında bazı parametrelerin dinamik olarak

ayarlanması ve çok sayıda başka parametrenin de sistem çalışmaya başlamadan statik olarak ayarlanması gerekir. Bu işlemin ayrıca bir parametre optimizasyonu olarak ele alınması etkin ve karalı sistemler üretmek için daha iyi bir çözüm yolu olacaktır.

Bu bağlamda; gerçek dünya problemleri için yapılacak çözüm yaklaşımlarının iyi olması, çeşitli faktörleri içermesine bağlıdır. Bunların en önemlisi ilgi alanında kapsamlı bir ön bilgiye sahip olmaktır. Eğer problem alanı hakkında tam bilgiye sahip olunursa, çözümü tasarlamak sadece problemin fenomenlerini düzenleyecek kuralları modellemekten geçer. Ancak bir problem hakkında tam bilgiye sahip olmak demek, bazı düzensizlikleri hesaba katmayı gerektirir. Bu nedenlerle problemin eksiksiz bir matematiksel modelinin oluşturulamaması bu durumu daha da kötüleştirir. Çünkü uygulamanın tüm durumlar için eşit derecede iyi olacak bir çözüm bulma şansı hemen hemen yoktur. Bu yüzden karar vericinin yapması gereken her durum için veya çoğu durum için "yeterince iyi" olan bir uzlaşma sağlamaktır. Diğer bir ifadeyle bu durum, sadece problem bilgisine bağlı kalarak değil aynı zamanda gözlemler sonucu edinilmiş tecrübeye dayalı ipuçlarıyla en iyi çözümü bulabilmek anlamına gelir. Bunlar makinelerden daha ziyade insanların sahip olduğu becerilerdir (Coello, 1998).

Esnek hesaplama alanında kullanılan teknikler (yapay sinir ağları, genetik ve evrimsel hesaplama, bulanık mantık, sürü zekâsı vb.) özellikle bahsi geçen problemlerin üstesinden gelecek etkili araçlardır. Sürü zekâsına dayanan optimizasyon algoritmaları, diğer esnek hesaplama paradigmalarında olduğu gibi, kendi kendini geliştirme, öğrenme, genelleme, değişkenliğe karşı dayanıklılık, adaptasyon ve bunun gibi canlılara ait doğal yetenekleri taklit edebilen bilgisayarları geliştirmeyi amaçlar.

Segmentasyon; örüntü tanıma, görüntü analizi, görsel gözetleme sistemleri ve hemen hemen tüm görüntü işleme ve makine-görüşü uygulamalarının başlangıç adımını teşkil eden düşük seviye görme işlemlerinin en önemlisidir. Bu nedenle, oluşturulacak sistemin kalitesini belirmede son derece etkilidir. Segmentasyonun kalitesi ise üzerinde çalışılan görüntünün taşıdığı renk bilgisi ve uygulanan algoritma ile paraleldir. Gelişen bilgisayar teknolojisinin sağladığı avantajla 3 boyutlu renk uzayında ve yüksek çözünürlükte yakalan görüntüler, segmentasyon işleminde yüksek boyutlu verilerle çalışmaya imkân sağlamaktadır. Bu amaçla renkli görüntüler üzerinde sürü zekâsı temelli paralel işlem yeteneğine sahip algoritmaların kullanılması ile yeni segmentasyon teknikleri geliştirilebilir. Geliştirilen yeni algoritmaların uygulama alanın genişliği göz önüne alındığında bu durum gerek literatür gerekse pratik bakımından önem arz eder.

Görüntü işlemenin ön aşamalarını oluşturan düşük seviye görme işlemleri üzerinde günümüze kadar birçok çalışma yapılmıştır. Bu işlemler tipik olarak; görüntü filtreleme, yumuşatma, zenginleştirme, gürültünün giderilmesi, parlaklık hesabı, kenar ve nokta tespiti, yeniden örnekleme, renk indirgeme ve sıkıştırma olarak sıralanabilir. Düşük seviye görüntü işleme genellikle görüntülerde yakın komşuluk ilişkilerini, morfolojik özellikleri veya 3D geometriyi dikkate alır. Bilgisayarla görme alanının tümünde çıktıların kalitesi çok önemlidir. Bu konu daima ilgi çekici olmuştur. Son yıllarda karmaşık optimizasyon yöntemlerinin kaynağı olan çok yönlü matematiksel teoriler ve istatistik metotları geliştirilmiştir. Buna ek olarak gömülü, gerçek-zamanlı bilgisayarla görme sistemlerindeki yeni kısıtlar; güçlü, esnek ve aynı zamanda uygun maliyetli algoritmaları zorunlu kılmaktadır.

1.2. Kaynak Araştırması

Görüntü segmentasyonu uygulama alanına göre özel olarak tasarlanması gereken oldukça zorlayıcı bir problemdir. Bu amaçla geliştirilen algoritmaların başarısı da büyük ölçüde bölge homojenliği için kullanılan benzerlik ölçütlerinin doğru tanımlanmasına bağlıdır. Görüntü segmentasyon teknikleri genellikle eşikleme-tabanlı, kenar belirleme, bölge-tabanlı ve kümeleme-tabanlı teknikler olarak sınıflandırılmaktadır (Gonzales ve Woods, 2002; Lucchese ve Mitra, 2001; Narkhede, 2013). Bu teknikler üçüncü bölümde detaylı olarak ele alınmaktadır. Bunların dışında genetik algoritmalar ve evrimsel hesaplamalar; içinde karmaşık optimizasyon problemleri barındıran görüntü segmentasyon problemleri için önerilmiş efektif araçlardır. Ancak çoğunlukla donanımsal gerekçelerle şimdiye kadar gri seviyeli görüntüler üzerinde çalışmalar yapılmıştır. Bunun yanı sıra son çalışmalar göstermektedir ki; sürü zekâsına dayanan karınca koloni optimizasyonu (KKO) ve parçacık sürü optimizasyonu (PSO) gibi algoritmalar, segmentasyon probleminin çözümünde kullanılabilen paralel işlem yeteneğine sahip faydalı tekniklerdir. Bu çalışmaların da birçoğu gri seviyeli görüntüler üzerinde gerçekleştirilmiştir. Dikkat çekici olan bir diğer nokta ise bir sürü zekâsı algoritması olan ABC için bu alanda yapılmış çok az sayıda çalışma bulunmasıdır.

Bhanu ve ark. (1995) görüntü segmentasyonunu bir optimizasyon problemi olarak tasarlamışlardır. Genel bir segmentasyon yöntemi tanımlayarak bunu birçok metodun parametrelerine uygulamışlardır. Çalışmada renkli dış mekân görüntüleri kullanmışlar ve Phoenix segmentasyon algoritmasının 4 parametresini genetik algoritmaya adapte etmişlerdir.

Bhandarkar ve Zhang (1999), genetik algoritma tek başına ya da diğer yerel arama teknikleri ve evrimsel algoritmalar ile birleştirildiğinde görüntü segmentasyonu probleminde verimli bir optimizasyon aracı olarak kullanılabileceğini iddia ettiler. Araştırmacılar genetik algoritmalar ile benzetimsel tavlama temelli teknikleri birleştirerek gri seviyeli görüntüler için yeni bir segmentasyon yaklaşımı önermişlerdir. Klasik genetik algoritmadan daha iyi bir performans alındığını gözlemlemişlerdir.

Ramos ve Muge (2000), görüntü segmentasyonunu bir kümeleme problemi olarak yeniden düzenlediler ve optimal kümelerin bulunması amacıyla genetik algoritmaları kullanmışlardır. Kromozomlar hangi piksellerin hangi bölgelerin elemanı olacağını gösterecek şekilde kodlanmıştır. Önerilen yöntemin en büyük sorunu, kümelerin sayısının girdi olarak verilmesi ve dolayısıyla danışmansız olarak çalışamıyor olmasıdır. Bu yüzden genel bir segmentasyon yöntemi değildir.

Zingarette ve ark. (2002) danışmansız renkli görüntü segmentasyonunda genetik algoritmaları kullanmayı önermişlerdir. Bu metotta çok-geçişli eşikleme kullanılmıştır. Farklı geçiş değerleri, genetik algoritmaların her bir geçiş boyunca uyarlanır. Bu yöntemin öncekine göre en büyük avantajı tamamen danışmansız olarak çalışabilmesidir. Herhangi bir öncü bilgiye gereksinim duymadan çok çeşitli görüntüler üzerinde başarılı sonuçlar elde edilmiştir. Ancak yoğun dokuya sahip görüntülerde tam olarak istenen sonuçlar alınamamıştır.

Pignalberi ve ark. (2003) da segmentasyon işlemini, genetik algoritmalarda optimizasyon problemi olarak ele almışlardır. Odaklandıkları nokta; nesne ile sensor arasındaki uzaklığa bağlı olarak bir pikselin renginde meydana gelen değişiklik ile görüntüdeki mesafe bilgisinin hesaplanmasıydı. Bu yöntemle 3B nesnelerin dış yüzeyleri bölütlenebilirken, 2B görüntülerin segmentasyonu yapılamamaktadır.

Piksel-düzeyinde yapılan segmentasyon yöntemlerinde; bir piksel özelliklerine bağlı olarak özel bir bölge içinde sınıflandırılır. Peng ve ark. (2000) bu yaklaşımı kullanarak, resimdeki her bir pikseli bir kromozom olarak temsil etmişlerdir.

Farmer ve Shugars (2006) görüntü segmentasyonunda genetik algoritmaların kullanımını (1) parametre seçimi ve (2) piksel düzeyinde segmentasyon olmak üzere iki önemli sınıfa bölmüşlerdi. Parametre seçimi; mevcut bir segmentasyon metodunun parametrelerini ayarlayarak sonucu iyileştirmek için kullanılan yöntemlerdir. Piksel-düzeyinde segmentasyon ise bölge etiketlemeyi gerçekleştirmek için kullanılan

yöntemlerdir. Çoğu segmentasyon metodunda optimal segmentasyon sonucunu elde etmek için ayarlanması gereken çok sayıda parametre, sabit ve eşik değer vardır. Bu yüzden çoğu uygulamada parametre optimizasyonu olan ilk sınıf daha sık kullanılmıştır.

Feitosa ve ark. (2006) benzer bir yaklaşım geliştirerek genetik algoritmaları, bir bölge-büyütme segmentasyon algoritmasının parametrelerini ayarlamak için uyguladılar. Uygunluk fonksiyonu olarak, sonuçta bulunan parçaların bir kullanıcı tarafından sağlanan hedef segmentasyona olan benzerliği kullanılmıştır. Hesaplaması doğru ve sezgisel olmasına rağmen, önceden bilinen manuel segmentasyon gerektirmektedir.

Ramos ve Almeida (2000), yapay karınca kolonisi için arama uzayı olarak sayısal görüntünün piksellerinde dolaştıkları bir çevre tasarlanmıştır. Bu çalışma ile yapay karınca kolonilerinin her türlü sayısal yaşam türüne karşı tepki verebileceğini ve uygun bir şekilde adapte edilebileceğini göstermişlerdir. Ayrıca karınca kolonilerinin kullanıldığı literatürdeki ilk bölge genişletme yaklaşımıdır.

Omran ve ark. (2005) görüntü segmentasyonu için parçacık sürü optimizasyonuna dayanan dinamik bir kümeleme yaklaşımı önerdiler. Küme merkezlerinin sayısını otomatik olarak belirlemek amacıyla önerilen yaklaşımda, veri kümesi çok sayıda parçaya bölünerek arama işlemi başlatılır. Önerdikleri yöntem K-Manş algoritmasından faydalanarak başlangıçta rasgele atanan küme merkezlerinin etkinliğine göre merkez sayısını azaltır ve optimal sayıyı bulmayı hedefler. Farklı türlerde görüntüler üzerinden alınan sonuçlar olumlu olmakla birlikte bölge-tabanlı bir segmentasyon algoritması olarak çalışmaz. Daha çok renk indirgemesi olarak kullanılabilir. Ayrıca bu çalışma da sadece gri-seviyeli görüntüler için önerilmiştir.

Huang ve ark. (2008), yapay karınca kolonilerinin özellikleri üzerinde durularak görüntü segmentasyonu için genişletilmiş bir model sunmuşlardır. Önerilen modelde her karınca yenisiyle karşılaşıncaya kadar bir referans nesneyi hafızasında saklar. Hedef ve referans nesne arasındaki benzerliği değerlendirmek için bulanık bir bağlantı ölçütü kullanılır. Önerilen yaklaşım gri-seviyeli beyin MR görüntülerinin segmentasyonu amacıyla kullanılmıştır.

Ma ve ark. (2011), sentetik aralıklı radar (SAR) görüntülerinin segmentasyonu için yapay arı kolonisine dayanan hızlı bir segmentasyon metodu önermişlerdir. Eşikleme tabanlı bir segmentasyon yöntemi kullandılar. Gri-seviye görüntüler için uygun eşik değer belirlenmesi bir arama prosedürü olarak modellenmiştir ve ABC ile

optimal eşik değer belirlenmeye çalışılmıştır. Etkili bir uygunluk fonksiyonu elde etmek amacıyla Grey teorisine göre gri seviye tanımları yapıldıktan sonra, görüntüler ayrık dalgacık dönüşümü ile sıkıştırılmıştır. Daha sonra filtrelenmiş görüntüden eş-benzerlik matrisi çıkarılmış ve gradiyent görüntüsü oluşturulmuştur. Buradan elde edilen 2-boyutlu gri entropi değerleri amaç fonksiyon olarak kullanılmıştır.

Cinsdikici ve Aydın (2009), bir eşleme filtresi (MF) ve karınca kolonisinin kullanıldığı melez bir yaklaşımı önermişlerdir. MF'nin kusurlu yönlerini kapatmak için geliştirilen metot oftalmoskop ile alınan medikal görüntülerden gözdeki kan damarlarının doğru bir şekilde tespit edilmesi amacıyla kullanılmıştır.

Akay (2013), PSO ve ABC algoritmalarının çok-seviyeli eşikleme amacıyla kullanmış ve arama uzayında gri seviyeli görüntüler üzerindeki performanslarını karşılaştırmalı olarak ortaya koymuştur. Bu amaçla maksimum entropi tekniklerinden biri olan Kapur entropisini ve sınıflar-arası farklılığı iki ayrı uygunluk fonksiyonu olarak kullanılmıştır.

Görüntü segmentasyonu pek çok yönüyle bir optimizasyon problemi olarak modellenebilmektedir. Bu bakış açısıyla; segmentasyonun karar verme sürecinde birden fazla amacı ayrı ayrı değerlendiren yaklaşımlar literatürde yer almaya başlamıştır. Çünkü birçok görüntü segmentasyon problemi gerçekten de özellikleri bakımından çok amaçlı optimizasyon problemlerine dönüştürülebilir. Çok-amaçlı optimizasyon algoritmaları, tek-amaçlı optimizasyonun bir uzantısı olarak geliştirilmiştir. Çalışma ilkeleri; birden fazla ve birbiriyle çelişen amaçların eş zamanlı olarak optimal çözümlerinin aramasına dayanır. Bu teknikler elektromanyetik soğurucu tasarımı (Mahmoud ve ark., 2007), I-beam için tasarım değişkenlerinin tanımlanması ve yüklenmesi (Sağ ve Çunkaş, 2009), gibi birçok mühendislik probleminin çözümünde başarılı bir şekilde kullanılmıştır.

Shirakawa ve Nagao (2009), çok-amaçlı evrimsel bir kümeleme algoritmasını görüntü segmentasyonu için uyarlamışlardır. Bu metotta toplam sapma ve kenar değerleri iki amaç fonksiyon olarak kullanılmıştır. MOCK metodu ile benzerlik gösteren pikselleri yönlü graflar şeklinde kodlayıp fenotipleri aday çözüm olarak kullanmışlar ve çeşitli gri-seviye görüntüler üzerinde uygulamışlardır. Kenar çıkarmaya dayalı önerilen segmentasyon metodu için ayrıca elde edilen çözüm kümesinden bir çözüm belirlemek için de bir sezgisel yaklaşım önermişlerdir.

Nakib ve ark. (2010), Pareto teorisine dayanan çok-amaçlı bir optimizasyon algoritması olan NSGA2 kullanılarak gri seviyeli görüntüler için yeni bir eşikleme metodu önermişlerdir. Bu metotta segmentasyon kalitesini arttırmak amacıyla sınıflar arası farklılık, Shannon entropisi ve iki-boyutlu entropi yöntemlerinin birer amaç fonksiyonu olarak kullanılması önerilmiştir.

Saha ve Bandyopadhyay (2010), multispektral beyin MR görüntülerinin otomatik segmentasyonu için yeni bir metot önermişlerdir. Bu amaçla segmentasyon problemi gri-seviye uzayında çok-amaçlı bir kümeleme problemi olarak modellenmiştir. Bu amaçla bir takım küme doğrulama göstergelerinin eş-zamanlı optimizasyonu yapılmıştır. Bu göstergeler XB-indeks ve Sym-indeksleridir. Benzetimsel tavlama algoritmasının çok-amaçlı optimizasyona uyarlanmasıyla elde edilen yaklaşım çok çekirdekli kümeleme algoritması olarak kullanılmış ve alınan sonuçlar mevcut manuel sonuçlarla karşılaştırılmıştır.

Saha ve ark. (2010), yaptıkları çalışmayla yeni çok-amaçlı bir diferansiyel evrim temelli bulanık kümeleme tekniği önermişlerdir. Önceki çalışmada önerdikleri kümeleme göstergelerini kullanarak uydu görüntüleri üzerinde gri-seviye bakımından benzer özellik gösteren piksel gruplarının kümelemişlerdir.

Bong ve Rajeshwari (2011), görüntü segmentasyonu amacıyla kullanılan sezgisel tekniklerle ilgili olarak kapsamlı bir literatür incelemesi yapmışlardır. Özellikle çok-amaçlı kümeleme ve sınıflandırma yaklaşımları üzerinde durmuşlardır. Çok-amaçlı optimizasyon tekniklerinin görüntü segmentasyonunda uygulama alanları ve değerlendirme ölçütleri hakkında detaylı bir çalışma sunmuşlardır.

Wang ve ark. (2011), destek vektör makinelerinden faydalanarak piksel temelli yeni bir renkli görüntü segmentasyon metodu önermişlerdir. Bu amaçla piksel düzeyinde düşük seviyeli bir öznitelik vektörü çıkarmışlardır. Öncelikle bu öznitelik vektörünü FCM ile kümelemişler ve sonrasında alınan sonuçları SVM de eğitim amacıyla kullanmışlardır. Eğitilmiş SVM algoritması benzer görüntüler için danışmansız şekilde segmentasyon işlemine uygulamışlardır.

Benaichouche ve ark. (2013), FCM ile görüntü segmentasyonu için bir iyileştirme metodu önermişlerdir. Bu metot ile algoritmaya üç aşamalı bir katkı sağlanması hedeflenmiştir. Birinci adımda FCM'nin başlangıç noktalarını yönlendirmek için sezgisel bir yaklaşım önerilmiştir. İkinci adımda segmentasyonda gri-seviye

değerleri benzerliğinin yanı sıra konumsal yakınlıkları da hesaba katılmıştır. Son olarak Mahalanobis uzaklığı kullanılarak farklı sınıflardaki geometrik şekil farklılığının etkisi azaltılmaya çalışılmıştır. Önerilen yöntem sentetik görüntüler ve beyin MR görüntüleri üzerinde denenmiştir.

Fornarelli ve Giaquinto (2013), PSO temelli bir kümeleme algoritması ile görüntü segmentasyonu için yeni bir yaklaşım önermişlerdir. Bu yaklaşımda çok sürülü bir kümele tekniği görüntü segmentasyonuna uygulanmıştır. Gri seviyeli görüntüler üzerinde gerçekleştirilen bu işlem iki aşamadan oluşur. Öncelikle klasik biçimde sürü işlemleri gerçekleştirilir. Daha sonra denk sürülere bölünerek alınan sonuçla iyileştirilmeye çalışılır. Önerilen yöntem danışmansız olarak gri seviyelerinin sayısını otomatik bir biçimde tespit etmeye çalışır. Yöntemin doğrulu kümeleme göstergeleriyle ispatlanmaya çalışılmıştır.

Sonuç olarak; yapılan kaynak araştırmasına göre düşük-seviye görüntü analizinin en zor ve en kritik aşaması olan segmentasyon işleminin aynı zamanda pek çok yönüyle bir çok-amaçlı optimizasyon problemi olduğu açıkça görülmüştür. Ayrıca segmentasyon işlemi, uygulama eksenli değerlendirilme gerektiren öznel bir problemdir. Bu bakış açısıyla; çok-amaçlı sezgisel optimizasyon tekniklerinin segmentasyon işlemine adaptasyonu ile geliştirilecek yeni çözümlere ihtiyaç olduğu görülmektedir. Diğer taraftan; sezgisel optimizasyon literatüründe yapılan son çalışmalar paralel işlem yeteneğine sahip sürü zekâsı teknikleri üzerinde yoğunlaşmaktadır.

1.3. Tezin Organizasyonu

Bu tez çalışmasında sürü zekâsı teknikleri kullanılarak görüntü segmentasyonu yapılmıştır. Çalışma toplam sekiz bölüm ve ekler kısmından oluşmaktadır.

Birinci bölüm giriş bölümü olup konunun genel tanımı yapılmış, çalışmanın amacı ve önemi üzerinde durulmuştur. Konu ile ilgili önceden yapılmış olan benzer çalışmalar hakkında araştırmalar yapılmış, bilgiler verilmiş ve kısa bir analiz yapılmıştır.

İkinci bölümde; sürü zekâsı algoritmalarının ortaya çıkışı, tanımı, kullanım alanları ve optimizasyon problemleriyle olan bağlantısı üzerinde durulmuştur. Ayrıca tez çalışmasında üzerinde odaklanılan başlıca algoritma olan yapay arı kolonisi (ABC) algoritmasının çalışması detaylarıyla anlatılmıştır. Bunun yanı sıra literatürde en iyi

bilinen ve yaygın şekilde kullanılan algoritmalardan bir diğer sürü zekâsı algoritması olan parçacık sürü optimizasyonu (PSO) algoritmasının çalışması da ele alınmıştır.

Üçüncü bölümde; görüntü segmentasyonuna giriş yapılmış, segmentasyon tekniklerinin sınıflandırılması yapılmış, kullanılan başlıca yöntemler açıklanmıştır. Ayrıca renk uzayları üzerinde durularak başlıca renk modelleri tanıtılmış ve renkli görüntü segmentasyonu için kullanılması uygun olanların altı çizilmiştir.

Dördüncü bölümde; PSO ve ABC temelli kümeleme algoritmaları ile gri-seviyeli görüntüler için segmentasyon metotları önerilmiş ve sonuçlar değerlendirilmiştir.

Beşinci bölümde; çok-amaçlı optimizasyon problemleri ve segmentasyon arasındaki yakın bağlantı açıklanmıştır. Daha sonra Pareto-Optimal kavramı ve çok-amaçlı optimizasyon problemlerinin kavramsal ve matematiksel tanımlamalarına yer verilmiştir. Son olarak bu alanda önde gelen ve birçok algoritmanın temelini oluşturan kavramsal çerçevenin oluşturulasında literatüre en büyük katkıyı sağlayan algoritmalardan birisi olan Bastırılmamışlık Sıralamalı Genetik Algoritma 2 (NSGA2) algoritmasının çalışması detaylı olarak anlatılmıştır.

Altıncı bölümde; tez çalışması kapsamında geliştirilen yapay arı kolonisi temelli yeni bir çok-amaçlı optimizasyon algoritması önerilmiştir. Önerilen algoritmanın çalışması detaylarıyla açıklandıktan sonra literatürde sıkça kullanılan birçok test problemi üzerinde uygulanmıştır. Elde edilen deneysel sonuçlar ve bu sonuçlara ait gösterge değerleri bu bölümde sunulmuştur. Ayrıca kullanılan değerlendirme ölçütleri de kapsamlı şekilde açıklanmıştır.

Yedinci bölümde; geliştirilen çok-amaçlı optimizasyon algoritması renkli görüntü segmentasyonu için uyarlanmış ve önerilen metot test görüntülerine uygulanmıştır. Bu amaçla oluşturulan öznitelik vektörünün yapısı, kullanılan teknikler, değerlendirme ölçütleri ve sunulan segmentasyon algoritmasının detayları bu bölümde irdelenmiştir. Son olarak sonuçlar üzerinde değerlendirmeler yapılmıştır.

Sekizinci bölüm, tez çalışmasında elde edilen sonuçlar tartışılmış ve bu konu üzerinde çalışma yapmak isteyen araştırmacılara önerilerde bulunulmuştur.

Ekler kısmında; önerilen algoritmalardan elde edilen sonuçların ayrıntılı çizelgeleri ve kullanılan test problemlerinin matematiksel tanımlamaları verilmiştir.

2. SÜRÜ ZEKÂSI

Sürü zekâsı; optimizasyon ve robotik başta olmak üzere çok-etmenli sistem tasarımıyla ilgili modern bir yapa zekâ disiplinidir. Bu sistemlerin tasarım paradigması klasik sistemlerden temel olarak daha farklıdır. Karıncalar, termitler, bal ve yaban arıları gibi sosyal özellik gösteren böceklerin yanı sıra kuş ve balık sürüleri gibi bazı hayvan topluluklarının kolektif davranışlarından esinlenilerek tasarlanmış algoritmalardır.

Sosyal böcek kolonileri uzun yıllardan beri araştırmacıların yoğun ilgisini çekmiş ve davranışlarını kontrol eden mekanizmalar uzunca bir zaman bilinmeyen olarak kalmıştır. Bu kolonilerin tekil olarak üyeleri karmaşık olmayan bireyler olmasına rağmen, işbirliği içinde karmaşık görevleri başarabilmektedirler. Koordineli koloni davranışı oldukça basit eylemlerden veya kolonilerin bireysel üyeleri arasındaki etkileşimlerden oluşur. Sosyal böceklerin kolektif faaliyetleri birçok yönüyle merkezi bir kontrol olmadan kendi kendine organize ederek yürütülen bir çalışmadır. Örneğin yaprak-kesen karıncalar yapraklardan parçaları koparır, bunları yuvalarına taşır ve larvaları için gıda olarak kullandıkları mantarları yetiştirirler. Dokumacı-karıncalar iki yaprak arasındaki boşluklardan geçmek için vücutlarıyla zincir örerler. Böylece iki yaprağın kenarları birbirlerine doğru çekilir ve ardı ardına olgun bir larva tarafından çıkarılan ipekle yapraklar bağlanır. Termitlerin ve yaban arılarının karmaşık yuvaları inşa etmekte gösterdikleri yetenek veya bal arılarının ve karıncaların yaşadıkları çevrede birbirlerini besin kaynakları ve tehlikelere karşı yönlendirebilmeleri gibi birçok örnek verilebilir (Blum ve Li, 2008).

Terim olarak sürü zekâsı ilk defa Beni (1988) tarafından en yakın komşularıyla etkileşimleri sonucu birbirlerini organize eden basit etmenlerin oluşturduğu hücresel robotik sistemleri açıklamak amacıyla kullanılmıştır. Gelinen noktada sürü zekâsı terimi çok daha geniş bir alan için kullanılmaktadır. Sürü zekâsı teknikleri; endüstride ve bilimde önemli bir konu olan optimizasyon alanında çok başarılıdır.

Genel olarak bir P optimizasyon problemi üç değişkenle (S, Ω, f) açıklanabilir. Birincisi $X_i, i = 1, \dots, n$ karar değişkenleri olmak üzere sonlu bir küme üzerinde tanımlanan S - arama uzayıdır. Karar değişkenlerinin ayrık değerlerden oluşması

durumunda *P ayrık (kombinasyonel) optimizasyon problemi* olarak adlandırılır. Karar değişkenlerinin sürekli değerlerden oluşması durumunda ise *P sürekli optimizasyon problemi* olarak adlandırılır. Karar değişkenlerinin hem ayrık hem de sürekli değerler içerdiği karışık durumlar da olabilir. Karar değişkenleri için tanımlanması gereken kısıtlar, Ω kısıt kümesi ile belirtilir. Son olarak arama uzayındaki çözümlerin kalitesini hesaplamak için bir f amaç fonksiyonu tanımlanır. Optimizasyon problemi ile ulaşılması istenen çözüm üç kategoride sınıflandırılır. Birincisi, arama uzayında amaç fonksiyonunun alabileceği en küçük değere ulaşmaktır ve minimizasyon problemi olarak tanımlanır. İkincisi, arama uzayında amaç fonksiyonunun alabileceği en büyük değere ulaşmaktır ve maksimizasyon problemi olarak tanımlanır. Üçüncü kategori ise özellikle gerçek dünya problemleri ile ilişkilidir. Bu kategoride amaç aynı karar değişkenleri ile oluşturulan birden fazla ve birbiriyle çelişen amaç fonksiyonlarının arama uzayında aynı zamanda alabilecekleri en uygun değerlere ulaşmaktır ve çok-amaçlı optimizasyon olarak tanımlanır (Blum ve Li, 2008).

Bugüne kadar pek çok alanda duyulan ihtiyaçlardan ötürü birçok optimizasyon algoritması geliştirilmiştir. Kombinasyonel optimizasyon bağlamında algoritmalar iki şekilde sınıflandırılır. Bunlar tam (complete) ve yakınsama (approximate) algoritmalarıdır. Tam algoritmalar sonlu boyutta bir kombinasyonel problem için sınırlı bir süre içinde optimal sonucu bulmayı garanti eder. Ancak NP-zor problemler (non-deterministic polynomial hard problems) için bu her zaman mümkün değildir. Bu nedenle tam-metotlar en kötü durumda üstel bir hesaplama zamanına ihtiyaç duyabilir (Kumar, 2010).

Bu durum pratik uygulamalar için çok fazla zaman kaybına neden olur. Sürü zekâsı temelli algoritmalar gibi yakınsama metotları kullanılarak harcanan zaman önemli ölçüde azaltılır ve optimal çözümler her zaman garanti edilemese de kabul edilebilir ölçüde iyi çözümler bulunur. Geçen 30 yıl içinde sezgisel yakınsama algoritmaları çok fazla ilgi çekmiş ve bu algoritmalar üzerinde yoğun olarak çalışılmıştır.

Tez çalışmasının bu bölümünde optimizasyon problemlerinin çözümü için en tanınmış sürü zekâsı tekniklerinden ikisi üzerinde odaklanılmaktadır. Bu teknikler: Yapay Arı Kolonisi (ABC) ve Parçacık Sürü Optimizasyonu (PSO) algoritmalarıdır.

2.1. Yapay Arı Kolonisi

Yapay Arı Kolonisi (ABC), bal arılarının bilgi paylaşımı amacıyla yaptıkları dans ve yiyecek aramak için sürü halinde gösterdikleri zeki davranışlardan esinlenerek geliştirilen sezgisel bir optimizasyon algoritmasıdır. Esinlenilen arı kolonisi modeli üç tür arı grubundan oluşur. Bu arılar: işçi arılar, gözcü arılar ve kâşif arılar olarak adlandırılır. Koloninin ilk yarısını işçi arılar, ikinci yarısını gözcü arılar oluşturur. Arama uzayında her bir besin kaynağı için sadece bir işçi arı görevlendirilir. Dolaysıyla işçi arıların sayısı kovanın etrafında taranan besin kaynağı sayısına eşittir. Kolonideki arılar tarafından besin kaynağı tüketilen işçi arı, kâşif arıya dönüşür ve yeni bir kaynak aramaya başlar. Algoritmanın temel adımları Şekil 2.1'de görülmektedir (Karaboğa, 2005).

```
Kâşif arılar ile ilk besin kaynaklarını oluştur
REPEAT
        İşçi arıları kaynaklarına gönder ve kaynaklardaki yiyecek miktarını belirle
        Kaynakların gözcü arılar tarafından tercih edilme ihtimallerini hesapla
        Gözcü arıları seçtikleri kaynaklara gönder ve yiyecek miktarlarını belirle
        Arıların gitmekten vazgeçtiği kaynağa gidişleri durdur
        Kâşif arıları rasgele yeni kaynak aramak için arama uzayına gönder
        Şimdiye kadar bulunan en iyi çözümü hafızaya al
UNTIL (durdurma ölçütleri sağlanıncaya kadar)
```

Şekil 2.1. ABC algoritmasının temel adımları

ABC algoritmasında arama sürecinin her çevrimi üç aşamadan oluşur. Bu aşamalar sırasıyla önce işçi ve daha sonra gözcü arıların besin kaynaklarına giderek kaynaklardaki yiyecek miktarının belirlenmesi ve tükenen kaynakların belirlenerek bu kaynaklardaki işçi arıların kâşif arıya dönüştürülmeyle olası yeni kaynakların belirlenmesi aşamalarıdır. Kâşif arı sayısı, limit parametresi ile kontrol edilir. Bir kaynağı temsil eden çözüm belli bir sayıda denemeden sonra hala iyileştirilemiyorsa; bu kaynakta besin kalmadığı ve tükendiği varsayılarak terkedilir. Böylece bu kaynağın işçi arısı kâşif arıya dönüşür. Bu amaçla her besin kaynağı için bir geliştirilememe sayacı tutulur.

İlk olarak besin kaynaklarının sayısı, durdurma ölçütü ve limit parametresinin değeri tanımlanır. Her biri arama uzayında olası bir çözümü temsil eden besin

kaynaklarının başlangıç değerleri rasgele bir şekilde denklem (2.1) kullanılarak tanımlanır ve her bir besin kaynağı için bir işçi arı atanır. Kovandaki gözcü arıların sayısı da işçi arıların sayısına eşittir.

$$x_i^j = x_{min}^j + rand(0,1)\left(x_{max}^j - x_{min}^j\right) \text{ ve } i = \{1, 2, \dots, SN\} \tag{2.1}$$

Burada x_i^j kolonideki i.çözümün j.parametresini göstermektedir. SN besin kaynaklarının sayısıdır. $j = \{1, 2, \dots D\}$ olmak üzere D olası bir çözümü oluşturan karar değişkenlerinin sayısıdır. x_{min}^j ve x_{max}^j ise j.karar değişkenin alt ve üst sınır değerleridir. Kolonin başlangıç çözümleri üretildikten sonra işçi arıların uygunluk değerleri denklem (2.2) ile hesaplanır (Karaboğa, 2005; Öztürk, 2011).

$$fit_i = \begin{cases} {}^{1}/_{(1+f_i)} & f_i \geq 0 \\ 1 + abs(f_i) & f_i < 0 \end{cases} \tag{2.2}$$

Burada fit_i kolonideki i.çözümün uygunluk değeri gösterilmektedir. f_i ise i.çözümün amaç fonksiyon değeridir. Çözülecek optimizasyon problemi için özel olarak tanımlanmalıdır.

Bu noktadan sonra algoritmanın çevrimsel süreci başlar. İlk aşama yukarıda bahsedildiği gibi işçi arı fazıdır. Kolonideki tüm işçi arılar sırasıyla hafızalarındaki besin kaynağının komşuluğunda başka bir besin kaynağı seçer ve kendi çözümünü iyileştirmeye çalışır. Komşu çözüm kendisinden farklı olarak koloniden rasgele seçilir. Denklem (2.3) kullanılarak yeni çözüm hesaplanır. Her çözüm için $[1, D]$ aralığında rasgele birer parametre seçilir ve belirlenen komşu çözüm ile delta değeri bu parametre için hesaplanır.

$$delta = \emptyset \times \left(x_i^j - x_k^j\right) \quad \text{ve} \quad v_i^j = x_i^j + delta \tag{2.3}$$
$$i = \{1, 2, \dots, SN\}, i \neq j \quad \text{ve} \quad j \in \{1,2, \dots, D\}$$

Burada $delta$ yeni v_i çözümünün seçilen j. parametresi için hesaplanan değişim miktarıdır. $\emptyset$ parametresi $[-1, 1]$ aralığında rasgele üretilen bir reel sayıdır. x_k ile çözümün komşu çözümü gösterilir ve kolonideki k. çözümdür. İşçi arıların komşularıyla yeni çözümler üretmesi sonucunda karar değişkenleri önceden belirlenen aralıkların dışına çıkabilir. Böyle bir durumda denklem (2.4) kullanılarak sınır değerleri yeni çözüm için ayarlanır.

$$v_i^j = \begin{cases} x_{min}^j & v_i^j < x_{min}^j \\ v_i^j & x_{min}^j \leq v_i^j \leq x_{max}^j \\ x_{max}^j & v_i^j > x_{max}^j \end{cases} \quad (2.4)$$

Yeni üretilen çözümlerin uygunluk değerleri denklem (2.2) ile hesaplanır. Burada eski ve üretilen çözüm arasında aç-gözlü bir seçim işlemi uygulanır. Eğer yeni üretilen çözüm eskisinden daha iyi ise yeni çözüm hafızaya alınır ve iyileştirilememe sayacı sıfırlanır. Aksi halde besin kaynağının iyileştirilememe sayacı bir arttırılır.

İşçi arılar arama işini tamamlayıp kovana döndüklerinde yaptıkları danslarla besin kaynaklarının kalitesiyle ilgili bilgiyi paylaşırlar. Dansı izleyip kaynak seçimini buna göre yapan arılara gözcü arılar denir. Gözcü arılar kaynakları besin miktarlarına bağlı olarak denklem (2.5) ile verilen olasılık değerine göre seçerler. Burada uygunluğa bağlı seçim işlemi için rulet tekerleği, oransal seçim, turnuva seçimi vb. herhangi bir metot kullanılabilir. Temel ABC algoritmasında, uygunluk değerleri ile orantılı dilim boyutlarına göre seçim işleminin yapıldığı bir rulet tekerliği kullanılır (Karaboğa, 2005; Öztürk, 2011).

$$p_i = \frac{fit_i}{\sum_{n=1}^{SN} fit_i} \quad (2.5)$$

Bu olasılıklı seçim planı sayesinde uygunluk değeri yüksek olan çözümlerin ziyaret edilme sıklığı arttırılmış olur. Böylece iyi çözümlerle arama işlemine pozitif bir geri besleme sağlanmış olur.

İkinci aşama olan Gözcü arı fazına geçilir. Burada tüm kaynaklar için [0,1] aralığında rasgele bir değer üretilir. Eğer bu kaynağın hesaplanmış olasılık değeri rasgele üretilen bu sayıdan büyükse, işçi arı aşamasında olduğu gibi, gözcü arı denklem (2.3) ile besin kaynağının pozisyonunda değişikliğe gider. Kaynağın yeni uygunluk değeri hesaplanır. Aç-gözlü seçim işlemine göre yeni kaynak daha iyi ise gözcü arı eskisini unutur ve yenisini hafızaya alır ya da eskisini saklamaya devam eder. Bu durumda iyileştirilememe sayacı bir attırılır. Bu işlemler tüm gözcü arılar için tekrarlanır.

İşçi ve gözcü arı fazları tamamlandıktan sonra, algoritmanın bir çevrimindeki son aşama olan kâşif arı fazına geçilir. Bu aşamada terkedilecek herhangi bir kaynak olup olmadığı kontrol edilir. Buna karar vermek için kaynakların iyileştirilememe

sayaçları limit parametresi ile karşılaştırılır. Eğer sayaç değeri limitten büyükse bu kaynak tükenmiş anlamına gelir. Bu durumda kaynakta görevli olan işçi arı kaynağı terk eder ve kâşif arıya dönüşerek yeni kaynak arar. Tükenen kaynaklar yerine denklem (2.1) kullanılarak yeni çözüm üretilir. Böylece arama sürecine negatif geri besleme yapan ve dalgalanmalara neden olan kötü çözümlerin atılması sağlanır.

Algoritma sonlanma koşulu sağlanıncaya kadar bu üç fazı tekrarlamaya devam eder. Sezgisel algoritmalarda bu amaçla kullanılan iki yaklaşım vardır. Birincisi maksimum çevrim sayısı, ikincisi ise maksimum uygunluk fonksiyonu değerlendirme sayısıdır (*FEs*). ABC algoritmasının önerilen temel sürümünde sonlanma koşulu olarak maksimum çevrim sayısı (MNC) kullanılır. Farklı algoritmaların karşılaştırılmasında bu farklığın üstesinden gelmek için $MNC = FEs/NP$ işlemi yapılır. Ancak bu işlem her çevrimde bir fonksiyon değerlendirmesi yapan algoritmalar için geçerlidir. ABC algoritması tek çevrim içinde işçi ve gözcü arı fazlarında ayrı ayrı fonksiyon değerlendirmeleri yapar. ABC algoritmasının yalnız çevrim sayısına göre çalıştırılması bir eksiklik olarak görülebilir. Bu eksikliği kapatmak amacıyla Mernik ve ark. (2015) ABC'ye maksimum fonksiyon değerlendirme sayısına göre sonlanma biçimini eklediler. ABC algoritmasının çalışma adımlarına bakıldığında kullanılan iki adet kontrol parametresi daha vardır. Bunlar işçi ve gözcü arıların sayısına eşit olarak ayarlanan koloni boyutu SN, diğer ise limit parametresidir.

Temel ABC algoritması üzerinde şimdiye kadar farklı araştırmacılar tarafından çeşitli değişiklikler önerilmiştir. Burada temel algoritmanın ana hatlarının dışına çıkmadan sadece geliştiricilerinin önerdiği bazı parametreler üzerinde durulacaktır. Algoritmanın mevcut kontrol parametrelerine ek olarak iki kontrol parametresi daha önerdiler (Akay ve Karaboğa, 2012). Kısıtlı-optimizasyon problemlerinde algoritmanın kimi zaman çözüme yavaş yakınsamasından dolayı modifikasyon oranı (MR) ve ölçekleme faktörü (SF) parametreleri eklendi.

Modifikasyon Oranı (MR): Temel ABC algoritmasında yeni bir çözüm üretilirken çözümün sadece bir parametresi değiştirilir. Bu da yakınsamanın yavaş olmasına neden olabilir. Bu frekans düzensizliğini düzeltmek amacıyla MR parametresi eklenmiştir. Bu parametre ile her bir x_i^j parametresi için $[0,1]$ aralığında rasgele üretilen bir R_i^j reel sayısı üretilir ve denklem (2.6) çözümün tüm parametreleri için kullanılarak yeni çözüm üretilir.

$$v_i^j = \begin{cases} x_i^j + \emptyset.\left(x_i^j - x_k^j\right) & R_i^j < MR \\ x_i^j & R_i^j \geq MR \end{cases} \quad (2.6)$$

Küçük bir *MR* değeri çözümün daha yavaş iyileşmesine neden olurken, büyük bir *MR* değeri ise çözümde çok fazla çeşitliliğe neden olabilir.

Ölçekleme Faktörü (SF): Bu parametre delta değişkeniyle ilgili değişiklik öngörür. Temel ABC algoritmasında delta değişkeni yerel minimumlara takılmayı önlemek amacıyla mevcut çözüme önceki çözümün $[-1,1]$ aralığında rasgele bir oranda ($\emptyset$) eklenmesini sağlar. Bu düzensizliğin büyüklüğünü kontrol edebilmek amacıyla $[-SF, SF]$ aralığında seçilen bir değişken önerilmiştir. Bu değer algoritma çalışmaya başlamadan önce ayarlanır. Küçük bir *SF* değeri hassas bir ayarlama sağlamasının yanında yavaş yakınsamaya neden olur. Büyük bir SF değerinin seçilmesi ise yakınsama hızını arttırabilirken, yeni çözümler bulma yeteneğini kısıtlayabilir. Bu sıkıntılardan dolayı uygun *SF* değerinin belirlenmesini otomatikleştiren bir yaklaşım önerilmiş ve *Adaptif SF* (*ASF*) olarak adlandırılmıştır. *ASF* Rechennberg'in 1/5 mutasyon kuralına göre denklem (2.7) kullanılarak her m adet çevrimde bir hesaplanarak güncellenir (Akay ve Karaboğa, 2012).

$$SF(t+1) = \begin{cases} SF(t) * 0.85 & \phi(m) < 1/5 \\ SF(t)/0.85 & \phi(m) > 1/5 \\ SF(t) & \phi(m) = 1/5 \end{cases} \quad (2.7)$$

2.2. Parçacık Sürü Optimizasyonu

Parçacık Sürü Optimizasyonu (PSO) popülasyon temelli sezgisel bir optimizasyon tekniğidir. Kuş ve balık sürülerinin sosyal davranışlarından esinlenilerek geliştirilmiştir. PSO algoritması evrimsel hesaplama teknikleri ile benzerlikler gösterir. Sistem rasgele üretilen aday çözümlerden oluşan bir popülasyonla başlatılır ve en iyi çözüm için iteratif olarak güncellemeler yapar. PSO algoritmasında çaprazlama ve mutasyon gibi evrimsel operatörler yoktur. Parçacık olarak adlandırılan aday çözümler, mevcut global ve yerel en iyi çözümleri takip ederek arama uzayında dolaşırlar. Esinlendiği analoji ile ifade edilirse parçacıklar arama uzayında uçarak besin kaynaklarını arar. PSO; fonksiyon optimizasyonu, yapay sinir ağları eğitimi, bulanık

sistem kontrolü ve sezgisel optimizasyon problemlerinin uygulanabildiği tüm alanlarda başarılı bir şekilde kullanılmaktadır (Kennedy ve Eberhart, 1995).

PSO algoritmasının uygulanması kolaydır ve az sayıda ayarlanması gereken parametresi vardır. Sürüdeki tüm parçacıkların, optimize edilecek uygunluk fonksiyonu tarafından değerlendirilen bir uygunluk değeri ve uçuşlarını yönlendiren hız bilgileri vardır. Parçacıklar problem uzayında mevcut en iyi parçacıkları takip ederek arama işlemini gerçekleştirirler.

Her çevrimde tüm parçacıkların konumu iki "en iyi" değere göre yeniden hesaplanır. Bunlardan birincisi bir parçacığın o ana kadar bulduğu en iyi uygunluk değeridir. Ayrıca bu değer daha sonra kullanılmak üzere hafıza tutulur ve "$pbest$" yani parçacığın en iyi değeri olarak isimlendirilir. Diğer en iyi değer ise popülâsyondaki herhangi bir parçacık tarafından o ana kadar elde edilmiş en iyi uygunluk değerine sahip çözümdür. Bu değer popülasyon için global en iyi değerdir ve "$gbest$" olarak isimlendirilir (Kennedy ve Eberhart, 1995).

D adet parametreden oluşan n adet parçacık için popülasyon matrisi denklem (2.8)'de verilen matris ile ifade edilir. Matrise göre, $i.$ parçacık $x_i = [x_{i,1}, x_{i,2}, \dots, x_{i,D}]$ ve $pbest_i = [p_i, p_2, \dots, x_D]$, $gbest_i = [p_i, p_2, \dots, x_D]$ şeklinde gösterilir.

$$x = \begin{bmatrix} a_{11} & \dots & a_{1D} \\ \vdots & \ddots & \vdots \\ a_{n1} & \dots & a_{nD} \end{bmatrix}_{n \times D} \tag{2.8}$$

$i.$ parçacığın her parametresindeki değişim miktarını göster hız vektörü $v_i = [v_{i,1}, v_{i,2}, \dots, v_{i,D}]$ olarak ifade edilir. Bu iki en iyi değer bulunduktan sonra; parçacık hızı denklem (2.9) ile hesaplanır. Parçacığın yeni konumu ise denklem (2.10) ile hesaplanır.

$$v_i^{k+1} = v_i^k + c_1.rand_1^k.\left(pbest_i^k - x_i^k\right) + c_2.rand_2^k.\left(gbest^k - x_i^k\right) \tag{2.9}$$

$$x_i^{k+1} = x_i^k + v_i^{k+1} \tag{2.10}$$

Denklemde $rand$ ifadesi [0,1] aralığında rasgele üretilen bir değeri, i sürüdeki parçacık indeksini, k ise iterasyon sayısını gösterir. c_1 ve c_2öğrenme katsayılarıdır. Bu faktörler parçacıkları $pbest$ ve $gbest$ konumlarına doğru yönlendiren sabitlerdir. c_1 katsayısı parçacığın kendi deneyimlerine göre; c_2 katsayısı ise sürüdeki diğer parçacıkların deneyimlerine göre hareketi yönlendirir. Düşük değerler seçilmesi

parçacıkların hedef bölgeye doğru çekilmeden önce, bu bölgeden uzak yerlerde dolaşmalarına imkân verir. Ancak hedefe ulaşma süresi uzayabilir. Diğer yandan, yüksek değerler seçilmesi, hedefe ulaşmayı hızlandırırken, beklenmedik hareketlerin oluşmasına ve hedef bölgenin es geçilmesine sebep olabilir. Genellikle $c_1 = c_2 = 2$ olarak almanın iyi sonuçlar verdiği belirtilmiştir. PSO algoritmasının sözde kodu Şekil (2.2)'de verilmiştir.

For her parçacık için
    Parçacığı başlangıç konumuna getir
End

Do
  For her parçacık için
    Uygunluk değerini hesapla
    Eğer uygunluk değeri pbest ten daha iyi **ise,**
      Şimdiki değeri yeni pbest olarak ayarla
  End

  Tüm parçacıkların bulduğu pbest değerlerinin en iyisini, tüm parçacıkların gbest'i olarak ayarla

  For her parçacık için
    Denklem (2.9)'e göre parçacık hızını hesapla
    Denklem (2.10)'ye göre parçacık konumunu güncelle
  End
While maksimum iterasyon sayısına veya minimum hata koşulu sağlanana kadar devem et.

Şekil 2.2. PSO algoritmasının sözde-kodu

PSO'nun avantajlarından birisi reel sayılarla çalışıyor olmasıdır. Genetik algoritmalardaki gibi hesaplama yapabilmek için ikili kodlamadan dönüştürme yapılması ya da bazı özel kullanılması zorunlu operatörlere ihtiyaç duyulmaz. Sonlanma kriteri olarak maksimum çevrim sayısı veya minimum hata koşulu kullanılabilir. PSO kontrol parametreleri; parçacık sayısı, parçacık boyutu, parçacık aralığı, *vmax*, öğrenme faktörleri ve sonlanma şartı olarak sayılabilir. Parçacık sayısı; genellikle 20 ila 40 arasında alınır. Aslında çoğu problem için sayıyı 10 almak iyi çözümler elde etmek için yeterlidir. Bazı zor veya özel problemler için 100 veya 200 parçacık kullanılması gerekebilir. Parçacık boyutu; probleme göre değişebilir. Parçacık aralığı; probleme göre değişmekle birlikte farklı boyutlarda ve aralıklarda parçacıklar tanımlanabilir. $vmax$ bir iterasyonda, bir parçacıkta oluşabilecek maksimum değişikliği belirler. Genellikle parçacık aralığına göre belirlenir. Örneğin X1 parçacığı (-10,10) aralığında ise $vmax$ 20

değeri ile sınırlandırılabilir. Öğrenme faktörleri c_1 ve c_2 genellikle 2 olarak ayarlanır. [0,4] aralığında değiştirilebilir. Sonlanma koşulu, maksimum çevrim sayısı veya uygunluk fonksiyonu istenilen seviyeye ulaştığında algoritma durdurulabilir.

Algoritmayı öneren araştırmacılar 1998 yılında yönteme bir kısıtlama katsayısı ekleyerek standart PSO olarak adlandırdıkları yeni versiyonu yayınladılar (Shi ve Eberhart, 1998). Standart PSO algoritmasında yeni hız değişkeni denklem (2.11)'e göre hesaplanır.

$$v_i^{k+1} = \omega . v_i^k + c_1 . rand_1^k . \left(pbest_i^k - x_i^k\right) + c_2 . rand_2^k . \left(gbest^k - x_i^k\right) \quad (2.11)$$

$$x_i^{k+1} = x_i^k + v_i^{k+1}$$

Eylemsizlik ağılığı (*inertia weight*) ω optimizasyon sürecinde önceki parçacık hızlarının etkisini ayarlayarak $vmax$ parametresinin yerini almak için tasarlanmıştır. Büyük seçilen değerler parçacığın keşif yeteneğini arttırır. Küçük seçilen değerler parçacığın daha hassas arama yapabilmesini sağlar. Bunun yanı sıra küçük değerler keşif özelliğini olumsuz etkileyerek yerel ekstremumlara takılmaya neden olabilir. Değeri probleme göre değişkenlik gösterdiği için dinamik değer belirleme yöntemleri de önerilmiştir (Shi ve Eberthart, 2001). ω ilk iterasyonlar için birden büyük bir değer ile başlatılır ve iterasyon sayısıyla ters orantılı olarak değeri birden daha küçük bir değere doğru azaltılarak kullanılır.

2.3. Bölüm Değerlendirmesi

Bu bölümde, sürü zekâsı kavramı tanımlanmış ve bu alanda özellikle optimizasyon problemlerinin çözümüne yönelik olarak geliştirilen başlıca iki algoritma detaylı bir şekilde anlatılmıştır. Yapılan araştırmalar göstermiştir ki, popülasyon temelli sezgisel algoritmalar içinde optimizasyon problemlerinin çözümü başta olmak üzere son 15 yıl içinde çok çeşitli alanlarda uygulanan sürü zekâsı algoritmaları kendi başına başlıca yeni bir çalışma alanı olarak literatürde yer bulmuştur. Bu algoritmalar için önerilmiş çok fazla sayıda modifikasyon çalışması yayınlanmıştır. Bununla birlikte, performans iyileştirme amacıyla önerilen bu modifikasyonlar çoğunlukla belirli bir uygulama alanına yönelik olduğu için genel olarak algoritmanın orijinali ile her zaman rekabet edebilir düzeyde olamamaktadır. Bu nedenle çalışma kapsamında ele alınan algoritmalar önerildikleri orijinal versiyonları temel alınarak açıklanmış ve sadece geliştiricilerinin önerdiği birkaç modifikasyon üzerinde durulmuştur.

3. GÖRÜNTÜ SEGMENTASYONU

Görüntü segmentasyonu, görüntü işlemenin ön hazırlık evrelerinde kullanılan düşük-seviyeli en zor işlemlerden biridir. Bu işlem birçok görüntü analizi, makine-görüşü ve örüntü tanıma uygulamasının en temel ve kritik öneme sahip olan adımıdır. Segmentasyonun performansı işlenecek sayısal görüntünün kalitesiyle doğrudan ilişkilidir ve görüntü kalitesi; çevresel etmenler, görüntüleme cihazları ve gün ışığı gibi birçok değişkenden etkilenir. Görüntü segmentasyonu, belirli bir uygulama için anlamlı olan ayrık ve homojen bölgelerin görüntüden çıkarılması işlemidir. Homojen bölge; yoğunluk, renk, kenar, doku, paralellik, süreklilik, kapalılık, yakınlık vb. bazı bir veya daha çok özellik bakımından benzerlik gösteren piksel grupları şeklinde tanımlanır. Algoritmaların başarısı büyük oranda homojenliği tespit etmek için kullanılan bu benzerlik ölçütüne ve görüntüden elde edilen ayırt edici özniteliklerin belirlenmesine dayanır.

Görüntü segmentasyonu genel bir matematiksel ifade ile denklem (3.1)'deki gibi açıklanabilir (Pal ve ark. 1993). Homojen bir bölgeyi oluşturan pikseller için P'nin bir homojenlik fonksiyonu olduğu varsayılsın. Eğer $P()$ bağlı piksel gruplarını tanımlayan bir homojenlik fonksiyonu ise, tüm bölgeler için $P(S_i) = true$ değerini alır. $i \neq j$ ve S_i ile S_j komşu bölgeler olmak üzere $P(S_i \cup S_j) = false$ değerini alır.

$$\bigcup_{i=1}^{n} S_i = F$$
$$S_i \cap S_j = \emptyset \qquad (i \neq j) \tag{3.1}$$
$$F = \{S_1, S_2, \dots, S_n\}$$

Tüm görüntülere kesin bir doğrulukla uygulanabilen bir segmentasyon algoritması geliştirmek çok zor bir iştir. Fu ve Mui (1981)'ye göre görüntü segmentasyonu temel olarak psiko-fiziksel bir algı problemidir. Bu nedenle tam olarak analitik bir çözüme duyarlı değildir. Geçmişte farklı türlerde birçok tek-renkli segmentasyon tekniği geliştirilmiştir. Renkli görüntü segmentasyonu ise bazı nedenlerden dolayı çok daha dikkat çekicidir. Bu nedenler iki maddeyle özetlenebilir: (1) renkli görüntülerin boyutu gereği gri-seviyeli görüntülerden daha fazla bilgi içermesi ve (2) son zamanlarda bilgisayar teknolojilerindeki ilerlemeye paralel olarak kişisel bilgisayarların görüntü işlemedeki performanslarının artmış olması.

İnsan gözü binlerce ton ve yoğunlukta rengi ayırt edebilmesine rağmen yalnızca iki düzine gri tonu algılayabilmektedir. Bu da insanın görüntülerdeki nesneleri tam olarak çıkarabilmek için yalnızca gri-tonları kullanmadığını ve renk bilgisinin önemini göstermektedir. Renk ve doku bilgisi görüntülerde içerik tabanlı bilgi ediniminde kullanılan en önemli düşük-seviye özelliklerdir. Bu anlamda renk bilgisi örüntü tanıma ve bilgisayar görüşü için çok önemlidir. Renkli görüntü segmentasyonu alanındaki çalışmalar, tek renkli görüntü segmentasyonu kadar geniş değildir. Bu alanda yayılanmış olan çalışmaların büyük çoğunluğu da gri-seviye görüntü segmentasyon yaklaşımlarına dayanmaktadır (Gonzales ve Woods, 2002).

Bu bölümde segmentasyon tekniklerinin sınıflandırılması ve başlıca kullanılan yöntemler açıklandıktan sonra, renk uzayları hakkında bilgi verilmekte ve son olarak bölüm değerlendirmesi yapılmaktadır.

3.1. Renk Uzayları

Bir renk uzayı, renklerin uzayda geometrik olarak temsil edilmesini sağlar. Genellikle üç bileşenin sayısal değerleri kullanılarak renk bilgisi tanımlanır. Bu bileşenler yardımıyla renkleri nitelemek mümkün olur. Kullanılan birçok uzay renkler arasındaki farkı gösteren bir ölçüt tanımlar. Renkli görüntü analizi çerçevesinde en klasik renk uzayları karakteristiklerine göre dört grupta sınıflandırılabilir (Busin ve ark, 2008).

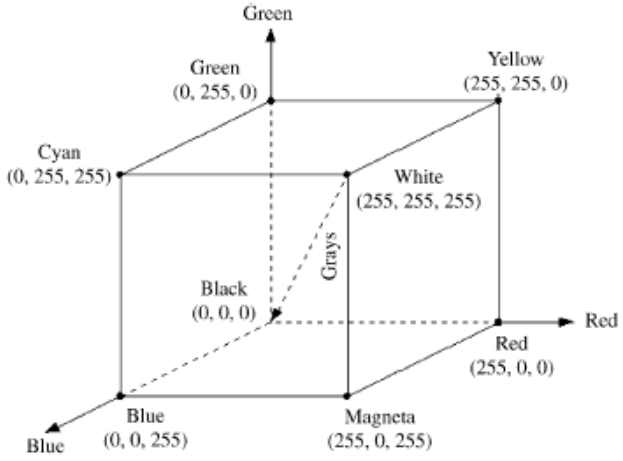
- *Birincil Renk Uzayları;* üç-renklilik (trikromatik) teorisine dayanır ve tüm renklerin üç birincil rengin uygun miktarlarda karıştırılmasıyla temsil edilmesinin mümkün olduğu varsayılır.
- *Parlaklık – Renklilik Renk Uzaylarında;* bileşenlerden biri parlaklığı diğer ikisi renkliliği ifade eder.
- *Algısal Renk Uzayları;* yoğunluk, ton ve doygunluk bileşenlerini kullanılarak öznel insan renk algısını ölçmeye çalışılır.
- *Bağımsız Eksen Uzayları;* mümkün olduğunca renk bileşenlerine bağımlılığın azalmasını sağlayan farklı istatistiksel metotlar kullanılarak elde edilen uzaylardır.

Renkli görüntü analiz uygulamaları için uygun renk uzayının seçilmesi zor bir iştir. Çünkü birçok farklı uygulama için geliştirilmiş ve farklı şekillerde tanımlanmış

çok çeşitli renk uzayları vardır. Örneğin $RGB, HSI, CIE\ L*u*v*$ gibi renkli görüntü segmentasyonunda kullanılan birçok renk uzayı vardır. Ancak bu uzayların hiçbiri her türlü görüntü için diğerine karşı üstün gelebilmiş değildir. Bu bağlamda; en uygun renk uzayını seçmek renkli görüntü segmentasyonunda halen önemini koruyan zorluklardan biridir (Gauch ve Hsia, 1992; Cheng ve ark., 2001). Bu bölümün devamında renkli görüntü segmentasyonunda yaygın olarak kullanılan bazı renk uzayları açıklanmaktadır.

3.1.1. RGB Renk Uzayı

RGB renk uzayı birincil renk uzaylarının başında gelir. İnsan gözünün renkleri algılama işlemini matematiksel olarak modelleyebilmek için geliştirilen tristimulus teorisine göre hesaplanan kırmızı (R), yeşil (G) ve mavi (B) renk koordinatlarının birleştirilmesinden oluşur. RGB renk uzayının geometrik olarak 3-boyutlu bir küp içindeki gösterimi Şekil 3.1.'de verilmiştir.



Şekil 3.1. 3-boyutlu bir küp içinde RGB renk uzayı gösterimi (Cheng ve ark. 2001)

Tüm renkler bu üç renk ile oluşturulur ve karışım oranlarına göre tekil renkler elde edilir. RGB uzayı, televizyon sistemlerinde ve sayısal kamerayla elde edilen resimleri görüntülemek için en çok kullanılan modeldir. RGB, renkleri görüntülemek için uygun olmasına rağmen; görüntü segmentasyonu ve analizi için elverişli değildir. Çünkü R, G, ve B bileşenleri arasında yüksek bir korelasyon vardır. Bunun anlamı yoğunluk değişimlerine karşı, tüm bileşenlerin benzer şekilde değişim göstermesidir. Dolayısıyla RGB uzayındaki bir rengin diğer renklerle olan farklılığını hesaplamak mümkün güçleşir (Littmann ve Ritter, 1997).

3.1.2. CIE Renk Uzayı Ailesi

CIE renk uzayları algısal benzerliği göstermek için geliştirilmiştir. X, Y ve Z şeklinde üç bileşenle gösterilir. Tüm renkler bu üç bileşenin birleşimiyle belirlenebilir. RGB uzayından doğrusal bir dönüşümle denklem (3.2)'deki gibi hesaplanır (Cheng ve ark., 2001).

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} 0.607 & 0.174 & 0.200 \\ 0.299 & 0.587 & 0.114 \\ 0.000 & 0.066 & 0.116 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} \tag{3.2}$$

XYZ tristimulus koordinatları bilindikten sonra bir dizi CIE uzayı oluşturulabilir. $CIE\ (L*a*b*)$ ve $CIE\ (L*u*v*)$ uzayları en tipik örnekleridir. Bu uzayların tamamı X, Y ve Z değerlerinin lineer olmayan dönüşümleriyle hesaplanabilir.

$CIE\ (L*a*b*)$ uzayı denklem (3.3) ile tanımlanır.

$$L^* = 116\left(\sqrt[3]{\frac{Y}{Y_O}}\right) - 16,$$

$$a^* = 500\left(\sqrt[3]{\frac{X}{X_O}} - \sqrt[3]{\frac{Y}{Y_O}}\right),$$

$$b^* = 200\left(\sqrt[3]{\frac{Y}{Y_O}} - \sqrt[3]{\frac{Z}{Z_O}}\right),$$

$L*$ bileşeni rengin parlaklık veya yoğunluk olarak ifade edilen özelliğini gösterirken; $a*b*$ ile $u*v*$ bileşenleri renkliliği ifade eder. Bu uzaylarda tanımlanan bir renk, üç-boyutlu bir uzayda tanımlanmış bir nokta gibi kabul edilebilir. Böylece bu uzaylarda tanımlanmış iki renk arasındaki farklılık Öklid mesafesi ile hesaplanabilir. $CIE\ (L*a*b*)$ uzayı için iki renk arasındaki fark denklem (3.4) ile hesaplanır.

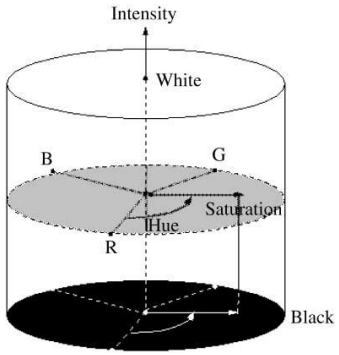
$$E_{ab} = \sqrt{(\Delta L^*)^2 + (\Delta a^*)^2 + (\Delta b^*)^2} \tag{3.4}$$

Renk bilgisindeki algısal farklılığı Öklid mesafesi ile açıklayabilme yeteneği renkli görüntü segmentasyonu için çok önemli bir özelliktir. CIE uzayları renk ve yoğunluk bilgilerini RGB'nin birincil renk bileşenlerinden daha bağımsız ve daha kolay bir şekilde kontrol edebilir. Doğrudan renk karşılaştırması yapılabilir. Bu sayede özellikle küçük renk değişimlerini kolayca algılayabilir. Diğer doğrusal olmayan sistemlerdeki gibi; renk bileşenlerindeki küçük farklılıklar, dönüştürülen değerlerde büyük atlamalara neden olabilir. Buna tekillik problemi denir (Cheng ve ark., 2001).

3.1.3. HSI Renk Uzayı Ailesi

HSI (Hue-Saturation-Intensity), görüntü işlemede yaygın bir şekilde kullanılan bir diğer renk uzayıdır. İnsan algı sistemini modellemeye çalışır. Hue rengin tonunu, saturation rengin doygunluğunu ve intensity rengin yoğunluk bilgisini gösterir. HSI renk uzayının aynı zamanda HSB (hue-saturation-brigthness), HSL(hue-saturation-lightness) ve HSV(hue-saturation-value) gibi bazı çeşitleri vardır. HSI sistemi bir görüntünün yoğunluk bilgisinden rengini ayırt edebilir. Renk bilgisi, ton ve doygunluk değerleri ile gösterilirken; ışık miktarıyla belirlenen görüntünün parlaklığı yoğunluk değeri ile açıklanır (Cheng ve ark., 2001; Kim ve ark., 1996).

HSI renk uzayı geometrik olarak Şekil (3.2)'deki gibi açıklanabilir. Ton değeri genellikle, RGB uzayındaki renk noktasıyla referans çizgisi arasındaki açı olarak kabul edilir. Doygunluk bileşeni silindir merkezinden radyan uzaklığı gösterir. Merkeze en yakın nokta en parlak rengi temsil eder. Yoğunluk bileşeni ise eksen doğrultusundaki yüksekliktir. Silindir ekseni rengin gri-seviye değerini ifade eder.



Şekil 3.2. HSI renk uzayı (Cheng ve ark., 2001)

Farklı yoğunluk veya doygunluk değerlerinin algılanması farklı renkleri tanımak anlamına gelmez. Farklı tonları da kolayca ayırabildiği için HSI renk sistemi, insan renk algısını çok iyi modelleyebilmektedir. HSI koordinatları RGB uzayından denklem (3.5) ile dönüştürülür.

$$H = arctan\left(\frac{\sqrt{3}(G-B)}{(R-G)+(R-B)}\right),$$

$$Int = \frac{(R+G+B)}{3}, \tag{3.5}$$

$$Sat = 1 - \frac{min(R,G,B)}{I}$$

Gri-seviyeli algoritmalar HSI uzayının yoğunluk bileşenine doğrudan uygulanabilir. Farklı renkleri ayrıştırmak için segmentasyon algoritmalarında sadece ton bileşenini kullanmak bile yeterli olabilir. Ton değeri yoğunluktan bağımsız olduğu için özellikle düzgün ışıklandırılmamış gölgeli görüntülerde etkili şekilde kullanılabilir. En önemli dezavantajlarından birisi R, G ve B'nin ufak farklarla ayrıldığı renk silindirinin ekseninin yanında tekilliğin kaldırılamamasıdır. Bu da dönüştürülen ton değerlerinin renk değerlerine göre çok fazla sıçrama yapmasına neden olur ve renklerde istenmeyen süreksizliğe yok açar.

3.1.4. YUV Renk Uzayı Ailesi

YUV, YIQ, YCbCr, YCC, YES renk uzayları televizyon yayınlarında kullanılır. YIQ ve YUV; NTSC ve PAL sistemlerinde kullanılan analog modellerdir. YCbCr modeli ise sayısal bir standarttır. İnsan gözü RGB görüntüler ve siyah-beyaz olmak üzere iki tür görüntü biçimini algılayabildiği için YUV renk uzayı bu iki görüntü biçimini televizyon sistemlerine uyumlu hale getirmek için geliştirilmiştir. Bu renk uzayları sıkıştırma uygulamalarında da çokça kullanılmaktadır (Noor ve ark., 2012; Cheng ve ark., 2001).

YUV Renk Uzayı: Y bileşeni parlaklığı, U ve V bileşenleri renkliliği ifade eder. Avrupa TV sisteminde kullanılan renk kodlama sistemidir. Denklem (3.6)'de verilen doğrusal dönüşümle RGB uzayından elde edilir.

$$\begin{pmatrix} Y \\ U \\ V \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & 0.437 \\ 0.615 & -0.515 & -0.100 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} \tag{3.6}$$

$$burada \quad 0 \leq R \leq 1, \qquad 0 \leq G \leq 1, \qquad 0 \leq B \leq 1.$$

YIQ Renk Uzayı: Amerikan TV sistemlerinde renk bilgisini kodlamak için kullanılır. Y bileşeni rengin parlaklığını ifade eder ve renkli bir görüntüde kenar tespiti için kullanılması uygundur. I ve Q bileşenleri görüntünün ton ve doygunluğunu gösterir. Bu renk uzayı kısmen RGB'deki korelasyon problemini ortadan kaldırır. Doğrusal bir dönüşümle oluşturuldukları için doğrusal olmayan dönüşümlere göre daha hızlıdırlar.

Dolayısıyla YIQ uzayı, doğrusal olmayan sistemler için tercih edilebilir. Denklem (3.7)'de verilen doğrusal dönüşümle RGB uzayından elde edilir.

$$\begin{pmatrix} Y \\ I \\ Q \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.274 & 0.322 \\ 0.211 & -0.253 & -0.312 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} \tag{3.7}$$

$$burada \quad 0 \leq R \leq 1, \qquad 0 \leq G \leq 1, \qquad 0 \leq B \leq 1.$$

YCbCr Renk Uzayı: Sayısal video görüntüleri için ITU tarafından tanımlanan bir standarttır. Avrupa TV sistemlerinde kullanılır ve RGB olmayan sinyal biçimlerini kodlamak için geliştirilmiştir. Y bileşeni parlaklığı gösterirken, Cb ve Cr renklilik bileşenleri sırasıyla referans değer ile mavi ve kırmızının farkını gösterir. YCbCr mutlak bir renk uzayı değildir. Denklem (3.8)'de verilen lineer dönüşümle RGB uzayından elde edilir.

$$\begin{pmatrix} Y \\ Cb \\ Cr \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ -0.169 & -0.331 & 0.500 \\ 0.500 & -0.419 & -0.081 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} \tag{3.8}$$

3.2. Segmentasyon Yöntemlerinin Sınıflandırılması

Segmentasyonun amacının, çalışmanın amacına ve görüntü türüne göre değişkenlik göstermesi çok çeşitli segmentasyon tekniklerinin önerilmesine neden olmuştur. Bu nedenle algoritmaların çok farklı şekillerde sınıflandırılmaları mümkündür. Rogowska (2000)'nın sınıflandırma şemasına göre görüntü segmentasyon teknikleri aşağıdaki gibi farklı kategorilerde açıklanabilir:

- Elle yapılan, yarı otomatik ve otomatik segmentasyon algoritmaları,
- Piksel-tabanlı ve bölge-tabanlı segmentasyon algoritmaları,
- Elle çizim, düşük-seviye segmentasyon (eşikleme, bölge büyütme, vb.) ve model-tabanlı segmentasyon (özellik haritalama teknikleri, dinamik programlama, çizgi izleme, vb.)
- Klasik (eşikleme, kenar-tabanlı ve bölge tabanlı teknikler), istatistiksel, bulanık ve yapay sinir ağları tekniklerine dayalı segmentasyon algoritmaları

En çok kullanılan segmentasyon teknikleri eşikleme-tabanlı, kenar-tabanlı, bölge-tabanlı ve kümeleme-tabanlı segmentasyon olarak sıralanabilir. Eşikleme tabanlı segmentasyon yöntemleri; görüntüyü parçalarına ayırmak için histogram-eşikleme ve

dilimleme tekniklerini kullanır. Bu yöntemler doğrudan segmentasyon amacıyla bir görüntüye uygulanabileceği gibi ön veya son işleme aşamalarında yardımcı süreçler olarak da kullanılmaktadır. Kenar tabanlı teknikler görüntüdeki kenarları tespit ederek nesnelerin sınırlarını belirlemeye çalışır. Bölge tabanlı segmentasyon ise kenar-tabanlı yöntemlerin aksine tespit edilecek nesnenin kenarlarından değil içinden başlayarak nesneye ait olan tüm piksellerin bulunmasına dayanır. Kümeleme-tabanlı segmentasyon teknikleri görüntünün piksellerini ve piksel düzeyinde elde edilen özellik kümesini bir arama uzayı olarak ele alır. Bu uzayda benzer özellik gösteren pikselleri yüksek-boyutlu veri analizi amacıyla kullanılan kümeleme algoritmaları yardımıyla gruplandırarak segmentasyon işlemini gerçekleştirirler. Bu segmentasyon tekniklerinin kısa açıklamaları bu bölümün alt-başlıklarında verilmektedir.

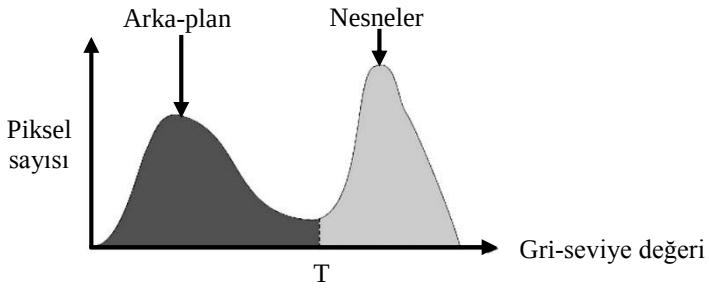
3.2.1. Eşikleme-Tabanlı Segmentasyon

Eşikleme-tabanlı yaklaşımlar görüntüyü parçalarına ayırmak için kullanılan en kolay tekniklerdir. Eşikleme işlemi ile seçilen bir aralığa göre görüntünün pikselleri yeniden eşleştirilir. Bu işlemin amacı görüntü içindeki bölgeleri görüntü arka planından ayırmaktır. Eşik değerine bağlı olarak çok seviyeli bir görüntünün pikselleri ön-plan ve arka-plan görüntüsü şeklinde ikili bir resme dönüştürülür. Renk değerlerine göre bir veya daha fazla aralık seçilerek bu aralık içinde kalan pikseller birbirinden ayrıştırılır. Eşikleme, çok-değişkenli sınıflandırma işlemleri için görüntünün birkaç özelliğine bağlı olarak pikselleri sınıflandıran çok-boyutlu bir diskriminant fonksiyon şeklinde genelleştirilebilir (Sankur ve Sezgin, 2004; Dias ve ark., 2002).

En basit eşikleme uygulaması için bir gri-seviye eşik değeri görüntü boyunca sabit tutular ve gri-seviye dağılım histogramından faydalanılır. Örneğin koyu bir arka-plan üzerinde açık renkli nesnelerden oluşan bir görüntünün gri-seviye histogramı çift durumludur. Göreceli olarak histogramdan belirlenen bir T global eşik değeri ile görüntüdeki piksel değerlerini karşılaştırmak yeterli olacaktır. Bu durum eşiklenmiş bir $g(x,y)$ görüntüsü için denklem (3.9)'deki gibi tanımlanır.

$$g(x,y) = \begin{cases} 1 & eğer\ (x,y) > T \\ 0 & eğer\ (x,y) \leq T \end{cases} \tag{3.9}$$

Bu eşiklemenin sonucu ikili bir görüntüdür. Piksel yoğunluk değeri 1 olanlar nesneleri, 0 olanlar ise arka-planı gösterir.



Şekil 3.3. Gri-seviye histogramı

Eşik değerinin seçimine göre iki yöntem mevcuttur. Bu yöntemler global ve yerel eşikleme olarak bilinir. Pal ve Pal (1993)'ın yaptığı literatür incelemesine göre eşikleme tekniklerini iki-seviyeli ve çok-seviyeli eşikleme olarak sınıflandırmışlardır. T eşik değeri tüm görüntü için sabit alındığında global (bütünsel) eşik olarak adlandırılırken, aksi durumda yerel (bölgesel) eşik olarak adlandırılır. Bütünsel eşik değerini belirlemek her zaman mümkün değildir. Örneğin arka-plan aydınlatması düzensiz olduğunda bütünsel eşik başarısız olabilir. Bu durum bölgesel eşikleme kullanılarak telafi edilebilir. Eşik değer seçimi etkileşimli olarak yapılabildiği gibi pik ve tepe noktalarını dikkate alarak hesaplama yapan otomatik eşik seçme algoritmaları kullanmak da mümkündür. Bunların en bilinen örneği Otsu metodudur. Eşikleme tekniğinin çok kanallı görüntülere uygulanmasının zor olması; daha ziyade sadece iki sınıfı ayrıştırmak için uygun olması; gürültü ve yoğunluk değişimlerine karşı duyarlı olması, yöntemin dezavantajları olarak söylenebilir.

Renkli görüntüler için bir pikselin renk bilgisi genellikle üç kanalın karışımından oluşur. Bu nedenle bu renk kanallarına ayrı ayrı histogram eşiklemesi uygulanarak görüntüyle ilgili bilgi üretilebilir. Her bir histogramdaki pik ve tepe noktalarına bakılarak renkli resimler için kullanılan çalışmalar vardır. Ancak literatürde daha çok görüntü boyutu bire indirgenerek tek-seviyeli eşikleme ile yapılan çalışmalara rastlanmaktadır.

3.2.2. Kenar-Tabanlı Segmentasyon

Genel bir tanımlamayla bir görüntünün piksel değerlerinde ani değişim gösteren bölgelere kenar adı verilir. Cisimlerin fiziksel özellikleri ile kenarları arasında doğrudan bir ilişki söz konusudur. Dolayısıyla görüntünün birçok fiziksel özelliği kenar bilgisinden ortaya çıkarılabilir. Kenar belirleme konusu, görüntü analizindeki önemli konulardan birisidir (Gonzales ve Woods, 2002). Görüntüdeki piksellerin renk değeri

kenarların konumu hakkında kısmi ve kararsız bilgi içerir. Kararsızlığın kaynaklarından birisi; öncelikle görüntüleme sürecinde daha sonra iletim ve örnekleme süreçlerinde oluşan gürültüden dolayı meydana gelir. Karasızlığın bir diğer sebebi de kullanılan ölçüm cihazlarının kusursuz olmaması ve sonuçlarının sadece kısmi gözlemlere dayanmasıdır. Gürültü yüksek frekanslı bileşenlerden oluştuğu için gürültü ile kenarları birbirinden ayırt etmek güçleşecektir. Bu açıdan bakıldığında; kenar belirleme metotlarının kötü tanımlanmış bir yapıları vardır. Bu kısıtlamalara bağlı çalıştıkları için eşsiz bir sonuç üretemezler.

Kenar belirleme teknikleri bölge sınırlarındaki pikselleri bulmak için kullanılır (Khan ve ark., 2011). Kenarların tespit edilmesi veya farklı bölgeler arasındaki hızlı yoğunluk geçişlerinin çıkarılması ile görüntü segmentasyonu elde edilmeye çalışılır. Daha sonra kapalı bölge sınırları oluşturmak için kenarlar bağlantılandırılır. Sonuç olarak ikili bir görüntü elde edilir. Kenar belirleme teknikleri görüntü segmentasyonunda kullanılan yapısal tekniklerden biridir. Bu bağlamda kenar tespitine dayanan başlıca iki segmentasyon yöntemi vardır (Kang ve ark., 2009). Bunlar gri-histogram tekniği ve gradiyent temelli yöntemlerdir.

Gri-histogram tekniğinde; kenarların tespit edilmesi büyük ölçüde doğru eşik değerinin seçilmesine dayanır. Maksimum ve minimum gri-seviye değerlenin tutarlı bir şekilde belirlenmesi, görüntüdeki gürültüden kaynaklanan histogram düzensizliğinden dolayı kolay değildir. Nesne ve arka-plan için kullanılan konik Gauss eğrilerinin kesişim noktası eşik değer olarak kullanılır. Gradiyent-temelli yöntemde; kenarları belirlemenin en kolay yolu görüntüde hızlı yoğunluk geçişi olan yerlere bakmaktır. Bu amaçla kullanılan iki temel yaklaşım bulunmaktadır. Bu yaklaşımlar bölgesel türev hesabına dayanır. Bölgesel olarak görüntünün birinci türevi kenar bölgelerinde en büyük değere sahip olur. Görüntünün ikinci türevi ise kenar bölgelerinde sıfır değerini alır. Bu yaklaşımda kenar tespiti için en çok kullanılan operatörler Sobel, Canny, Laplace ve LOG (Laplacian of Gaussian) vb. şeklindedir (Narkhede, 2013).

3.2.3. Bölge-Tabanlı Segmentasyon

Bölge-tabanlı yöntemler çoğunlukla bir bölgedeki komşu piksellerin benzer özellik gösterdiği varsayımına dayanmaktadır. Kenar belirleme tekniklerine göre daha basittirler ve gürültüden daha az etkilenirler. Temel prosedürleri bir pikselin komşularıyla karşılaştırılmasıdır. Karşılaştırma sonucunda, seçilen benzerlik ölçütü

sağlanıyorsa piksel ve benzer komşu pikseller aynı kümeye etiketlenirler. Dolaysıyla benzerlik ölçütünün belirlenmesi en önemli etkendir. Bölge-tabanlı segmentasyon algoritmaları genellikle bölge-genişletme yöntemleri ve parçala-birleştir yöntemleri olarak iki ana başlık altında incelenir (Gonzales ve Woods, 2002; Narkhede, 2013; Wang, 2009; Cheng ve ark., 2001).

3.2.3.1. Bölge Genişletme

Bölge-genişletme yöntemleri, önceden tanımlanan bazı ölçütlere göre benzerlik gösteren pikselleri iteratif şekilde tümevarım yöntemiyle belirleyerek görüntüdeki homojen bölgeleri çıkarmak için kullanılır. Literatürde çekirdekli ve çekirdeksiz bölge-genişleme algoritmaları şeklinde iki türü bulunmaktadır (Gonzales ve Woods, 2002; Narkhede, 2013; Wang, 2009;).

Çekirdekli Bölge Genişletme (SRG)

Görüntünün piksellerinden, bölgeler için başlangıç noktaları olarak bir piksel kümesi oluşturulur. Bu kümedeki piksellerin her biri çekirdek olarak bilinir ve bu çekirdeklerden genişleyerek görüntüdeki homojen ve ayrık bölgeler elde edilir. Algoritmanın işleyişi aşağıda adım adım açıklanmaktadır (Wang, 2009).

Adım1. $C_1, C_2, \dots, C_n$ bölgelerinde kümelenmek üzere n adet $p_1, p_2, \dots, p_n$ başlangıç çekirdek noktaları belirlenir.

Adım2. Başlangıç çekirdek noktası p_i ve komşu pikselleri arasındaki farklılık hesaplanır. Bulunan değer tanımlanan eşik değerinden küçükse, komşu pikseller C_i kümesine eklenir.$(i = 1,2, \dots, n)$

Adım3. C_i kümesine yeni eklenen piksellerle birlikte kümedeki tüm piksel değerlerinin ortalaması alınarak p_i çekirdeğinin değeri güncellenir.

Adım4. Tüm pikseller uygun bir kümeye atanıncaya kadar, adım 2-3 tekrarlanır.

Eşik değeri kullanıcı tarafından tanımlanır ve genellikle yoğunluk, gri-seviye veya renk değerlerine göre belirlenir. Bölgeler mümkün olduğunca homojendir. Bununla birlikte SRG tekniğinin iki önemli dezavantajı vardır: (1) optimal başlangıç çekirdek noktalarının belirlenmesi ve (2) çok zaman harcaması. Farklı çekirdeklerin seçilmesi farklı segmentasyon sonuçlarına neden olabilir. Bu durum segmentasyon sonuçlarının kararlılığını azaltır. Diğer bir problem, zaman-tüketimi yöntemin en ciddi sıkıntısıdır.

Çekirdeksiz Bölge Genişletme (URG)

Çekirdeksiz bölge genişletme algoritması SRG algoritmasından türetilerek geliştirilmiştir (Lin ve ark., 2000). URG tekniğinin, SRG'den farklı olarak haricen çekirdek seçmesi gerekmez. Çekirdek noktaları otomatik olarak üretilir. Bu şekliyle URG tekniği tam otomatik olarak çalışan bölge-tabanlı bir segmentasyon algoritmasıdır. URG'nin çalışma adımları aşağıda verilmiştir (Wang, 2009).

Adım1. Süreç tek bir piksel içeren C_1 kümesi ile başlar ve çalışmanın ilerlemesine göre yeni kümeler tanımlanır.

Adım2. Kümelerden en az bir tanesi ile çevrelenen etiketlenmemiş pikseller bulunur. Denklem (3.10)'deki gibi tanımlanır.

$$S = \left\{ x \notin \bigcup_{i=1}^{n} C_i \wedge \exists k : N(x) \cap C_k \neq \emptyset \right\} \tag{3.10}$$

$N(x)$, x noktasının mevcut komşularıdır. δ ise farklılık ölçüsüdür ve denklem (3.11)'deki gibi tanımlanır.

$$\delta(x, C_i) = \left| g(x) - \underset{y \in C_i}{\text{mean}}[g(y)] \right| \tag{3.11}$$

$g(x)$, x noktasının piksel değerini gösterir.

Adım3. Bir z noktası ve C_j kümesi seçilir. $z \in S$ ve $j \in [1, n]$ dir. Öyle ki

$$\delta(z, C_j) = \min_{x \in S, k \in [1,n]} \{\delta(x, C_k)\} \tag{3.12}$$

Eğer $\delta(z, C_j)$ önceden tanımlanmış bir t eşik değerinden küçük ise piksel C_j kümesine eklenebilir. Aksi durumda en benzer C kümesi seçilmelidir.

$$C = arg \min_{C_k} \{\delta(z, C_k)\} \tag{3.13}$$

Eğer $\delta(z, C_j) < t$ ise piksel C kümesine eklenebilir. Her iki şart da sağlanmamış ise, bu pikselin şu ana dek bulunan kümelerden birine ait olmadığı anlaşılır. Dolayısıyla yeni bir C_{n+1} kümesi üretilir ve z noktası başlangıç noktası kabul edilir.

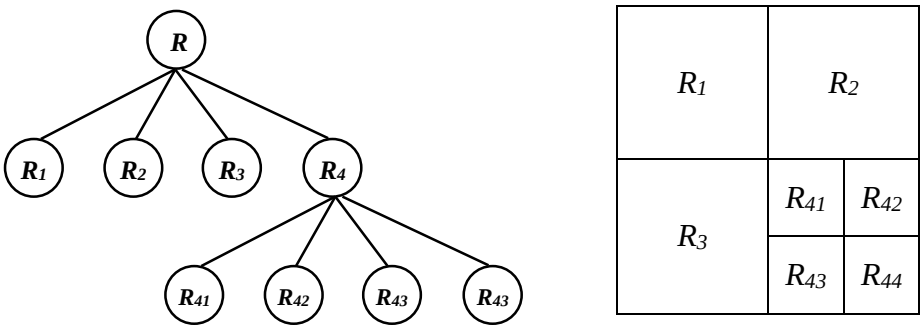
Adım4. Piksel kümeye eklendikten sonra, kümenin ortalama piksel değeri güncellenir.

Adım5. Tüm pikseller sınıflandırılıncaya kadar Adım-2 ile Adım-4 arası tekrarlanır.

3.2.3.2. Parçala – Birleştir

Bölge parçalama ve birleştirme işleminin amacı tümdengelim yöntemiyle görüntüyü daha homojen parçalara ayırmaktır. Dörtlü-ağaç yapısından faydalanır. Bu ağaç türünde her düğümün 4 çocuğu vardır ve kök düğümü görüntünün tamamını ifade eder. Bu durum şematik olarak Şekil (3.4)'de gösterilmektedir. Burada sadece dördüncü alt bölge tekrar alt bölgelere ayrılmıştır. Algoritmanın temel çalışma adımları aşağıda verilmiştir (Gonzales ve Woods, 2002).

R görüntünün tamamını temsil eder. Ayrıca P bir öngörü fonksiyonu olmak üzere $P(R_i) = FALSE$ olduğu durumda R_i bölgesinin dörde bölünmesi gerektiğini ve $P(R_i) = TRUE$ durumunu sağlayacak homojenlik elde edilinceye kadar parçalama işleminin devam etmesi gerektiğini gösteren homojenlik ölçütüdür.



Şekil 3.4. Dörtlü ağaç yapısı (Gonzales ve Woods, 2002)

Algoritma genel olarak 3 adımda özetlenebilir (Wang, 2009).

Adım1. Parçalama adımları: herhangi bir R_i bölgesi için $P(R_i) = FALSE$ ise bölge dört eşit dikdörtgene parçalanır.

Adım2. Birleştirme adımları: başka bir parçalama mümkün olmadığında, bitişik R_j ve R_k bölgeleri için $P(R_j \cup R_k) = TRUE$ şartını sağlayan bölgeler birleştirilir.

Adım3. Birleştirilebilecek küme kalmayıncaya kadar işlem devam eder.

Bölge sayısının otomatik olarak tespit edilebilmesi, parçalama ve birleştirme ölçütlerinin isteğe bağlı olarak farklı seçilebilmesi yöntemin avantajlarıdır. Bununla birlikte çok küçük parçaların oluşabilme ihtimali ve bu durumun birleştirme aşasında fazla zaman tüketmesi yöntemin dezavantajı olarak ifade edilir.

3.2.4. Kümeleme-Tabanlı Segmentasyon

Kümeleme; sonlu bir küme kategorisi tanımlamak için ihtiyaç duyulan danışmansız bir öğrenme problemidir. Bir veri kümesindeki verileri benzerliklerine göre belli gruplar halinde düzenleme işi olarak tanımlanır. Dolayısıyla kümeleme işleminin amacı; (1) aynı küme içindeki verilen benzerliklerinin maksimum olmasını sağlamak, (2) farklı kümlerdeki verilen benzerliklerinin ise minimum olmasını sağlamaktır. Farklı veri türleri için geliştirilmiş birçok kümeleme tekniği vardır. Görüntü segmentasyonu; renk bilgisi, doku, yoğunluk, şekil faktörü vb. bazı görüntü nitelikleri bakımından benzerlik gösteren piksellerin gruplanarak bölgelere ayrılması olarak düşünüldüğünde; bu süreç bir kümeleme işlemi olarak ele alınabilir. Renk bilgisinin bir piksel için genellikle kırmızı, mavi ve yeşil gibi üç bileşenle ifade edildiği dikkate alınırsa, renkli görüntülerin gri-seviyeli görüntülerden daha karmaşık bir yapıları olduğu anlaşılır. Bu durumda kümeleme-temelli metotların renkli görüntü segmentasyonu için kullanılabilecek ideal araçlar olduğu söylenebilir (Turi, 2001; Saraswathi ve Allirani, 2013).

Kümeleme algoritmaları genellikle hiyerarşik ve bölünmeli metotlar olarak ikiye ayrılır (Jain ve ark., 1999). Bölünmeli metotlar verileri önceden tanımlanan kümelere doğrudan eşleştirir. Veriler birbirini kapsamayan alt kümelere ayrılır. Her bir veri alt kümelerden sadece birinde yer alır. Hiyerarşik metotlar ise veriyi kümelemek için iç içe geçmiş hiyerarşik veri kümeleri kullanır. Tek bir kümeden tüm verileri içeren evrensel bir kümeye doğru bir gruplama yapılabileceği gibi, bu durumun tersi de mümkündür. Bölünmeli metotlar, hiyerarşik metotlardan daha hızlı çalışır. Çünkü tüm verileri kapsayan bir benzerlik matrisinin kullanılması gerekmez. Bu yönüyle büyük veri kümeleri için daha uygundurlar.

Bu bölümde görüntü segmentasyonunda yoğun olarak kullanılan iki temel kümeleme algoritması üzerinde durulacaktır.

3.2.4.1. K-Ortalamalar Algoritması (K-Means)

En bilinen bölünmeli kümeleme algoritmasıdır. Uygulanması kolay, hesaplama karmaşıklığı düşük, hızlı çalışan bir algoritmadır. Bununla birlikte başlangıçta rasgele seçilen küme merkezlerine karşı sonucu fazlaca etkileyecek kadar hassastır. K-

Ortalamalar kümeleme algoritması temel olarak dört adımdan oluşur (MacQueen, 1967; Thilagamani ve Shanthi, 2011).

Adım1. İstenen toplam küme sayısı N tanımlanır ve küme merkezleri rasgele oluşturulur.

Adım2. Veriler kendisine en yakın küme merkezine atanır. Bu amaçla genellikle benzerlik ölçütü olarak piksellerin yoğunluk değerleri kullanılmakla beraber oluşturulan bir öznitelik vektörü de kullanılabilir.

Adım3. Yapılan sınıflandırmaya göre küme merkezleri yeniden hesaplanır. Bu işlem denklem (3.14)'ye göre yapılır. Burada m_j j.küme merkezi değerini; n_j j. kümedeki toplam piksel sayısını; C_j j. kümeyi gösterir.

$$m_j = \frac{1}{n_j} \sum_{\forall p \in C_j} p \tag{3.14}$$

Adım4. Küme merkezlerinin değeri kararlı hale gelinceye kadar 2. ve 3. adımlar tekrarlanır.

3.2.4.2. Bulanık C-Ortalamalar Algoritması (FCM)

Bulanık C-Ortalamalar, K-Ortalamaların uzantısı olarak geliştirilmiş yaygın kullanılan bir diğer bölünmeli kümeleme tekniğidir. Bulanık C-Ortalamalar 1973 yılında Dunn tarafından önerilmiş ve 1981 yılında Bezdek tarafından geliştirilmiştir (Bezdek, 1981). Görüntü analizi ve örüntü tanıma uygulamalarında yaygın olarak kullanılmaktadır (Höppner ve ark. 2000). Bulanık mantık prensibi ile veriler birden fazla kümeye ait olabilir. Bu anlamda; veri kümesindeki her veri, kümelerin her birine $[0,1]$ aralığında değişen birer üyelik değeri ile aittir. Bir verinin tüm sınıflara olan üyelik değerleri toplamı bir olmalıdır. Algoritma piksellerin kümelenmesi olarak düşünüldüğünde; bir pikselin üyelik değeri hangi bölge merkezi için daha fazlaysa, bu pikselin o bölgeyi oluşturan elemanlardan biri olma ihtimali daha yüksektir.

Bulanık C-ortalamalar algoritması, amaç fonksiyonu temelli bir metottur. Algoritma, en küçük kareler yönteminin genellemesi olan denklem (3.15)'deki amaç fonksiyonunu öteleyerek minimize etmeye çalışır (Jain ve ark., 1999).

$$J_m = \sum_{i=1}^{N}\sum_{j=1}^{C} u_{ij}^{m} \left\|x_i - c_j\right\|^2 \tag{3.15}$$

Denklemde kullanılan m birden büyük herhangi bir reel sayıdır. x_i verisinin j.kümeye olan üyelik derecesi u_{ij} ile gösterilir. c_j ise j. kümenin küme merkez vektörünü gösterir. x ile gösterilen veriler $N \times d$ boyutlu bir vektördür. $||*||$ ifadesi verilerle küme merkezleri arasındaki benzerliği ölçmek için kullanılır. Genellikle Öklid mesafesi kullanılmakla birlikte herhangi bir uzaklık ölçütü kullanılabilir.

Bulanık c-ortalamalar algoritmasının çalışması dört adımda açıklanabilir.

Adım1. $U = [u_{ij}]$ üyelik matrisine rasgele değer atanarak algoritma başlatılır.

Adım2. Merkez vektörleri denklem (3.16)'daki denkleme göre hesaplanır.

$$c_j = \frac{\sum_{i=1}^{N} u_{ij}^{m} x_i}{\sum_{i=1}^{N} u_{ij}^{m}} \tag{3.16}$$

Adım3. Hesaplanan küme merkezlerine göre U matrisi denklem (3.17) kullanılarak yeniden hesaplanır.

$$u_{ij} = \frac{1}{\sum_{k=1}^{C} \left(\frac{\left\|x_i - c_j\right\|}{\left\|x_i - c_k\right\|} \right)^{\frac{2}{m-1}}} \tag{3.17}$$

Adım4. Eski $U(k)$ ve yeni $U(k+1)$ matrisleri karşılaştırılır ve fark ε'dan küçük oluncaya kadar işlem devam eder. Eğer $\|U(k) - U(k+1)\| < \varepsilon$ ise durdurulur aksi durumda; ikinci adımdan devam edilir.

Kümeleme sonucunda U üyelik matrisine gerekirse durulaştırma işlemi uygulanır ve değerler 0 ve 1'lere dönüştürülür.

3.2.5. Diğer Segmentasyon Yaklaşımları

Görüntü segmentasyonu için bahsedilen metotların dışında kullanılan ve başlı başına ayrı birer kategori olarak değerlendirilebilecek türde birçok melez yaklaşım

literatürde yer almaktadır. Bu metotlar; bulanık teknikler, fizik-tabanlı teknikler ve yapay sinir ağlarına dayalı teknikler olarak sıralanabilir (Cheng ve ark., 2001).

Bulanık Teknikler – Ayrıştırılması istenen bölgeler görüntü üzerinde belli belirsiz tanımlanmamış olabilir. Bu belirsizlik; görüntü analizinin ve örüntü tanımanın işlem aşamalarının tümünü etkileyebilir. Bulanık küme teorisi kesin olmayan ve muğlâk durumları temsil eden ve bu durumları yöneten bir mekanizma sunar. Görüntü segmentasyonu literatüründe; bu durumların ele alınması amacıyla, bulanık işlemcilerin, özelliklerin ve çıkarım kurallarının kullanıldığı çok önemli çalışmalar mevcuttur.

Fizik-tabanlı segmentasyon yaklaşımları – renkli görüntülerdeki gölge veya parlamaların oluşturduğu sahte kenarları elemine ederken; nesnelerin sınırlarını tespit edecek fiziksel modellerden faydalanarak problemi çözmeyi amaçlar. Fiziksel modeller arasında "iki renkli yansıma modeli" ve "yaklaşık renk yansıtma modeli" en yaygın olanlarıdır.

Yapay sinir ağları, örüntü tanımada oldukça yaygın bir şekilde kullanılmaktadır. Birbirlerine değişken ağırlıklarla bağlanmış çok sayıda hesaplama elemanından oluşur. Doğrusal olmayan, karışık ve birbirine bağımlı veri kümelerini temsil eder. Eğitim süresi genellikle çok uzundur. Ancak eğitimden sonra YSA kullanılarak yapılan sınıflandırma hızlıdır. Bu bakımdan görüntüden çıkarılan öznitelik vektörünün kümelenmesi veya sınıflandırılması amacıyla da kullanılan örüntü tanıma çalışmaları vardır.

4. ABC VE PSO İLE KÜMELEME TABANLI GÖRÜNTÜ SEGMENTASYONU

Çalışmanın bu bölümünde görüntü segmentasyonu için dört farklı kümeleme algoritması kullanılmıştır. Bu algoritmalardan ikisi klasik danışmansız kümeleme algoritmaları olan K-Ortalamalar ve Bulanık C-Ortalamalar algoritmalarıdır. Bir diğeri Omran tarafından önerilen PSO-temelli kümeleme algoritmasıdır (Omran, 2004). Son olarak, görüntü segmentasyonuna uyarlanmak üzere yeni bir yaklaşım olarak ABC-tabanlı bir kümeleme algoritması önerilmiştir.

Aşağıdaki alt başlıklarda, sırasıyla kümeleme yoluyla görüntü segmentasyonuna uyarlanan PSO ve ABC temelli algoritmaların işleyişi açıklanmıştır. Daha sonra amaç fonksiyonu olarak kullanılan değerlendirme ölçütleri üzerinde durulmakta ve son olarak deneysel sonuçlar verilmiştir.

4.1. PSO Temelli Kümeleme Algoritması

Omran (2004) tarafından önerilmiş olan kümeleme metodunda küme merkezlerinin sayısının önceden bilindiği varsayılmıştır. Bu algoritma ile optimal küme merkez değerlerinin bulunması hedeflenir. PSO ile üretilen çözümlerin parametrelerini küme merkezleri oluşturur. Piksellerin gri-seviyelerinden faydalanan algoritma, [0-255] aralığında N adet reel sayıdan oluşan parçacık vektörleri üretir. Böylece pikseller gri-seviye değerine göre kendisine en yakın merkez değerine sahip kümede toplanırlar. Algoritmanın avantajı; PSO algoritmasının paralel işlem yeteneğinden faydalanarak; klasik kümeleme yöntemlerinin aksine ilk üretilen küme merkezlerinin değerine bağlılığı azaltması ve kararsız sonuç üretme ihtimalini en aza indirgemesidir.

PSO temelli kümeleme algoritmasının çalışmasını özetleyen sözde-kod Şekil 4.1'de verilmiştir.

4.2. ABC Temelli Kümeleme Algoritması

ABC algoritmasının farklı birçok optimizasyon problemi üzerinde PSO ile kıyaslandığında daha iyi sonuçlar verdiği ve problemlerin parametre sayılarının artmasıyla algoritmanın performansını yeterli düzeyde koruyabildiği bildirilmiştir (Karaboğa ve Akay, 2009). Literatürde PSO'nun görüntü segmentasyonu için önerilen

N adet başlangıç küme merkez değerini rasgele üret
for $t = 1\ to\ t_{max}$
Tüm parçacıklar için
For each pixel z_p
z_p pikselinin tüm küme merkezlerine C_{ij} olan uzaklığını $d(z_p, m_{ij})$ hesapla
z_p pikselini denklem (4.1)'e göre belirlenen küme merkezine ata

$$\mathrm{d}(\mathrm{z_p}, \mathrm{m_{ij}}) = \min_{\forall c=1,\dots,\mathrm{N_c}}\{\mathrm{d}(\mathrm{z_p}, \mathrm{m_{ic}})\} \quad (4.1)$$

Uygunluk değerlerini hesapla,
Global en iyi çözüm bul ve sakla
end for i
Küme merkezleri denklem (4.2)'e göre güncelle

$$v_i(t+1) = \omega v_i(t) + c_1 r_1(t)(y_i(t) - x_i) + c_2 r_2(t)\big(gbest(t) - x_i(t)\big) \quad (4.2)$$

$$x_i(t+1) = x_i(t) + v_i(t+1)$$

end for t_{max}

Şekil 4.1. Omran'ın PSO temelli kümeleme algoritması

çok sayıda başarılı uygulaması bulunmasına rağmen; görüntü analizi alanında sınırlı sayıda ABC temelli çalışma vardır. Önerilen yaklaşımda ABC algoritması uygun küme merkezlerinin tespitinde kullanılmıştır ve kümeleme yoluyla görüntü segmentasyonuna uyarlanmıştır. Uygunluk fonksiyonu olarak *niceleme hatası* kullanılmıştır. Önerilen ABC temelli kümele algoritmasının çalışmasını adım adım açıklanmıştır.

Adım1. Popülasyondaki çözümleri oluşturan küme merkezlerinin ilk değerleri denklem (4.3) kullanılarak üretilir. Kullanılan gri-seviyeli görüntülerde tüm pikseller için $x_{min} = 0\ ve\ x_{max} = 255$ olarak alınır.

$$x_{ij} = x_{min}^{j} + rand(0,1)\big(x_{max}^{j} - x_{min}^{j}\big) \quad (4.3)$$

Adım2. Uygunluk değerleri amaç fonksiyonuna göre hesaplanır.

Adım3. Tüm işçi arılar için aşağıdaki işlemler tekrarlanır.

[a] v_i yeni küme merkezi hesaplanır.

[b] Yeni çözümün uyguluk değeri hesaplanır.

[c] Eski ve yeni üretilen çözüm arasında aç-gözlü seçim işlemi uygulanır.

Adım4. Denklem (4.4) kullanılarak hafızadaki çözümler için seçilme olasılıkları hesaplanır. fit_i : i. aday çözümün uygunluk değeridir.

$$p_i = \frac{fit_i}{max(fit)} \quad (4.4)$$

Adım5. Tüm gözcü arılar için aşağıdaki işlemler tekrarlanır.

[a] Olasılık değerlerine göre rulet tekerleği ile bir komşu çözüm seçilir.

[b] Denklem (4.5)'e göre yeni bir aday çözüm üretilir.

$$v_i = x_{ij} + \emptyset_{ij}(x_{ij} - x_{kj}) \tag{4.5}$$

[c] Eski ve yeni üretilen çözüm arasında aç-gözlü seçim işlemi uygulanır.

Adım6. İyileştirilemeyen çözümler varsa bunların yerine yenileri üretilir.

Adım7. Popülasyondaki en iyi çözüm hafızaya alınır.

Adım8. Maksimum çevrim sayısı sağlanıncaya kadar adım3'ten devam edilir.

4.3. Değerlendirme Ölçütleri

Başarılı bir kümeleme işlemi için elde edilen küme-içi benzerliğin maksimum, kümeler-arası benzerliğin ise minimum olması istenir. Bir kümeleme algoritmasının kalitesini değerlendirmek için kullanılan çeşitli göstergeler mevcuttur. Bunlar arasında sık kullanılan yöntemlerden birisi de niceleme hatasıdır. Denklem (4.6)'da matematiksel ifadesi verilmiştir. Çalışmada *niceleme hatası* PSO ve ABC algoritmalarında amaç fonksiyon olarak kullanılmıştır. Denklemde $C_j : j$.kümeyi gösterir. n_j ise bu kümenin eleman sayısıdır. z_p j. küme içinde bulunan her bir pikseli ifade etmek için kullanılır. m_j ise j. kümenin merkezidir.

$$J_e = \frac{\sum_{j=1}^{N_c} \left[\sum_{\forall z_p \in C_j} d(z_p, m_j)\right] / n_j}{N_c} \tag{4.6}$$

Bu göstergeye ek olarak elde edilen sonuçlar üzerinden kümelemenin başarısını anlamak amacıyla d_{min} ve d_{max} olarak adlandırılan iki ölçüt değeri daha hesaplanmıştır. d_{min} farklı kümeler arasındaki minimum mesafeyi gösterir. Denklem (4.7) kullanılarak hesaplanır.

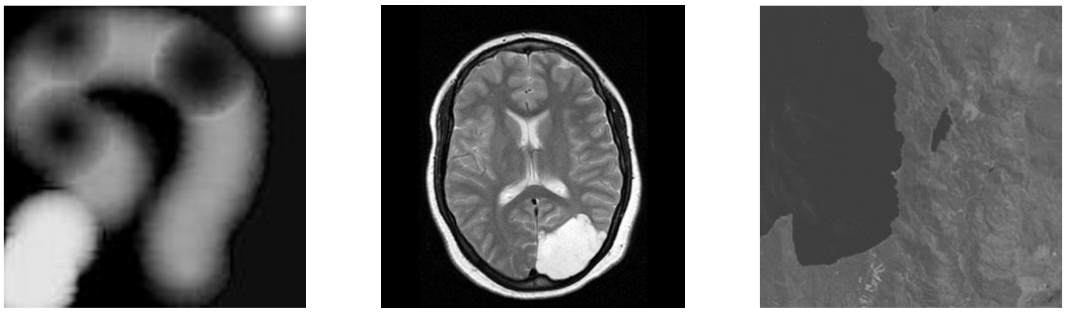
$$d_{min} = \min_{\forall j_1, j_2, j_1 \neq j_2} \{d(m_{j_1}, m_{j_2})\} \tag{4.7}$$

d_{max} ise tüm kümeler arasında maksimum küme-içi uzaklığı verir. Denklem (4.8) kullanılarak hesaplanır.

$$d_{max} = \max_{j = 1, \ldots, Nc} \left\{\sum_{\forall z_p \in C_j} d(z_p, m_j)/n_j\right\} \tag{4.8}$$

4.4. Deneysel Sonuçlar

Karşılaştırılan kümeleme algoritmalarının (K-Ortalamalar, Bulanık C-Ortalamalar, PSO ve ABC) hepsi Omran'ın çalışmasında kullandığı üç tür gri-seviye görüntüye önerilen parametre değerleri ile uygulanmıştır. PSO'da kullanılan parametre değerleri $V_{max} = 255, w = 0.72\ and\ c1 = c2 = 1.49$ olarak ayarlanmıştır. Test görüntüleri Şekil 4.2'de görülmektedir.



(a) Sentetik (b) MRI (c) Uydu

a. 300x300 boyutunda 8-bit gri-seviye – küme sayısı: 3
b. 300x300 boyutunda 8-bit gri-seviye – küme sayısı: 8
c. 100x100 boyutunda 8-bit gri-seviye – küme sayısı: 4

Şekil 4.2. (a) Sentetik, (b) MRI insan beyni ve (c) Uydu görüntüsü

Karşılaştırmanın sağlıklı olması için ABC ve PSO temelli kümeleme algoritmalarının her ikisinde de aynı parametre değerleri kullanılmıştır. PSO ve ABC için koloni boyutu 50 olarak ayarlanmıştır. Tüm algoritmalar 100 çevrim için her bir test görüntüsüne 20'şer defa uygulanmıştır. Elde edilen sonuçların ortalama gösterge değerleri ve standart sapma değerleri Çizelge 4.1'de verilmiştir. Ayrıca alınan sonuçların kümeleme sonuçları Şekil 4.3'te gösterilmiştir.

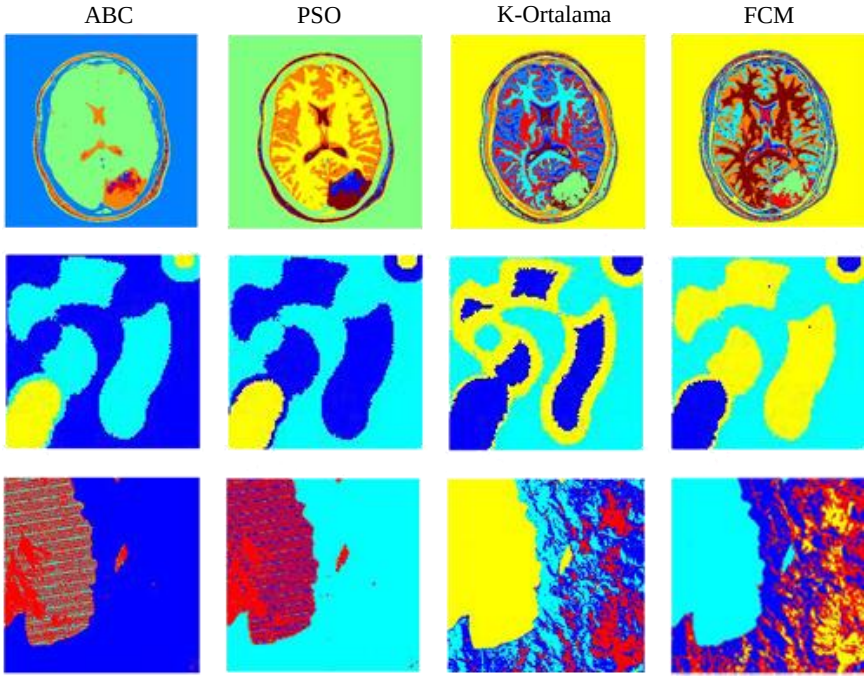
Sonuçlara bakıldığında genel olarak sürü zekâsı temelli kümeleme algoritmalarının klasik yöntemlerden daha iyi sonuçlar ürettiği görülmektedir. ABC temelli kümeleme algoritmasının kısmen PSO'ya göre daha iyi sonuçlar verdiği söylenebilir. Bu durum Şekil 4.4, Şekil 4.5 ve Şekil 4.6'dan görülebilir. Özellikle standart sapmalar dikkate alındığında ABC'nin bu test görüntüleri üzerinde farklı çalıştırmalar için daha kararlı çalıştığı anlaşılabilir. Bu durum Şekil 4.7, Şekil 4.8 ve Şekil 4.9'dan görülebilir.

Yapılan çalışmada kullanılan test görüntülerinin küme merkez sayıları literatürde kullanılan değerlerden alınmıştır. Algoritmalar her ne kadar danışmansız kümeleme teknikleri olsalar da, performansları ilk aşamada rastgele atanan küme merkezlerinin değerinden ve verilerin dağılımından oldukça etkilenmektedir. Önerilen ABC-temelli kümeleme algoritmasının başarımının, veri dağılımından ve başlangıç küme merkezlerinden bağımsız hale gelebilmesi için farklı optimizasyon teknikleriyle birleştirilerek akıllı küme merkezlerini seçebilecek ve bu yolla algoritmayı iyileştirecek yaklaşımlara ihtiyacı vardır.

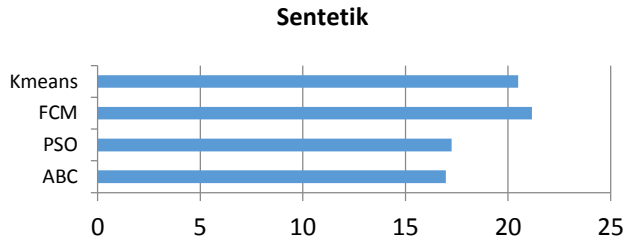
Ayrıca çalışmada kümeleme amacıyla sadece piksellerin gri-seviye renk değerleri kullanılmıştır. Sonuçların, daha verimli bir şekilde alınabilmesi için piksellerin konum bilgileri, yoğunluk değerleri ve renkli görüntülerin kullanılmasıyla oluşturulacak bir özellik vektörü kullanılabilir. Bunun yanı sıra, çok amaçlı optimizasyon tekniklerinin kullanılması faydalı olacaktır.

Çizelge 4.1. Gösterge sonuçları

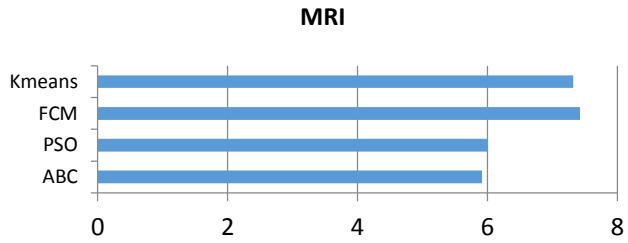
Algoritma	Test Görüntüsü	J_e	d_{min}	d_{max}
FCM	***Sentetik***	21.1574 ± 1.821e-05	30.0132 ± 4.2493e-05	79.5506 ± 7.4553e-06
	MRI	7.4239 ± 0.1118	12.1090 ± 0.6450	22.1548 ± 1.3528
	Uydu	3.1647 ± 0.0844	4.9993 ± 0.2088	10.9706 ± 2.1088
K-Ortalama	***Sentetik***	20.4943 ± 08238	28.6993 ± 2.2606	77.3356 ± 8.8083
	MRI	7.3219 ± 0.0921	12.6999 ± 1.1006	12.7236 ± 7.7773
	Uydu	3.2685 ± 0.0924	5.2229 ± 0.2772	9.2997 ± 2.8583
ABC	***Sentetik***	16.9628 ± 0.0040	24.4850 ± 0.1021	91.8500 ± 0.4894
	MRI	5.9177 ± 0.1497	14.6159 ± 1.6112	2.7500 ± 1.6182
	Uydu	2.6381 ± 0.1773	6.8998 ± 1.1693	2.0500 ± 2.0894
PSO	***Sentetik***	17.2487 ± 0.4405	27.2525 ± 0.6525	71.1500 ± 32.9741
	MRI	6.0026 ± 0.2238	13.3286 ± 2.1261	4.9500 ± 6.0391
	Uydu	2.7525 ± 0.1765	6.2508 ± 0.7248	4.3000 ± 3.1136



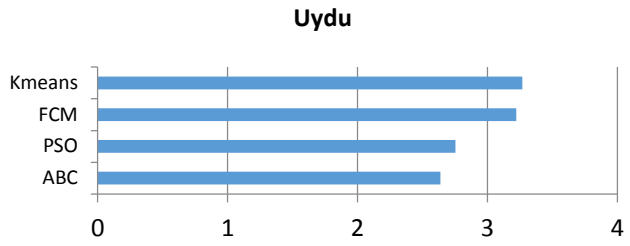
Şekil 4.3. Segmentasyon sonuçları



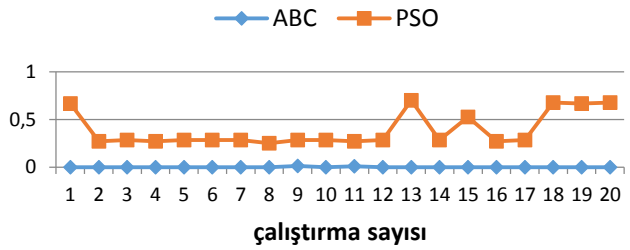
Şekil 4.4. Sentetik test görüntüsü için ortalama J_e değerleri



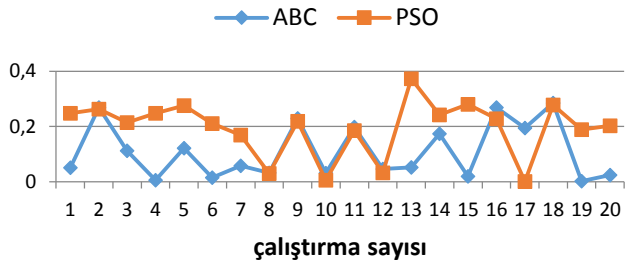
Şekil 4.5. MRI test görüntüsü için ortalama J_e değerleri



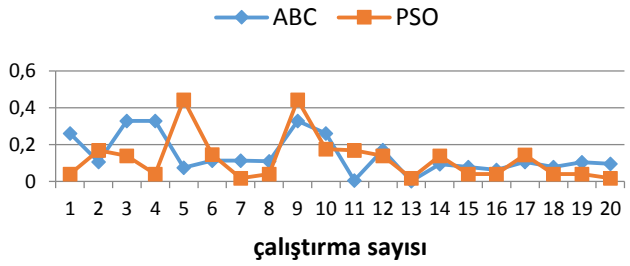
Şekil 4.6. Uydu test görüntüsü için ortalama J_e değerleri



Şekil 4.7. Sentetik test görüntüsü için ortalama J_e değerinden sapmaları



Şekil 4.8. MRI test görüntüsü için ortalama J_e değerinden sapmaları



Şekil 4.9. Uydu test görüntüsü için ortalama J_e değerinden sapmaları

5. ÇOK AMAÇLI OPTİMİZASYON VE SEGMENTASYON

Gerçek dünyada tanımlanan problemlerin pek çoğu birden fazla ve birbiriyle çelişen kısıtlamaların ve isteklerin birlikte sağlanmasını gerektirir. Örneğin bir köprü inşası için iyi bir tasarım, düşük kütle ve yüksek dayanıklılıkla ifade edilir. Uçak tasarımı; yakıt verimliliği, yük ve ağırlığın eş zamanlı optimizasyonunu gerektirir. Bir arabanın tavan pencere tasarımı için sürücünün duyacağı gürültünün minimize edilmesi, havalandırmanın ise maksimize edilmesi amaçlanır. Bir nakliye işinin planlanarak güzergâhın belirlenmesi için süre, yolun uzunluğu, park sorunu ve hatta bakım-onarım imkânları hesaba katılmalıdır (Osyczka, 2002; Erkut ve ark., 2007).

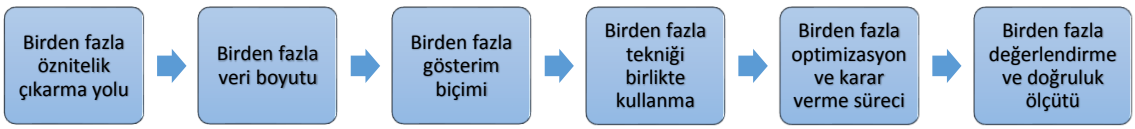
Bu bakımdan, çoklu-amaç durumunda tüm amaçlara göre en iyi olan tek bir çözüm mevcut olmayabilir. Bu durumda karar vericinin, üzerinde uzlaşma sağlanmış sonlu bir çözüm kümesinden bir çözüm seçmesi beklenir. Uygun çözüm, tüm amaçlar için kabul edilebilir düzeyde bir performans sağlamalıdır.

Son zamanlarda yeni bir trend olarak, görüntü segmentasyonunu formülize etmek için karar verme sürecinde birden fazla faktörü ayrı ayrı değerlendiren yaklaşımlar kullanılmaya başlanmıştır (Maulik ve Saha, 2009; Saha ve ark., 2009; Shirakawa ve Nagao, 2009; Nakib ve ark., 2009). Çünkü birçok görüntü segmentasyon problemi gerçekten de özellikleri bakımından çok amaçlı optimizasyon problemlerine dönüştürülebilir. Görüntü segmentasyonu, en basit tanımıyla bazı ölçütlere göre görüntüdeki piksellerin kümelenmesi şeklinde ele alındığında; başarılı bir segmentasyon algoritmasının sağlanması gereken ve birbiriyle çelişen amaçlar çıkarılabilir. Örneğin bölge-içi benzerlik sapmasının minimize edilmesi, bölgeler-arası farklılığın maksimize edilmesi, öznitelik vektörünün boyutunun indirgenmesi veya kümeleme göstergelerinden alınan hata oranının minimize edilmesi gibi.

Genel olarak görüntü segmentasyonu bir takım süreçlerin ardı adına uygulanmasıyla elde edilir. Öncelikle algoritmanın gruplayacağı toplam bölge sayısı bilinmelidir. Bölge sayısı; görüntünün örüntü bilgisi, türü ve ölçeği ile ilişkilidir. Daha sonra öznitelik seçim aşamasına geçilir. Öznitelik seçimi kümelemede kullanılacak özniteliklerin en etkili alt kümesinin belirlenmesi sürecidir. Diğer taraftan öznitelik çıkarma işlemi ile bir veya daha fazla dönüşüm işlemi kullanılarak yeni belirgin özelliklerin üretilmesi sağlanabilir. Bu bağlamda segmentasyonun amacına bağlı olarak

birçok özniteliğin seçilme ihtimali vardır. Sonraki aşama bölge benzerliğinin tespit edilmesidir. Bu işlem için genellikle Öklid uzaklığı gibi iki bölge arasındaki farklılığı yansıtan basit bir ölçüt kullanılabileceği gibi, bölgeler arası kavramsal benzerliği niteleyen ölçütler de kullanılabilir. Bölge benzerliği çok-kriterli bir konudur. Örneğin bir bölgeyi oluşturan piksellerin konumsal uyumluluğu, özniteliklerinin benzerliği ile birlikte düşünülmeli; bölgeler-arası farklılığa karşı bölge-içi benzerlik eş zamanlı olarak düşünülmelidir. Öznitelik çıkarımı ve benzerlik hesaplamalarından sonra veriler gruplanır. Gruplama işlemi değişik şekillerde yapılabilir. Seçilen ölçütlere bağlı olarak farklı algoritmalar değişik bölgeler üretebilir. Son olarak küme doğruluk analizi yapılır. Küme doğruluk ölçütleri oluşturulurken birçok kriter kullanılarak formülize edilir. Bu göstergelerin optimizasyonuyla segmentasyondaki optimal küme sayısı elde edilir (Bong ve Mandava, 2010).

Segmentasyon işlemi, sürecin başından sonuna kadar çok kriterli bir perspektife sahiptir. Belirli bir segmentasyon problemi için (1) öznitelikleri çıkarmanın birçok yolu vardır, (2) görüntüler çok boyutlu bir veri kümesi gibi düşünülebilir, (3) birçok metot birleştirilerek uygun çıktı elde edilebilir, (4) birçok değerlendirme ölçütü bir arada kullanılarak doğruluk analizi yapılabilir. Bu perspektif Şekil 5.1'de gösterilmektedir.



Şekil 5.1. Görüntü segmentasyonunun girişten çıkışa kadar çok kriterli yapısı

Tek-amaçlı optimizasyon ile arama uzayında küresel minimum veya küresel maksimuma ulaşarak tek bir kararın veya tasarımın elde edilmesi nispeten kolay bir süreçtir. Bunun aksine; görüntü segmentasyonu da dâhil olmak üzere gerçek dünya problemlerinde karşılaşılan çoklu-amaçların optimizasyonu ve amaçların birlikte değerlendirilmesi zor bir problemdir. Sezgisel teknikler bu problemlerin çözümü için kullanılabilecek uygun araçlardır.

Çok-amaçlı optimizasyon metotlarını üç sınıfta toplamak mümkündür. Bunlar toplama fonksiyonları, popülasyon temelli yaklaşımlar ve Pareto-temelli yaklaşımlardır. Toplama fonksiyonları; bu konuda, en temel ve en basit olarak nitelendirilebilecek yaklaşım türüdür. Bu yöntem, çoklu amaçları optimize edebilmek için, amaçların her birini tek bir sayısal değerde birleştirdiğinden, toplama fonksiyonları olarak adlandırılır.

Popülasyon temelli yaklaşımlarda; algoritmanın türüne göre popülasyon olarak adlandırılan bir aday çözümler listesi her bir amaç için alt popülasyonlara bölünür. Ancak bu algoritma kısmen başarılı sonuçlar üretiyor gibi görünse de oldukça zayıf ve dezavantajları olan bir algoritmadır (Coello ve Carlos 2005).

Pareto-temelli yaklaşımlarda ise Pareto-Optimal kavramına dayanan sınıflandırma kullanılarak bastırılmamış çözümler kümesi elde edilmeye çalışılır. Bu yaklaşımı kullanan algoritmalar, Goldberg tarafından önerilen Pareto derecelendirme metodundan esinlenir. Goldberg önerisinde, mevcut popülâsyona göre bastırılmamış olan uygun çözümlerin, bir seçim mekanizması kullanılarak Pareto yüzeye doğru popülâsyonun taşınması gerektiğini söyler.

Bu bölümde öncelikle Pareto-Optimal kavramı üzerinde durulmaktadır. Daha sonra çok amaçlı sezgisel optimizasyon algoritmaları arasında en başarılı ve orijinal özellik gösteren bastırılmamışlık sıralamalı genetik algoritma-II (NSGAII) algoritması açıklanmaktadır. Ayrıca temel PSO algoritması, bastırılmamışlık sıralama stratejisi ile birleştirilerek NSPSO olarak isimlendirilen yeni bir Pareto-temelli çok-amaçlı optimizasyon algoritması önerilmektedir. Son olarak bölüm değerlendirmesi yapılmıştır.

5.1. Pareto-Optimal Kavramı

Pareto Optimal kavramı 1900'lü yılların başlarında, esasen İtalyan bir iktisatçı ve sosyolog olan Vilfredo Pareto tarafından bulunmuş olan ekonomik yeni refah teorisinden gelmektedir. Pareto bir ulusun ekonomisinde olası olan denge durumlarını araştırırken, adını kendisinden alan Pareto optimal kavramını buldu. Pareto optimal duruma ulaşılması halinde insanların yararlı mallara ve hizmetlere olan gereksinimleri, en iyi biçimde tatmin edilmiş olur. Bununla modern refah teorisinin temellerini kurmuştur. Bu teoriye göre;

1. Herkesin durumunu daha iyi yapan veya kimsenin durumu değiştirmediği halde en az bir kişinin durumunu daha iyi hale getiren,

2. Herkesin durumunu daha kötü yapan veya kimsenin durumunu değiştirmediği halde en az bir kişinin durumunu daha kötü yapan değişmelerin değerlendirilmesini yaparak şu sonuçlara varmıştır:

Bazılarının durumunu daha iyi hale getirip bazılarının durumunu daha kötü hale getiren değişmelerin refahı ne yönde etkilediği söylenemez. Refah, ancak hiç kimsenin

durumunu kötüleştirmeden bazı kişilerin durumunu iyileştirebiliyorsa artmış sayılır ve şayet kimsenin durumunu kötüleştirmeden bir kişinin dahi durumunu iyileştirme olanağı yoksa refah maksimuma ulaşmış sayılır (Wikipedia).

Pareto-optimal kavramı, evrimsel algoritmalara uyarlanmasıyla, çok amaçlı evrimsel optimizasyon algoritmaları geliştirilmiştir. Buna göre, popülâsyondaki bir çözüm amaç değerlerine göre, en iyi, en kötü ve diğer çözümlerden farksız olabilir. En iyi çözüm, amaçların herhangi biri içinde en kötü olmayan ve en azından bir amaç içinde diğerlerinden daha iyi olan çözüm anlamındadır. Optimal çözüm, arama uzayında, herhangi bir diğer çözüm tarafından bastırılmayan çözümdür. Böyle bir optimal çözüme Pareto-Optimal Çözüm ve bütün bu şekildeki optimal trade-offs çözümlerin kümesi de Pareto-Optimal Küme olarak adlandırılır (Dias ve ark., 2002; Osyczka, 2002; Deb, 2002).

Genellikle çok amaçlı evrimsel algoritmaların amacı, tradeoff yüzeyi yani bastırılmamış çözüm noktaları kümesini tespit etmektir. Bu küme, yukarıda tanımlanan Pareto-Optimal ya da diğer bilinen ifadeleriyle bastırılmamış (nondominated) ya da ikinci dereceden olmayan (noninferior) çözümler kümesi olarak adlandırılır. Bu çözümlerden geriye kalanlar ise bastırılmış (dominated) çözümler olarak bilinir. Bastırılmamış çözümlerden herhangi birisi çözüm olarak kabul edilebilir. Bir çözümün tercih edilmesi gerekiyorsa problem bilgisi ve bir takım problemle ilişkili faktörlerin bilinmesine ihtiyaç vardır. Bu yüzden karar verici tarafından seçilen bir çözüm başka bir karar vericiye göre veya değişik bir çevre içinde uygun olmayabilir. Bu nedenle çok amaçlı optimizasyon problemlerinde, Pareto-Optimal çözümleri değerlendirebilmek için problem hakkında bilgi sahibi olmak önemlidir.

5.2. Çok Amaçlı Optimizasyon Problemleri

Çok-amaçlı optimizasyon problemi birden fazla amaçtan, eşitlik ve eşitsizliklerden oluşur. Bu problemlerin genel matematiksel formülü aşağıdaki gibi yazılabilir (Osyczka, 2002; Deb, 2002; Coello, 2005):

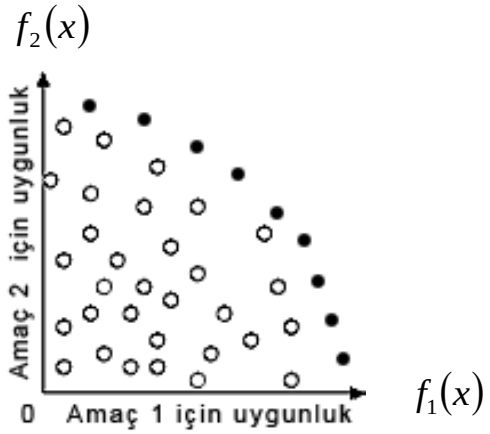
$$\begin{gathered} \textit{maksimize / minimize} \\ y = f(x) = \{f_1(x), f_2(x), \ldots\ldots f_M(x)\} \\ \textit{aşağıdakilere bağlı olarak} \\ g(x) = \{g_1(x), g_2(x), \ldots, g_J(x)\} \le 0 \\ h(x) = \{h_1(x), h_2(x), \ldots, h_K(x)\} = 0 \end{gathered} \tag{5.1}$$

Burada $x = \{x_1, x_2, ..., x_N\} \in X$ ve $y = \{y_1, y_2, ..., y_M\} \in Y$ x belirlenen bir karar değişkenler vektörü ve $f_1(.), f_2(.), ..., f_n(.)$ maksimize/minimize edilecek amaç fonksiyonlardır. Aşağıdaki eşitsizlikler *x* ve *y* çözümleri arasında tutulduğunda, *x* çözümü *y* çözümü tarafından bastırılır.

$$\forall i : f_i(x) \leq f_i(y) \text{ ve } \exists j : f_j(x) < f_j(y) \tag{5.2}$$

Bastırılmamış çözümlerin örnekleri Şekil 5.2'de görülmektedir. Bastırılmamış ve bastırılmış çözümler sırasıyla kapalı ve açık çemberlerle iki boyutlu amaç uzayında görülmektedir. Şekil 5.2'deki iki boyutlu amaç uzayı, aşağıdaki iki amaçlı optimizasyon problemini ifade eder.

$$\text{Maksimize } f_1(x) \text{ ve } f_2(x) \tag{5.3}$$



Şekil 5.2. Bastırılmamış çözümler (kapalı çemberler) ve bastırılmış çözümler (açık çemberler)

5.3. Bastırılmamışlık Sıralamalı Genetik Algoritma II (NSGA-II)

Daha önce Deb tarafından önerilmiş olan NSGA'nın yüksek hesaplama karmaşıklığı, seçkinlik eksikliği ve σ_{share} parametre değerinin ayarlanmasındaki belirsizlik gibi zayıf yönler göz önüne alınarak, yine Deb (2002) tarafından NSGA-II algoritması geliştirildi. NSGA-II strateji olarak NSGA'ya paralellik gösterse de algoritma açısından oldukça farklı ve yeni yöntemler içerir.

5.3.1. Hızlı bastırılmamış sınıflandırma yaklaşımı

N boyutlu bir popülâsyonda ilk bastırılmamış yüzeyin çözümlerini tanımlayabilmek için, her bir çözümün popülâsyondaki diğer tüm çözümlerle karşılaştırılması gerekir. Bu işleme popülâsyondaki bastırılmamış yüzeyin tüm üyelerini bulmak için devam edilir. Sonraki bastırılmamış yüzeylerdeki bireyleri bulmak için, ilk yüzeyin çözümleri geçici olarak dikkate alınmaz ve yapılan işlem tekrarlanır. Hesaplama karmaşıklığı açısından en kötü durum, her bir yüzeyde sadece tek çözümün olduğu, N tane yüzey bulunduğu zamandır. Hızlı bastırılmamış sınıflandırma yaklaşımı için sözde kod Şekil 5.3'te verilmiştir.

Hızlı bastırılmamış çözümleri sınıflandırma yaklaşımında, iki değer öncelikle hesaplanmalıdır. Bunlar; üstünlük sayısı n_p: bir p çözümünü bastıran çözümlerin sayısı ve S_p: p çözümünün bastırdığı çözümlerin kümesi.

```
fast_nondominated_sort(P)
for each p ∈ P
    S_p = 0
    n_p = 0
    for each q ∈ P
        if (p ≺ q) then                Eğer p, q yu bastırırsa
            S_p = S_p ∪ {q}            q yu p tarafından bastırılan çözümler kümesine ekle
        else if (q ≺ p) then
            n_p = n_p + 1              p nin üstünlük sayısını artır
    if (n_p = 0) then                  p ilk yüzeye mi ait?
        p_rank = 1
        F_1 = F_1 ∪ {p}
    i = 1                              yüzey sayacını başlangıç durumuna getir
    while F_i ≠ ∅
        Q = ∅                          sonraki yüzeyin elemanlarını depolamak için kullanılır
        for each q ∈ F_i
            for each q ∈ S_p
                n_q = n_q − 1
                if n_q = 0 then
                    q_rank = i + 1             q sonraki yüzeye mi ait?
                    Q = Q ∪ {q}
        i = i + 1
        F_i = Q
```

Şekil 5.3. Hızlı bastırılmamış çözüm sıralama metodu için sözde kod (Deb, 2002)

İlk bastırılmamış yüzeydeki tüm çözümlerin *üstünlük sayısı* sıfırdır. n_p = 0 olan her bir p çözümünün, S_p kümelerindeki her bir q elemanın üstünlük sayısı bir azaltılır. Bu işlemden sonra, eğer herhangi bir q elemanı için üstünlük sayısı sıfır olmuşsa, bu

ayrı bir *Q* kümesine koyulur. *Q* kümesindeki bu üyeler ikinci bastırılmamış yüzeyi oluşturur. Daha sonra, yukarıdaki işlem *Q*'nun her bir üyesi için devam ettirilir ve üçüncü yüzey tanımlanır. Bu işlem tüm yüzeyler tanımlanıncaya kadar devam eder.

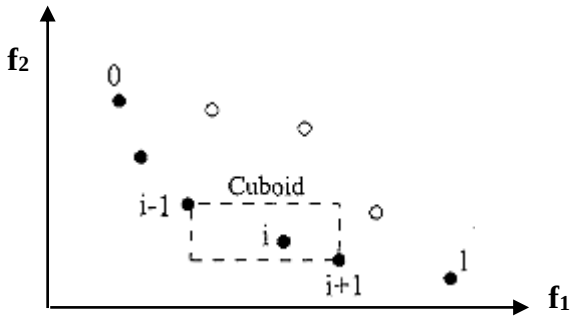
İkinci veya daha sonraki bastırılmamış yüzeylerde bulunan her bir *p* çözümü için üstünlük sayısı n_p, en fazla *N–1* olabilir. Böylece her bir *p* çözümü, üstünlük sayısı sıfır olmadan önce, en fazla *N–1* defa ele alınacaktır. Bu noktada, çözüm bir bastırılmamış yüzeye atanır ve tekrar ele alınmaz. En fazla böyle *N–1* çözüm olduğu için, toplam karmaşıklık $O(N^2)$ dir.

5.3.2. Çeşitliliğin sağlanması

Evrimsel algoritmanın Pareto-Optimal kümeye, doğru yakınsamayı sağlaması ve elde ettiği çözümlerin iyi bir yayılıma sahip olması istenir. Orijinal NSGA bu amaçla, bilinen bir paylaşım fonksiyonu kullanır. Bu fonksiyon, ilgili parametreleri uygun şekilde ayarlayarak popülâsyonda çeşitliliği sağlar. Paylaşım fonksiyonu, bir paylaşım parametresi (σ_{share}) içerir. Bu parametre, problemde istenen paylaşımın kapsamını ayarlar ve iki birey arasındaki yakınlığı hesaplamak için seçilen mesafe metriği ile ilişkilidir. σ_{share} parametresi, her biri diğerinin uygunluğunu paylaşan herhangi iki çözüm için mesafe metriğinin en geniş değerini gösterir. Bu parametre genellikle kullanıcı tarafından ayarlanır. Bu paylaşım fonksiyonu yaklaşımında iki zorluk vardır (Deb, 2002; Deb ve ark., 2002; Nebro ve ark., 2008).

1) Paylaşım fonksiyonu metodunun performansı için çözümlerin dağılım sağlanması, büyük ölçüde σ_{share} değerinin seçilmesine bağlıdır.
2) Her bir çözüm popülâsyondaki diğer tüm çözümlerle karşılaştırılmak zorunda olduğu için, paylaşım fonksiyon yaklaşımının tüm karmaşıklığı $O\left(N^2\right)$dir.

NSGA-II algoritmasında, yukarıdaki sözü edilen her iki güçlüğü ortadan kaldırmak amacıyla paylaşım fonksiyon yaklaşımı yerine *kalabalık-karşılaştırma* yaklaşımı kullanılmıştır. Yeni yaklaşım, popülâsyonun bireyleri arasındaki çeşitliliği sağlamak için herhangi bir kullanıcı tanımlı parametre gerektirmez. Ayrıca, daha iyi bir hesaplama karmaşıklığına sahiptir. Bu yaklaşımın anlaşılması için, yoğunluk-tahmin metriği ve kalabalık-karşılaştırma operatörü öncelikle açıklanması gerekir.



Şekil 5.4. Kalabalıklık mesafe hesaplaması

Popülâsyonda herhangi bir noktanın yoğunluğunu tahmin etmek için; ele alınan noktanın her iki kenarında bulunan iki noktanın ortalama uzaklığı tüm amaçlar için hesaplanır. Bu miktar ($i_{distance}$), en yakın komşular olan köşeler kullanılarak şekillendirilen küpün çevre uzunluğunun tahmini için kullanılır ve buna kalabalıklık mesafesi (crowding-distance) denir. Şekil-5.4'de içi dolu dairelerin oluşturduğu yüzeyde, i. bireyin kalabalıklık mesafesi, küboidin (*kesikli kutu ile gösterilen*) ortalama kenar uzunluğudur. Kalabalık-mesafe hesabı, amaç fonksiyon değerlerine göre artan büyüklükle popülâsyonu sıralamayı gerektirir. Daha sonra, her bir amaç fonksiyonunun sınır çözüm değerlerine (en küçük ve en büyük fonksiyon değerli çözümler) sonsuz bir uzaklık değeri atanır. Diğer tüm ara çözümler için, bitişik iki bireyin fonksiyon değerlerinde, tam normalleştirilmiş farka eşit bir uzaklık değeri hesaplanır. Bu hesaplama diğer amaç fonksiyonlarla devam eder. Bütün kalabalık-mesafe değeri, tüm amaçlarla ilgili birey uzaklıklarının toplamı olarak hesaplanır. Her bir amaç fonksiyon, kalabalık-mesafesi hesaplanmadan önce normalleştirilir. Bir *I* bastırılmamış çözüm kümesindeki tüm çözümler için kalabalıklık mesafesini hesaplayan algoritma Şekil 5.5'te verilmiştir.

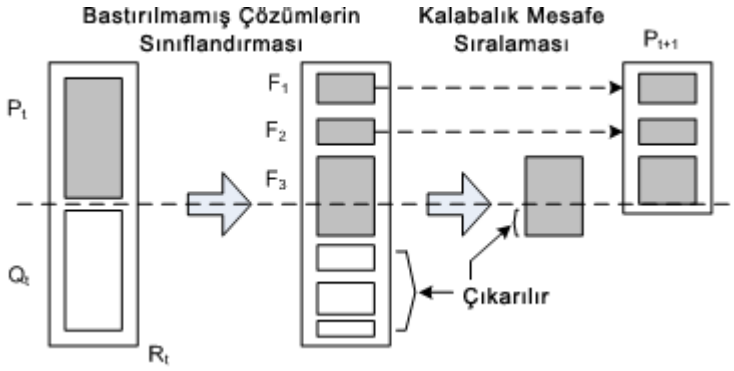
$\underline{\text{crowding mesafe ataması}(\text{I})}$
$l = |I|$
for each i, set $\text{I}[\text{i}]_{\text{distance}} = 0$
for each objective m
  $I = sort(I, m)$
  $I[1]_{distance} = I[l]_{distance} = \infty$
  $for\ i = 2\ to\ (l-1)$
    $I[i]_{distance} = I[i]_{distance} + (I[i+1].m - I[i-1].m)/(f_m^{\max} - f_m^{\min})$

Şekil 5.5. Kalabalıklık karşılaştırma mesafesi için sözde kod (Deb ve ark., 2002)

burada $I[i].m$, I kümesindeki i. bireyin m. amaç fonksiyon değerini gösterir ve $f_m^{\max}$ ve $f_m^{\min}$ m. amaç fonksiyonunun maksimum ve minimum değerleridir.

5.3.3. Ana döngü

İlk olarak rasgele bir şekilde P_0 ebeveyn popülâsyonu oluşturulur ve popülâsyon bireylerin Pareto üstünlüklerine dayalı olarak sıralanır. Her bir çözümün uygunluğu, kendi bastırılmamışlık seviyesine eşit olarak atanır (1 en iyi seviye, 2 ikinci en iyi seviye ve bu şekilde). Bu yüzden uygunluğun minimizasyon kullandığı varsayılır. Daha sonra, doğal ikili turnuva seçimi, yeniden birleşme ve mutasyon işlemcileri, *N* boyutlu Q_0 popülâsyonunun bireylerini oluşturmak için ebeveyn popülâsyona uygulanır. Seçkinlik işlemi ise, önceki en iyi bastırılmamış çözümlerle mevcut popülâsyon karşılaştırarak gerçekleştirilecektir. Bu yüzden, başlangıç jenerasyonundan sonra prosedür farklı bir şekilde işler. Buna göre, NSGA-II algoritmasının *t.* jenerasyonu Şekil 5.6'da gösterilmektedir (Deb, 2002; Deb ve ark., 2002; Sağ ve Çunkaş, 2009).



Şekil 5.6. NSGA-II bastırılmamışlık sıralaması (Deb, 2002)

NSGA-II prosedürünün adım adım çalışma süreci Şekil 5.7'de görülmektedir. Öncelikle, birleşik popülâsyon $R_t = P_t \cup Q_t$ oluşturulur. Bu popülâsyon 2*N* boyutundadır ve bastırılmamışlığa göre sıralanmalıdır. Önceki popülâsyonun tümü ve mevcut popülâsyonun bireyleri R_t ye eklendiği için, seçkinlik garantilenmiş olur. En iyi bastırılmamış küme olan F_1'e ait olan çözümler, birleşik popülâsyondaki en iyi çözümlerdir ve birleşik popülâsyondaki diğer çözümlerden daha fazla vurgulanmalıdırlar. Eğer F_1'in boyutu *N*'den küçükse, yeni popülâsyon P_{t+1} için F_1

kümesinin tüm üyeleri kesin olarak seçilir. P_{t+1} popülâsyonunun kalan üyeleri, oranlarına göre sonraki bastırılmamış yüzeylerden seçilir. Diğer bir ifadeyle, sırasıyla önce F_2 kümesinden, sonra F_3 kümesinden çözümler alınır ve bu şekilde devam edilir. Bu işlem P_{t+1} popülâsyonu doluncaya kadar devam eder. Genelde, F_1'den F_l'e tüm kümelerde çözümlerin sayısı popülâsyon boyutundan daha geniştir. Kesin olarak N popülâsyon üyesi seçmek için, son yüzey F_l'nin çözümlerini kalabalık-karşılaştırma operatörünü $\prec_n$ kullanarak azalan düzende sıralanır ve tüm popülasyon boşluklarını doldurmak için ihtiyaç duyulan en iyi çözüm seçilir. N boyutlu yeni popülâsyon P_{t+1}, yeni bir N boyutlu Q_{t+1} popülasyonunu oluşturacak seçim, çaprazlama ve mutasyon için kullanılır. Burada kalabalık-karşılaştırma operatörüne dayalı ikili turnuva seçim operatörü kullanılmaktadır.

$$
\begin{aligned}
&R_t = P_t \cup Q_t \\
&F = fast-non-do\min ated-sort(R_t) \\
&P_{t+1} = \phi \ ve \ i = 1 \\
&until \left|P_{t+1}\right| + \left|F_i\right| \le N \\
&\qquad crowding-mesafe-ataması(F_i) \\
&\qquad P_{t+1} = P_{t+1} \cup F_i \\
&\qquad i = i+1 \\
&Sırala(F_i \prec_n) \\
&P_{t+1} = P_{t+1} \cup F_i\left[1:\left(N-\left|P_{t+1}\right|\right)\right] \\
&Q_{t+1} = yeni-pop-yap\left(P_{t+1}\right) \\
&t = t+1
\end{aligned}
$$

Şekil 5.7. NSGA2 ana prosedürü (Deb, 2002)

5.4. Bastırılmamışlık Sıralamalı Parçacık Sürü Optimizasyonu (NSPSO)

Çok-amaçlı optimizasyon problemlerinin çözümüne yönelik olarak, temel PSO algoritması bastırılmamışlık sıralama stratejisi ile birleştirildi. Böylece Pareto-temelli yeni bir çok amaçlı optimizasyon algoritması geliştirildi. Önerilen algoritma bastırılmamışlık sıralamalı parçacık sürü optimizasyonu (NSPSO) olarak isimlendirildi.

NSPSO algoritmasında her çevrim sonunda popülasyonun iki katı boyutunda bir çözüm kümesi oluşturulur. 2N boyutlu bu popülasyon; popülasyonu oluşturan N adet

çözüm ve bu çözümlerden yeni üretilen N adet çözümün birleştirilmesiyle oluşur. Daha sonra tüm çözümler, popülasyondaki çeşitliliği koruyacak şekilde sıralanır. Sıralama işleminde literatürde sıklıkla kullanılan iki ölçüt vardır. Bunlardan birisi Goldberg ve Richardson (1987) tarafından önerilmiş olan hücreleme (niching) tekniğidir. Bir diğeri ise Deb (2002) tarafından önerilmiş olan kalabalıklık mesafesi olarak bilinir. NSPSO algoritması opsiyonel olarak bu iki metottan herhangi birini kullanarak çözümleri sıralar. 2N boyutlu popülasyonda bulunan parçacıklar, bastırılmamış parçacıkların depolandığı harici arşiv kümesindeki çözümler ile karşılaştırılır ve harici arşiv güncellenir. Böylece önceki popülasyonda bulunan tüm en iyi çözümlerin korunması sağlanır ve güçlü bir seçkinlik stratejisi uygulanmış olur. Sonraki çevrime geçilmeden önce, harici arşivden rasgele seçilen N adet çözüm yeni popülasyona aktarılır. Harici arşiv boyutunun popülasyon boyutundan (N) küçük olması durumunda, bastırılmamışlık seviyelerine göre sıralanmış kalan popülasyon içinden alınarak tamamlanır. Öncelikle bastırılmamışlık derecesi 1 olan çözümler alınır. Daha sonra bu çözümler popülasyondan çıkarılır ve kalan popülasyondaki yeni bastırılmamış çözümler diğer bir ifadeyle bastırılmamışlık derecesi 2 olan çözümler bulunur. Sırasıyla N adet çözüm tamamlanıncaya kadar işleme devam edilir. Standart PSO'da parçacıkların güncellenmesi iki en iyi değer olan pbest ve gbest'e bağlı olarak gerçekleştirilirken; NSPSO algoritmasında parçacıklar, harici arşiv kümesi ile güncellenir. NSPSO algoritmasının çalışması Şekil 5.8. ile özetlenmiştir.

5.5. Bölüm Değerlendirmesi

Bu bölümde, öncelikle çok-amaçlı optimizasyon konusunun çalışma prensipleri ve kavramsal yapısı üzerinde durulmuştur. Çok-amaçlı optimizasyon problemlerini çözmek için önerilen yollar hakkında açıklayıcı ve sınıflandırıcı bilgiler verilmiştir. Bu bağlamda; ilk olarak iktisat bilimleri literatüründe ortaya konulan Pareto-Optimal kavramının çok-amaçlı optimizasyon problemlerine uyarlanmasıyla bazı sezgisel algoritmalar geliştirilmiştir. Bu algoritmalar deterministik olmayan çok-amaçlı problemlerin çözümünde etkin bir şekilde kullanılan en başarılı araçlardır. Bu nedenle; Pareto-Optimal kavramının tanımı ve metodoloji kısaca açıklanmıştır. Konunun teorik çerçevesi detaylı bir şekilde açıklandıktan sonra, görüntü segmentasyonunun çok-amaçlı bir optimizasyon problemi olarak nasıl ele alınabileceği tartışılmıştır. Daha sonra,

1. Popülasyonu oluştur ve *PSOList* listesine depola.

  a. i. parçacığın konumunu X_i ve hızını V_i belirlenmiş karar değişkenlerinin değer aralığında rasgele reel sayılar uzayında üret. V_i farklı doğrultularda 0.5 olasılıkla üretilebilir.

  b. Başlangıç olarak her parçacığın en iyi değeri olan *Pi* değeri X_i olarak ayarla ve V_{MAX} parametresini karar değişkenlerinin alt ve üst değerlerine göre ayarla

  c. Popülasyondaki tüm parçaıkları değerlendir, çevrim sayacını $t=0$ olarak ayarla.

2. t := t + 1

3. Popülasyondaki bastırılmamış çözümleri belirle ve *nonDomPSOList* listesinde depola.

4. Tüm parçacıklar için hesapla - a) niche count, veya b) kalabalıklık mesafesi

5. *nonDomPSOList* listesini seçilen ölçüte göre yeniden sırala.

  ölçütler a) niche counts, veya b) kalabalıklık mesafesi.

6. *For(i*=0; i < *numParticles*; i++) (step through *PSOList):*

  a. Sıralanmış *nonDomPSOList* listenin üst bölümünden (örneğin ilk %5 lik kısmından) i. parçacık için rasgele bir global en iyi P_n çözümünü seç.

  b. Yeni hız V_i değerini denklem(1)'e göre; yeni X_i değerini denklem(2)'ye göre hesapla.

  c. i. parçacık *Pi* yi ve yeni üretilen X_i parçacığını sonraki popülasyona aktarılmak üzere geçici bir listede depola. Böylece tutulan *nextPopList* geçici listesi döngü sonunda 2N lik boyuta sahip olacaktır.

  d. if *i* < *numParticles* ise (a) adımına geri dön.

7. *nextPopList* deki bastırılmamış çözümleri tespit et ve *nonDomPSOList* içinde depola.

8. Sonraki çevrimde kullanılmak üzere *PSOList* listesini boşalt.

9. *nonDomPSOList* listesinden boyulasyon boyutu kadar parçacığı rasgele seç ve yeni *PSOList* I oluştur.

10. Eğer *PSOList boyutu* < *numParticles* ise aşağıdaki adımları tekrarla;

  a. *nonDomListRest* listesindeki bastırılmamış çözümleri belirle ve *nextNonDomList içinde depola*.

  b. Eğer *PSOList size* < *numParticles ise, nextNonDomList* listesinin elemanlarını *PSOList'*e ekle.

  c. *nextPopListRest* listesini n*extPopListRestCopy* listesine kopayla, sonar *nextPopListRest* listesini resetle.

  d. *nextPopListRest* listesinde boş kalan alanları *nextPopListRestCopy* listesini kullanarak tamamla.

  e. Eğer *PSOList boyutu* < *numParticles* ise (a) adımına git.

11. Eğer *t* < *maxIterations* ise Adım 2'ye git.

Şekil 5.8. NSPSO algoritmasının sözde kodu

Pareto-temelli sezgisel algoritmalar arasında en yaygın olarak kullanılan bastırılmamışlık sıralamalı genetik algoritma II (NSGA-II) algoritması açıklanmıştır. NSGA-II literatürde bulunan birçok algoritmanın esin kaynağı olması ve çok-amaçlı optimizasyon alanındaki literatürün oluşmasındaki katkılarından dolayı son derece önemlidir. Son olarak; tek amaçlı temel PSO algoritması, NSGA-II algoritmasının çalışma prensiplerinden faydalanarak çok-amaçlı optimizasyon problemlerine uyarlanmış ve PSO-temelli yeni bir çok-amaçlı optimizasyon algoritması geliştirilmiştir. Önerilen hibrid yöntem bastırılmamışlık sıralamalı parçacık sürü optimizasyonu (NSPSO) olarak isimlendirilmiştir. Ayrıca tez çalışması kapsamında sonraki bölümlerde NSPSO ve NSGA-II algoritmalarının karşılaştırılmalarına yer verilmektedir.

6. ABC TEMELLİ ÇOK AMAÇLI OPTİMİZASYON ALGORİTMASI

Bu çalışmada, çok amaçlı optimizasyon problemlerinin çözümü için yapay arı kolonisi üzerinde durularak yeni melez bir algoritma geliştirilmiştir. Bu algoritma, başlangıç popülasyonunu üretmek için bir çeşitlendirme fonksiyonu kullanılarak başlatılır. Önerilen algoritmada; Pareto-üstünlüğü, kalabalıklık mesafesi (crowding distance) ve haricen arşivleme gibi çok-amaçlı optimizasyon kavramlarından faydalanılmaktadır. NSGA-II ile ABC algoritmalarının temel çalışma mekanizmalarını birleştirilmektedir. Ayrıca Pareto yüzeye yakınsamayı hızlandıran bir iyileştirme prosedürü de içerir. Bu iyileştirme prosedürü, işçi ve gözcü arı fazlarının her ikisine de bir yerel arama mekanizması olarak eklenmiştir. Önerilen algoritma çok amaçlı optimizasyon için iyileştirilmiş arı kolonisi algoritması (Improved Bee Colony Algorithm for Multi-Objective Optimization – IBMO) olarak isimlendirilmiştir.

Bu bölümde IBMO algoritmasının çalışması detaylı bir şekilde açıklandıktan sonra, sırasıyla çalışmada kullanılan performans indeksleri anlatılacak, deneysel sonuçlar sunulacak ve bölüm değerlendirmesi yapılacaktır.

6.1. İyileştirilmiş Arı Kolonisi Optimizasyon Algoritması (IBMO)

Önerilen yöntem; temel tek amaçlı ABC algoritmasının çok-amaçlı optimizasyon problemlerinin çözümüne uyarlanması ile elde edilmiş yeni melez bir algoritmadır. Bu yöntemde; NSGA-II algoritmasının bastırılmamışlık sıralama stratejisinden ve Pareto-temelli çok-amaçlı optimizasyonun temel kavramlarından faydalanılmaktadır. IBMO olarak isimlendirdiğimiz yöntemin çalışma adımları Şekil 6.1'de verilmiştir. Algoritmanın işleyişi aşağıda adım adım açıklanmaktadır:

Adım #00 – Başlangıç Durumuna Getirme: Bu adımda algoritmanın kontrol parametrelerinin değer atamaları yapılır. Bu parametreler, doğa-esinli algoritmanın analojisine uygun olarak gerekli veri yapısını kurmak için kullanılan sabit değerlerdir. Bu parametreler: koloni boyutu (*NP*), harici arşiv boyutu (*nEXA*), azami fonksiyon değerlendirme sayısı (*maxFes*), azami çevrim sayısı (*MNC*) ve iyileştirme metodu için gerekli olan maksimum adım sayısı (*limit*) olarak kullanılır. Önerilen arı kolonisinde eşit sayıda işçi ve gözcü arı bulunur. Dolayısıyla toplam arı sayısının *NP* algoritmanın başlangıcında tanımlanması gereken bir parametredir. Bu değer fiilen arama uzayının

karmaşıklığına, değişkenlerin ve amaç fonksiyonlarının sayısına bağlıdır ve probleme göre belirlenmelidir. Kalabalıklık mesafesi ile revize edilen harici arşiv, o ana dek bulunan bastırılmamış çözümleri depolamak için kullanılır.

00 **Algoritmanın kontrol parametrelerini başlangıç değerlerini ayarla:**

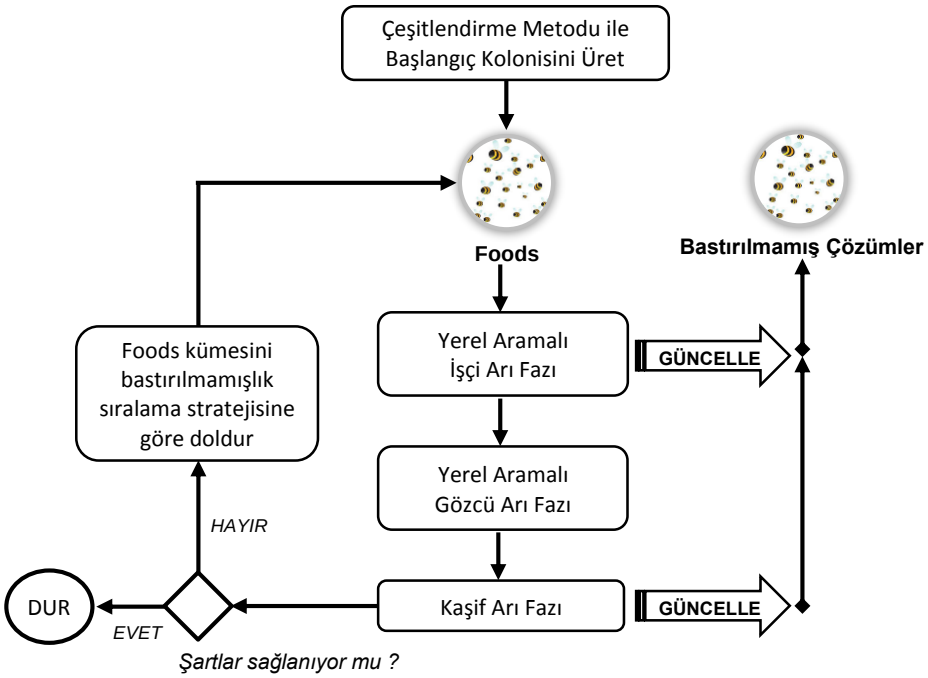
NP:	*NP* = Koloni boyutu = işçi arı + gözcü arı Dolaysıyla *SN = işçi arı sayısı = gözcü arı sayısı = NP / 2*
maxFes veya *MNC*:	Maksimum fonksiyon değerlendirme sayısı veya çevrim sonlanma ölçütü olarak kullanılır
nEXA:	Harici arşiv boyutu
limit:	İyileştirme metodunun tekrarlanma sayısı

01 Kolonideki arıların başlangıç konumlarını Şekil 5.3.'te detaylı bir şekilde açıklanan çeşitlendirme metodunu kullanarak üret.
02 Başlangıç kolonisini değerlendir
03 Başlangıç kolonisindeki bastırılmamış çözümleri bul ve *EXA*'da sakla
04 **REPEAT**
05 **for each** işçi arı **{**
06 *Foods* kümesinden rasgele bir komşu çözüm seç
07 Şekil 6.4.'te açıklanan frekans değerine bağlı olarak değiştirilecek parametreyi belirle
08 Denklem (6.3)'ü kullanarak yeni çözümü u_i üret
09 Şekil 6.5.'te açıklanan metodu kullanarak yeni üretilen çözümü iyileştirmeye devam et
10 **} // end of for each**
11 *EXA* kümesini değişen *Foods* kümesindeki çözümleri kullanarak Güncelle
12 *EXA* nın kalabalık mesafesine değerlerine göre çözümlerin seçilme olasılıklarını hesapla
13 **for each** gözcü arı **{**
14 Hesaplanan olasılık değerlerine göre *EXA* dan bir komşu çözüm seç
15 Şekil 6.4.'te açıklanan frekans değerine bağlı olarak değiştirilecek parametreyi belirle
16 Denklem (6.3)'ü kullanarak yeni çözümü u_i üret
17 Şekil 6.5.'te açıklanan metodu kullanarak yeni üretilen çözümü iyileştirmeye devam et
18 **} // end of for each**
19 **for each** kâşif arı **{**
20 **Eğer** limit >= limit_count || Constraint > 0
21 Bu kaynağı terk et ve rasgele yeni bir çözüm üret
22 **} // end of for each**
23 EXA kümesini değişen Foods kümesindeki çözümleri kullanarak Güncelle
24 EXA ve Foods kümeleri bastırılmamışlık sıralama stratejisine göre yüzey yüzey sırala ve sonraki çevrim için birleştirilmiş bu kümeden yeni Foods kümesini doldur
24 **UNTIL** sonlanma koşulu sağlanmadı mı?

Şekil 6.1. IBMO algoritmasının sözde-kodu

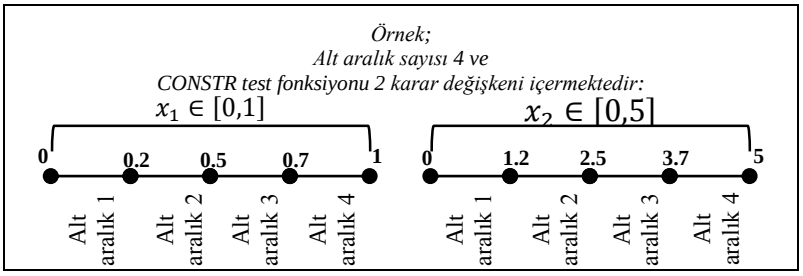
Seçkinliğin korunması bakımından son derece önemli bir role sahiptir. Ancak arşiv boyutu *nEXA* yakınsamayı hızlandırmak için sınırlandırılmalıdır. *maxFes* ve *MNC* algoritmayı sonlandırmak için kullanılan kontrol parametreleridir. *limit* ise iyileştirme yönteminin en fazla kaç defa uygulanabileceğini gösterir.

Algoritmanın çalışması şematik olarak Şekil 6.2'de görülmektedir.



Şekil 6.2. IBMO algoritmasının akış diyagramı

Adım #01 – Çeşitlendirme Metodu ile Başlangıç Popülasyonun Üretilmesi: Algoritmada kullanılmak üzere; ilk üretilecek olan besin kaynaklarının pozisyonları diğer bir ifadeyle, başlangıç aday çözümlerinin parametre değerleri literatürde (Nebro ve ark., 2008; Tang ve Wang, 2013; Glover ve ark., 2003) de kullanılan bir çeşitlendirme tekniği ile tayin edilir. Bu sayede arama uzayı içinde iyi-dağıtılmış bir popülasyonun elde etme olasılığı arttırılmış olur. Bu teknikte bir çözümün tüm parametreleri eşit boyutlu alt aralıklara bölünür. Bu durum Şekil 6.3'de gösterilmektedir.



Şekil 6.3. Çeşitlendirme metodu

Her bir alt-aralığa bir frekans değeri atanır. Bu frekans değerleri başlangıçta sıfırdır. Frekans değerleri ile ters orantılı olarak bir rulet tekerleği oluşturulur ve değeri belirlenecek parametre için bir alt-aralık seçilir. Daha sonra parametrenin değeri seçilen alt-aralıkta denklem (6.1)'e göre rasgele bir şekilde üretilir. Bu işlem yeni üretilecek bir çözümün tüm parametreleri için tekrarlanır.

$$x_{ij} = x_{min}^{j} + rand(0,1)\left(x_{max}^{j} - x_{min}^{j}\right) \tag{6.1}$$

Burada x_i - $i = \{1, 2, \dots SN\}$ olmak üzere popülasyondaki $i.$ çözümü gösterir. SN popülasyonun boyutudur. $j = \{1, 2, \dots D\}$ olmak üzere D bir çözümdeki karar değişkenlerinin (parametrelerin) sayısını ifade eder. Seçilen alt-aralığa göre x_{min}^{j} $j.$ parametrenin alt sınırı; x_{max}^{j} ise üst sınırı anlamındadır.

Adım #02 – Uygunlukların Değerlendirilmesi: Popülasyonda yeni üretilen tüm aday çözümlerin uygunluk değerleri bu adımda hesaplanır. Tüm amaç fonksiyonlar için uygunluk değerleri ayrı ayrı denklem (6.2) kullanılarak bulunur.

$$fit_{in} = \begin{cases} 1/(1 + f_{in}) & f_{in} \geq 0 \\ 1 + abs(f_{in}) & f_{in} < 0 \end{cases} \tag{6.2}$$

Burada fit_{in} , $i.$ çözümün $j.$ Amaç fonksiyonuna göre uygunluk değerini gösterir. $n = \{1, 2, \dots, N\}$ olmak üzere N amaç fonksiyonlarının sayısıdır.

Adım #03 : Pareto üstünlük kavramına göre aday çözümlerin hesaplanmış uygunluk değerleri kullanılarak; mevcut popülasyondaki bastırılmamış çözümler tespit edilir ve harici arşivde depolanır.

Adım #04 : Algoritmanın ana döngüsü bu adımda başlar ve sonlandırma ölçütü sağlanıncaya kadar devam eder. Bu ölçüt belli bir çevrim sayısı olabileceği gibi belirlenmiş bir fonksiyon değerlendirme sayısı da seçilebilir.

Adım #05 - #10 – İşçi Arı Fazı: Tek amaçlı ABC algoritmasında yeni ve eski aday çözümler arasında bir açgözlü seçim mekanizması kullanılır. Önerilen çok-amaçlı ABC algoritmasında ise o ana dek bulunmuş olan en iyi çözümler harici arşivde

saklanır. Sonraki iterasyona aktarılacak olan çözümler, mevcut popülasyon ve harici arşivde tutulan bastırılmamış aday çözümler arasından NSGA-II'nin bastırılmamışlık sıralama stratejisi kullanılarak belirlenir. Böylece Pareto-yüzey üzerindeki seçim basıncı arttırılır.

Orijinal ABC algoritmasında işçi arılar kovana nektar taşımaktan ve buldukları besin kaynakları hakkında kovanda diğer arılarla bilgi paylaşmaktan sorumlu görevli arılardır. Her besin kaynağından sorumlu olan bir işçi arı vardır ve bu kaynakların hepsi ayrı bir aday çözümü temsil eder. Bir işçi arı, komşuluğunda bulunan kaynaklardan birini rasgele belirler. Bu komşu kaynak, işçi arı fazında *Foods* kümesinden yani mevcut popülasyon içinden seçilir. Daha sonra denklem (5.3) kullanılarak mevcut çözümün komşuluğunda yeni bir çözüm üretilir. Üretilen yeni çözüm mevcut çözümün bir parametresi değiştirilerek oluşturulur.

$$delta = \emptyset_{ij}\left(x_{ij} - x_{kj}\right) \qquad ve \qquad v_{ij} = x_{ij} + delta \tag{5.3}$$

Burada $delta$ yeni parametrenin değişim miktarıdır. $\emptyset_{ij}$ [-1, 1] aralığında rasgele bir sayıdır. v_i üretilen yeni çözümü, $k \in \{1,2,\dots,SN\}$ olmak üzere x_k rasgele seçilen komşu çözümü gösterir. x_{ij} notasyonu $i.$ çözümün $j.$ parametresini göstermek için kullanılır. j değeri $[1, D]$ aralığında rasgele üretilir. Bu rasgeleliği tutarlı bir zemine oturtmak amacıyla çeşitlendirme metodunda kullanılan yaklaşım parametre indekslerine de uyarlanmıştır. Her parametre için seçilme sayısını tutan bir frekans değeri tanımlanır. Bu frekans değeri ile ters orantılı olarak değiştirilecek parametrenin indeksi belirlenir. Böylece bir çözüm içindeki parametrelerin yüksek olasılıkla eşit oranlarda ve rasgele seçilmeleri sağlanmış olur. Bu yöntemin ana hatları Şekil 6.4'te gösterilmiştir.

```
FUNCTION DeğiştirilecekParametreyiBul
00  Parametrelerin frekanslarını ters orantılı şekilde hesapla, reverseFrequency
01  Belirtilen aralıkta rasgele bir sayı üret : [ 1, reverseFrequency ]
02  Eğer toplam reverseFrequency=0 ise, param = 1 yap ve adım 7'ye git
03  While value > reverseFrequency[ param ] {
04      Set value = value - reverseFrequency[ param ]
05      Set param = param + 1
06  } // end of while
07  param'ın frekansını bir arttır
END
```

Şekil 6.4. Değiştirilecek parametrenin belirlenmesinin kaba kodu

Denklem (6.3) ile yeni çözümün üretilmesinin ardından, iyileştirme prosedürüne geçilir. Bu prosedür algoritmanın işleyişine yerel arama mekanizması olarak entegre edilmiştir. Her yeni bir çözüm üretilmesi durumunda devreye girer. Bu aşamada, yeni aday çözüme bir aç-gözlü seçim işlemi uygulanır. Bu işlemde NSGA2 algoritmasında önerilen ve kısıtları hesaba katan baskınlık metodu (*constrained-dominance method*) kullanılır. Hesaplanan delta değeri yeni aday çözüm bir öncekinden daha iyi olduğu müddetçe seçilen parametreye eklenmesine devam edilir. Delta değerinin sıfır olması durumunda artış miktarı [-0.5, 0.5] aralığında rasgele yeniden üretilir. Böylece Pareto-Optimal yüzeye yakınsama hızı arttırılmış olur. Fonksiyon değerlendirme sayısının yerel pik noktalarına takılıp çok fazla artmasını önlemek amacıyla bu işlem en fazla *limit* parametresi ile öngörülen sayıda tekrarlanır. Önerilen iyileştirme prosedürünün sözde kodu Şekil 6.5'te verilmektedir.

00 **Eğer** *delta* = 0 ise set $\boldsymbol{delta} = \boldsymbol{rand}(-\mathbf{0.5}, \mathbf{0.5})$ ***else*** continue
01 Set $newSol = v_i$
02 Set $temp = v_i$
03 Set $limit_count$ = 0
04 **REPEAT**
05 Set $limit_count = limit_count + 1$
06 Set $temp_j = temp_j + delta * limit_count$
07 **if** $temp$ dominates $newSol$; **break**
08 Set $newSol = temp$
09 **UNTIL** $limit_count < limit$
10 $newSol$ 'u *Foods* içinde sakla

Şekil 6.5. İyileştirme metodunun sözde kodu

Yerel arama işleminin her adımının sonunda üretilen yeni çözüm önceki çözüm kısıtlı-baskınlık yöntemine göre karşılaştırılır ve daha iyi olan seçilerek popülasyondaki ile değiştirilir.

Adım #11 – Harici Arşivin Güncellenmesi: Harici arşiv, arama işlemi süresince bulunan bastırılmamış çözümleri koruyarak düzgün yayılmış bir Pareto-Optimal yüzeye ulaşmayı güvence altına almak için kullanır. Böylece algoritmanın Pareto-yüzeyin tamamına yakınsayabilme şansı olur. Bununla birlikte, arşivin eleman sayısı arttıkça yeni elde edilen çözümlerin depolanıp depolanmayacağının kontrol edilmesi hesaplama karmaşıklığından dolayı zorlaşır. Bu yüzden, Pareto yüzeydeki çeşitliliği kaybetmeden

arşiv boyutunu sınırlandırabilen bir güncelleme mekanizmasına ihtiyaç duyulur. Literatürde bu amaçla yaygın bir şekilde kullanılmakta olan iki yöntem bulunmaktadır. Bunlardan birisi: (Knowles ve Corne, 2000) tarafından PAES algoritmasında kullanılmak üzere önerilen hiper-küpler ile oluşturulan adaptif ızgaralama stratejisidir. Diğeri ise kalabalık mesafesine dayanan seçim yöntemidir (Deb, 2002). Her iki yöntemde strateji olarak benzer doğrultudadır. IBMO yönteminde kalabalıklık mesafesine dayanan hesaplama kullanılarak arşiv güncellemesi yapılır.

Kalabalıklık mesafesi bir çözümün etrafındaki çözüm yoğunluğunu gösterir. Bir çözümün tüm amaçlar için her iki doğrultudaki en yakın iki komşusu arasındaki uzaklıkların toplanmasıyla hesaplanır. İki amaçlı bir problem için kalabalıklık-mesafesi Şekil 5.4'de verilmiştir. Çözümler her bir amaç fonksiyonuna göre ayrı ayrı sıralanır. Uç noktalarda bulunan çözümlere olan uzaklığın sonsuz olduğu varsayılır. Böylece uç çözümlerin arşivden çıkarılması önlenmiş olur. Kalabalıklık mesafesi denklem (6.4) ile hesaplanır.

$$CD(bee_k) = \sum_{k=2}^{n-1} \frac{(bee_{k+1}).m - (bee_{k-1}).m}{f_m^{max} - f_m^{min}} \tag{6.4}$$

burada $(bee_{k+1}).m$, $(k+1)$. çözümün m. amaç fonksiyon değerini gösterir.

Bu durumda komşu çözümlerle etkileşim sonucunda değiştirilen yeni popülasyondaki çözümler kümesi *Foods* içindeki her bir çözüm, *EXA* ile karşılaştırılır. Yeni elde edilen bastırılmamış çözümlerin sayısı, *nEXA*'dan küçük ya da eşitse çözümlerin tamamı depolanır. Aksi halde arşiv maksimum kapasitesine ulaştığında, Pareto-yüzeyde çözümlerin en sık bulunduğu diğer bir ifadeyle kalabalıklık mesafesi en yüksek olan çözüm arşivden çıkarılır. Pareto yüzeyde daha az çözümün bulunduğu yeri işaret eden diğer çözüm (kalabalıklık mesafesi daha düşük olan çözüm) arşive dâhil edilir. Böylece Pareto-yüzeyin çeşitliliği korunur ve arşiv boyutunun sabit kalması sağlanır.

Adım #12 – Seçilme Olasılıklarının Hesaplanması: Gözcü arı fazında lider çözümlerin seçim basıncını arttırmak amacıyla bir rulet tekerleği kullanılır. Çok-amaçlı optimizasyon problemleri için lider çözüm, bastırılmamış çözüm demektir. Bastırılmamış çözüm olamaması durumunda ise minimum kısıt değerine sahip çözümler demektir. Bu bağlamda; rulet tekerleğinde kullanılmak üzere ihtiyaç duyulan kümülatif

toplam dizisi bu adımda hesaplanır. Kısıtlı optimizasyon problemlerinin çözümü esnasında, algoritmanın ilk iterasyonlarında hiçbir bastırılmamış çözüm elde edilemeyebilir. Harici arşivin boş olması durumunda, çözümlerin kısıt değerleri kullanılırken; aksi durumda arşivdeki bastırılmamış çözümlerin CD değerleri kümülatif toplam dizisi için kullanılır. Böylelikle algoritma hem Pareto-yüzeyin tamamına odaklanabilir, hem de lider çözümlerin seçilme olasılıklarının yüksek olması sayesinde daha hızlı bir yakınsama elde edilir.

Adım #13 - #18 – Gözcü Arı Fazı: Orijinal ABC modelinde gözcü arılar, işçi arıların danslarını izleyerek besin kaynaklarının kalitesi hakkında bilgi edinirler. Daha sonra bu bilgiye bağlı olarak bir kaynak seçerler. Komşu çözümleri seçmek için oluşturulan rulet tekerleği için çözümlerin uygunluk değerleri kullanılır. Önerilen yöntemde klasik yöntemin aksine komşu çözümler *Foods* kümesi yerine *EXA'dan* seçilir. Seçilme olasılıkları adım #12'de bahsedildiği gibi hesaplanır. Değiştirilecek parametrenin belirlenmesi, yeni çözümlerin üretilmesi ve iyileştirme prosedürü işçi arı fazındaki gibi uygulanır.

Adım #19 - #22 – Kâşif Arı Fazı: İşçi ve gözcü arı fazlarında uygulanan iyileştirme prosedürü boyunca tüm çözümler için bir deneme (trial) sayacı tutulur. Bir çözümün uygunluk değerinde öncekine göre herhangi bir iyileşme sağlanamaması durumunda çözümün deneme değeri bir arttırılır. Aksi durumda bu değer sıfırlanır. Algoritmanın bu adımında; deneme sayacı önceden tanımlanmış *limit* değerini aşan veya kabul edilebilir düzeyin üzerinde bir kısıt değeri olan çözümler (besin kaynakları) terk edilir. Bu kaynakların tükendiği ve daha fazla iyileştirilemeyecekleri varsayılarak; *Foods* kümesinden çıkarılırlar. Bu çözümler yerine aralık dâhilinde rasgele yeni çözümler üretilir.

Adım #23 – Harici Arşivin Güncellenmesi: Adım #11'de yapılan işlem kâşif arı fazı sonunda tekrarlanarak, *EXA*'nın güncellenmesi sağlanır.

Adım #24 - #25 : Bir sonraki çevrimde kullanılacak çözümler seçilerek *Foods* yeniden yapılandırılır. Tam olarak *SN* adet çözüm seçmek için üç durumla karşılaşılır: (1) Harici arşiv boyutu *nEXA*, popülasyon boyutuna *SN* eşit olabilir. Bu durumda *Foods* kümesi tamamen arşivdeki bastırılmamış çözümlerle doldurulur. (2) *nEXA*, *SN'den* büyük olabilir. Bu durumda *EXA*'da tutulan bastırılmamış çözümler içinden rasgele *SN* adet bastırılmamış çözüm seçilir. *Foods* kümesi bu çözümlerle doldurulur. (3) *nEXA*,

SN'den küçük olabilir. Bu durumda ise öncelikle yine harici arşivdeki çözümler yeni oluşturulacak popülasyona kopyalanır. Eksik kalan çözüm adedi (*SN–nEXA*) kadar çözüm mevcut popülasyondan seçilir. Bu amaçla *Foods* kümesindeki çözümler kalabalık-mesafelerine göre artan düzende sıralanırlar. En düşük CD değerine sahip çözümler yeni *Foods* kümesine aktarılır. Böylece o ana dek bulunmuş en iyi çözümlerin tamamı sonraki çevrim işlemine aktarılarak seçkinlik korunmuş olur. Sonlandırma ölçütü sağlanıncaya kadar, algoritma adım #06'ya dönerek işçi arı fazından çalışmaya devam eder.

6.2. Deneysel Sonuçlar

IBMO'nun performansını değerlendirmek için literatürde en iyi olarak bilinen dört farklı algoritma ile karşılaştırma yapılmıştır. Bunlar sırasıyla: NSGA-II, SPEA2, OMOPSO ve AbYSS algoritmalarıdır. IBMO ve seçilen algoritmalar çok-sayıda test problemi üzerinde eşit koşullarda ve sonlandırma ölçütü olarak fonksiyon değerlendirme sayısı dikkate alınarak çalıştırılmıştır. Bu bölümde öncelikle kullanılan test problemlerinin açıklamaları verilmiştir. Daha sonra 4 farklı performans göstergesi detaylı şekilde anlatılmıştır. Ayrıca IBMO ve diğer algoritmaların parametre ayarlamaları ayrı bir alt başlık olarak ele alınmıştır. Son olarak, elde edilen deneysel sonuçların ortanca (median) ve çeyrekler açıklığı (interquartile ranges) değerleri tüm göstergeler için hesaplanmış ve tüm test problemleri için karşılaştırmalı olarak analizleri yapılmıştır. Tüm algoritmalar Intel core I7-2600 3.40Ghz CPU, 10GB bellek ve Windows işletim sistemi kurulu bir kişisel bilgisayar üzerinde çalıştırılmıştır.

6.2.1. Test Problemleri

Genel bir performans değerlendirmesi ortaya koyabilmek için yapay olarak oluşturulmuş test fonksiyonlarını kullanmak uygun bir yaklaşımdır. Çünkü bu fonksiyonlar; kolay uygulanabilir olmaları, kolay bir şekilde görselleştirilebilir olmaları ve gerçek Pareto-Optimal yüzeylerinin önceden biliniyor olması gibi birçok avantaja sahiptirler. Ancak literatürde farklı şekillerde tasarlanmış çok sayıda test fonksiyonu bulunmaktadır. Çalışmada karşılaştırmaların doğruluğunu ortaya koymak açısından 26 farklı test problemi kullanılmıştır. Bu problemler kısıtlı ve kısıtsız çok sayıda fonksiyondan oluşmaktadır.

Test problemleri 4 kategoride ele alınabilir Birincisi 2-amaçlı kısıtsız problemlerdir: Schaffer (Schaffer, 1985), Fonseca (Fonseca ve Flemming, 1998), Kursawe (Kursawe, 1990) ve Zitzler-Deb-Thiele'nin ZDT test paketi (Zitzler ve ark., 2000).

İkinci kategori 2-amaçlı kısıtlı problemlerdir: Constr (Deb 2002), Srinivas (Srinivas ve Deb, 1995), Osyczka2 (Osyczka ve Kundu, 1995), Golinski (Kurpati ve Azarm, 2002), Tanaka (Tanaka ve ark., 1995) ve Binh2 (Binh ve Ulrich, 1997).

Üçüncü kategori Deb ve arkadaşlarının oluşturduğu amaç fonksiyon sayıları istenilen bir değere uyarlanabilen DTLZ test paketidir (Deb ve ark., 2002). Bu çalışmada ilk 7 tanesi kullanıldı. Ancak, bu test paketi toplam 9 test problemi içermektedir. DTLZ8 ve DTLZ9 yan kısıtlamalar ve yüksek boyutlarından dolayı genellikle göz ardı edilir.

Son olarak dördüncü kategori test problemleri Viennet test paketi (Viennet ve ark. 1996) ve Tamaki (Tamaki ve ark., 1996) test probleminden oluşur. Viennet1, Viennet2 ve Viennet3 3-amaçlı kısıtsız test fonksiyonlarıdır. Bunun yanı sıra Viennet test problemleri diğerlerin farklı olarak ayrık Pareto yüzeye sahip fonksiyonlardır. Tamaki problemi de 3-amaçlı ve tek kısıtlı bir problemdir. Tüm test problemlerinin formülleri Ek-1'de verilmiştir.

6.2.2. Performans Göstergeleri

Çok-amaçlı optimizasyon algoritmalarının performansını nicel olarak değerlendirmek için belirlenen iki temel kavram bulunmaktadır. Birinci kavram, gerçek Pareto-Optimal yüzey ile elde edilen bastırılmamış çözümler arasındaki uzaklıkla ilgilenir ve yakınsama olarak tanımlanır. Yakınsama, bir algoritma tarafından bulunan sonuçların doğruluğunu yansıtan bir faktör olduğu için çok önemlidir. Çeşitlilik olarak bilinen ikinci kavram, bulunan çözümlerin farklılığını ölçmek ve Pareto-Optimal yüzey üzerinde ne ölçüde tutarlı bir dağılım gösterdiklerini anlamak için hesaplanır. Literatürde bu husus başlı başına ayrı bir çalışma alanı olarak ele alınmıştır. Bu kavramları ölçmek için geliştirilen birçok metot önerilmiştir (Deb, 2002; Zitzler ve Thiele, 1999; Tanaka ve ark., 1995). Bu çalışmada en sık kullanılan 4 performans göstergesi tercih edilmiştir ve burada kısa açıklamalarına yer verilmiştir.

6.2.2.1. Genel Uzaklık Göstergesi (GD)

En çok bilinen yakınsama göstergesidir. Elde edilen bastırılmamış çözümlerin gerçek Pareto yüzeye olan uzaklıklarının ortalaması şeklinde hesaplanır. Olması istenen en ideal durum değerinin sıfır olmasıdır. Matematiksel olarak denklem (6.5)'teki gibi tanımlanır (Binh ve Ulrich, 1997).

$$GD = \frac{\sqrt{\sum_{i=1}^{nEXA} {d_i}^2}}{nEXA} \tag{6.5}$$

Burada $nEXA$ bastırılmamış çözümlerin sayısını, d_i ise harici arşivdeki i. çözümün Pareto yüzeydeki kendisine en yakın elamana olan Öklid uzaklığını gösterir.

6.2.2.2. Yayılma Göstergesi (SP)

Bulunan bastırılmamış çözümlerin Pareto-Optimal yüzey boyunca dağılımını ölçmek için kullanılır. Normalleştirilmiş amaç uzayında birbirlerine en yakın ardışık iki çözüm arasındaki mesafenin hesaplanmasına dayanır. SP göstergesinin sıfıra yakınsaması istenir. Matematiksel olarak denklem (6.6)'daki gibi ifade edilir (Li, 2003).

$$SP = \frac{\sum_{i=1}^{n} d(e_i,S)+\sum_{X\in S^*}|d(X,S)|-\bar{d}}{\sum_{i=1}^{n} d(e_i,S)+|S^*|.\bar{d}} \tag{6.6}$$

$$d(X,S) = \min_{Y\in S, Y\neq X} \|f(X) - f(Y)\|^2$$

$$\bar{d} = \frac{1}{|S^*|}\sum_{X\in S^*} d(X,S)$$

Burada S bir algoritma tarafından elde edilen bastırılmamış çözümler kümesini, S^* ise önceden bilinen gerçek Pareto-Optimal çözüm kümesini gösterir. $(e_1, \dots, e_n)$ olmak üzere S^* kümesindeki uç çözümlerdir ve n amaç fonksiyonlarının sayısıdır. $d(X,S)$ bir X çözümünün gerçek Pareto yüzeye olan minimum mesafesidir. $\bar{d}$ ortalama uzaklık anlamına gelir.

6.2.2.3. Maksimum Yayılma Göstergesi (MS)

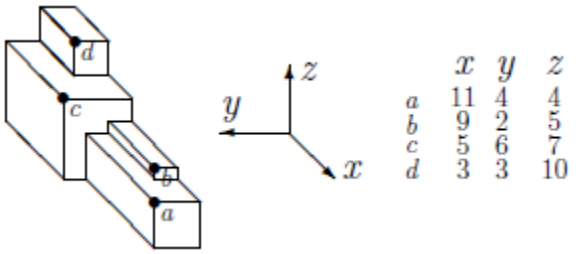
MS bir diğer çeşitlilik göstergesidir (Deb ve ark., 2001). Bu gösterge özellikle algoritmanın uç çözümleri bulabilmedeki başarısını ortaya koyar. Matematiksel olarak ifadesi denklem (6.7)'de verilmiştir.

$$MS = \sqrt{\frac{1}{n}\sum_{l=1}^{n}\delta_l} \quad \text{ve} \quad \delta_l = \left(\frac{min\left(f_l^{max},F_l^{max}\right)-max\left(f_l^{min},F_l^{min}\right)}{F_l^{max}-F_l^{min}}\right)^2 \tag{6.7}$$

Burada f_l^{max} ve f_l^{min} sırasıyla harici arşivdeki l. amaç fonksiyonunun maksimum ve minimum değerleridir. F_l^{max} ve F_l^{min} ise gerçek Pareto yüzey üzerinde l. amaç fonksiyonunun maksimum ve minimum değerlerini gösterir. $f_l^{min} \geq F_l^{max}$ olması durumunda δ_l değerinin sıfır olduğu varsayılır. MS göstergesinin bire yakınsaması istenen durumdur. Göstergenin tam 1 değerini alması, algoritmanın tüm yüzeye tam olarak yayılabildiğini gösterir.

6.2.2.4. Hypervolume Göstergesi (HV)

HV metriği performans değerlendirmesi için Zitzler ve Thiele tarafından önerilmiş popüler bir göstergedir. S-metriği veya Lebesgue ölçütü olarak da bilinir (Viennet ve ark., 1996). Bu göstergeye göre, elde edilen bastırılmamış çözüm kümesi amaç uzayda bulunan noktalardır. HV metriği ile arama uzayının bu çözümler ile bastırılan bölgesinin boyutu hesaplanır. Gösterge değeri ne kadar büyük olursa bastırılan alanın boyutu o kadar çok olacağı için daha iyi bir yüzey elde edildiği anlaşılır. Bu sebeple normalleştirilmiş amaç fonksiyon değerlerini kullanır. 3-boyutlu bir HV göstergesinin modellenmesi Şekil 6.6'da gösterilmektedir.



	x	y	z
a	11	4	4
b	9	2	5
c	5	6	7
d	3	3	10

Şekil 6.6. 3-boyutlu bir hypervolume

Burada Q = {a, b, c, d} bastırılmamış çözümler dairelerle işaretlenmiştir. Göstergenin matematiksel ifadesi denklem (6.8)'de verilmiştir.

$$HV = volume\left(\bigcup_{i=1}^{|Q|} v_i\right) \tag{6.8}$$

6.3. Parametre Ayarları

Tüm algoritmalar 25.000 ve 50.000 fonksiyon değerlendirmesi için 100'er defa bağımsız olarak ve her bir problem için ayrı ayrı çalıştırılmıştır. Tüm problemlerin ölçüm değerleri (merkezi eğilimleri) için ortanca $\tilde{x}$ değerleri ve istatistiksel dağılımları için ise çeyrekler açıklığı (*IQR*) değerleri hesaplanmıştır. Test problemleri içinde en iyi değere sahip sonuçlar koyu gri zemin rengi ile belirginleştirilmiştir. Eğer birden fazla algoritma aynı en iyi değere ulaşabildiyse, daha küçük *IRQ* değerine sahip olan algoritma en iyi olarak kabul edilmiştir. Arı kolonisinin boyutu *NP* 100 olarak ayarlandı. Dolayısıyla popülasyon boyutu 50 oldu. Harici arşivin boyutu *nEXA* 2-amaçlı problemler için 100; 3-amaçlı problemler için 400 olacak şekilde ayarlanmıştır. Son olarak iyileştirme prosedürünün azami adım sayısını belirleyen limit parametresinin değeri 5 olarak atanmıştır. Karşılaştırılan diğer algoritmaların aksine IBMO ayarlanması gereken daha az sayıda kontrol parametresine sahip olmasından dolayı avantajlıdır. Ayrıca bu parametreler ve kullanılan değerler Çizelge 6.1'de verilmiştir.

Çizelge 6.1. IBMO'nun kontrol parametreleri

Parametre	Değeri	
maxFes	25000 ve 50000	
NP	100	
SN	50	
nEXA	100	iki – amaçlı problemler için
	400	2′den fazla amaç fonksiyonlu problemler için
Limit	5	

IBMO, Matlab ortamında geliştirilmiş ve literatürün önde gelen en iyi 4 algoritması ile karşılaştırılmıştır. Bu algoritmalar: NSGA2 (Deb, 2002), SPEA2 (Zitzler ve ark., 2001), AbYSS (Nebro ve ark., 2008) ve OMOPSO (Niu ve Shen, 2007). Bu

algoritmaların hepsi önerildikleri orijinal makalelerde tavsiye edilen parametre değerleri baz alınarak test problemlerine uygulanmıştır.

Elde edilen deneysel sonuçlar sezgisel algoritmalar tarafında üretildikleri için, istatistiksel güvenilirlik analizleri kullanılarak sonuçların uygulanabilir bir güven aralığında olup olmadıkları kontrol edilmiştir. Bu amaçla ilk olarak üretilen verilerin normal dağılım gösterip göstermediğini tespit etmek için Kolmogorov-Smirnov testi uygulanmıştır. Verilerin normal dağılım göstermesi durumunda One-Way ANOVA testi yapılmıştır. Aksi halde ise Kruskal-Wallis testi uygulanmıştır. Tüm istatistiksel testlerde güven aralığı olarak 0.05 değeri kullanılmıştır. Sonuçların çoklu karşılaştırma işlemleri SPSS programı yardımıyla yapılmıştır. Sonuçlar Çizelge 6.2–6.3–6.4 ve 6.5'de verilmiştir. Başarılı sonuçlanan testler sonuç çizelgelerinin son sütununda "+" sembolü ile işaretlenirken; başarısız olanlar "-" sembolü ile gösterilmiştir. Başarısızlık durumu algoritmanın istatistiksel olarak güven vermeyen bir rasgelelikle çalıştığı anlamına gelir.

6.4. Gösterge Sonuçları ve Tartışma

Çok-amaçlı optimizasyon problemlerinin çözümü için önerilen IBMO algoritması; literatürde iyi-bilinen ve standart olarak kabul gören iki algoritma (NSGA-II ve SPEA2) algoritmaları ve son senelerde geliştirilmiş diğer iki başarılı algoritma (OMOPSO ve AbYSS) ile karşılaştırılmıştır. 25.000 fonksiyon değerlendirmesi sonunda alınan değerlerin performans göstergelerinin hesaplanmasıyla oluşturulan sonuçlar GD, SP, MS ve HV göstergeleri için sırasıyla Çizelge 6.2, Çizelge 6.3, Çizelge 6.4 ve Çizelge 6.5'de verilmiştir. Bu sonuçların yanı sıra 50.000 fonksiyon değerlendirmesi için çalıştırılan algoritmaların gösterge değerleri EK-2'de verilmiştir.

Çizelge 6.2'den görüldüğü gibi IBMO; hemen hemen tüm problemler için istatistiksel güven aralığında kalmayı başararak 26 test probleminin 9'unun dışında tüm problemlerde GD göstergesi için en düşük değerlere ulaşmıştır. Ayrıca gerçek Pareto yüzeye tutarlı bir yakınsama sağlarken, iyi yayılmış bir yüzey elde edebilmesi de dikkat çekicidir. IBMO özellikle ilk iki gruptaki test problemleri için daha iyi sonuçlara sahiptir. Bunun yanı sıra, IBMO ikiden fazla amaç fonksiyonuna sahip olan son iki gruptaki problemler için de önemli derecede başarılı sonuçlar elde edebilmiştir.

DTLZ3, DTLZ7, Viennet, Viennet2 ve Tamaki problemleri için diğer algoritmalardan üstün düzeydedir. Geriye kalan diğer problemlerin sonuçları dikkate

alınırsa, önerdiğimiz IBMO algoritması özellikle son zamanlarda iyi-geliştirilmiş bir algoritma olan AbYSS'ın aldığı sonuçlarla karşılaştırıldığında tatmin edici düzeydedir.

Çizelge 6.3'de verilen yayılma göstergesine göre IBMO, diğer algoritmalarla tüm test problemleri için yarışabilen değerler üretmiş ve dört test problemi için diğer tüm algoritmalardan daha başarılı sonuçlar elde edebilmiştir. Unutulmamalıdır ki; çeşitlilik değerlendirmesi, algoritmanın ulaştığı Pareto yüzeyin doğruluğu dikkate alınarak yapılmalıdır. Bu bağlamda SP göstergesinin sonuçları yeniden incelendiğinde, iyi gözüken algoritmalara göre rekabetçi sonuçlar alan IBMO algoritması, genel itibariyle aynı problemler için GD göstergesi açısından daha başarılı olduğu görülmüştür. Bu durum Şekil 6.7-6.8-6.9-6.10 ve 6.11'de grafiksel olarak verilmiştir. Şekil 6.7'den görüldüğü gibi, IBMO kısıtlı ve kısıtsız çok-amaçlı optimizasyon problemleri üzerinde tam olarak doğru Pareto yüzeye ulaşabilmiştir. Devam eden sonraki şekiller, ikiden fazla amaç fonksiyonu olan DTLZ1, DTLZ3, DTLZ4 ve DTLZ7 problemlerine ait grafiksel gösterimlerdir. Şekil 6.8'de gösterilen DTLZ1 problemi için IBMO doğru Pareto-Optimal yüzeyi hemen hemen tamamıyla elde ederken, OMOPSO ve SPEA2 algoritmaları gerçek yüzeyden çok uzaktadırlar. Diğer taraftan NSGA-II ve AbYSS algoritmaları ise yetersiz bir dağılım gösterirler. Şekil 6.9'da gösterilen DTLZ3 problemi için IBMO Pareto yüzeye en fazla yaklaşan algoritmadır. Ayrıca diğerlerine nispeten daha iyi bir dağılım gösterir. Şekil 6.10'da gösterilen DTLZ4 problemi için GD ve SP değerleri de dikkate alınarak bakıldığında; makul bir doğrulukla birlikte en iyi dağılımı SPEA2 algoritması göstermiştir. Şekil 6.11'de verilen DTLZ7 problemi için ise IBMO, OMOPSO, SPEA2 ve kısmen NSGA-II algoritmaları ayrık dört bölgeyi çıkarabilmesine rağmen, AbYSS bu problemin çözümünde başarısız olmuştur. IBMO dışındaki algoritmalar optimizasyon problemlerine göre farklı sonuçlar verebilirken, geliştirilen IBMO algoritması AbYSS ile birlikte ele alınan 5 algoritma arasında en tutarlı ve başarılı sonuçları vermiştir. Önerilen iyileştirme prosedürünün etkisiyle Pareto yüzeyin sınırlarındaki seçim basıncını arttırması sayesinde IBMO'nun test problemleri üzerindeki performansı oldukça iyidir. Ayrıca harici arşivin kalabalıklık mesafesi stratejisi ile güncelleniyor olması çeşitliliğin artmasında etkili olmuştur.

MS gösterge sonuçları Çizelge 6.4'de verilmiştir. Hesaplanan maksimum yayılma göstergesi değerlerine göre IBMO, Binh2 ve DTLZ6 dışındaki 21 test probleminin tamamında uç çözümleri veya büyük ölçüde uç çözümlere çok yakın çözümleri bulabilmiştir.

HV göstergesi bulunan çözümlerin doğru Pareto yüzey boyunca gösterdikleri dağılımla birlikte doğru yüzeye olan yakınlıklarını birleştirebilen etkili bir göstergedir. Diğer bir ifadeyle GD ve SP göstergelerini birleştirerek tek bir değer içinde bulunan sonuçların kalitesini hesaplamaya çalışır. HV göstergesinin hesaplama sonuçları Çizelge 6.5'de görülmektedir. IBMO bu sonuçlara göre 10 problem için en iyi değerlere ulaşırken, OMOPSO dışındaki tüm algoritmalar DTLZ3 ve DTLZ6 problemleri için makul bir sonuç üretemedikleri görülür. Geriye kalan tüm çok-amaçlı optimizasyon problemleri için IBMO, HV göstergesine göre de oldukça rekabetçi ve başarılı sonuçlar alabilmiştir.

GD, SP, MS ve HV göstergelerinin 50.000 fonksiyon değerlendirmesinden sonra alınan değerler için olan hesaplamaları sırasıyla Çizelge Ek-2.1, Ek-2.2, Ek-2.3. ve Ek-2.4.'de verilmiştir. Değerlendirme sayısının artmasından dolayı tüm algoritmaların sonuçlarında belirli bir iyileşme izlenmektedir. Bu çizelge sonuçları dikkate alınarak bir değerlendirme yapılırsa; IBMO algoritmasının yukarıda verilen sonuçlara uygun olarak yine oldukça başarılı olduğu ve diğer algoritmalara göre daha üstün ya da en azından rekabet edebilir düzeyde sonuçlar aldığı ifade edilebilir. Çalışmada kullanılan çok-amaçlı test problemlerinin tanımlamaları Ek-1'de verilmiştir.

6.5. Bölüm Değerlendirmesi

Çok amaçlı optimizasyon problemlerinin çözümüne yönelik olarak yapay arı kolonisine dayanan yeni melez bir teknik önerilmiştir. Çok amaçlı optimizasyon için iyileştirilmiş arı kolonisi optimizasyon algoritması kelimelerinin İngilizce baş harfleri ile (IBMO) olarak ismini kısalttığımız algoritmanın çalışması detaylı bir şekilde açıklanmıştır. IBMO ana hatlarıyla bastırılmamışlık sıralama stratejisini ve çok-amaçlı optimizasyon ilkelerini birleştirir. Klasik tek amaçlı ABC algoritmasını çok-amaçlı optimizasyona uyarlar. IBMO çeşitliliği sağlamak için kalabalıklık mesafesinden yararlanırken; bastırılmamış çözümleri depolamak içinse harici bir arşiv oluşturur. Ayrıca işçi ve gözcü arı fazlarının içine entegre edilen bir iyileştirme prosedürü sayesinde Pareto-Optimal yüzeye yakınsamanın hızlanması sağlanmıştır. Bunların yanı sıra başlangıç popülasyonun üretilmesinde bir çeşitlendirme yönteminin kullanılmıştır. Böylece daha düzgün dağıtılmış bir popülasyon oluşturulması sağlanmıştır. Aynı

çeşitlendirme yöntemi değiştirilecek parametrelerin seçimine de uygulanarak, parametrelerin eşit oranlarda ve yine rasgele seçilmeleri sağlanmıştır.

Önerilen yöntem 4 farklı ve başarılı algoritma (NSGAII, SPEA2, AbYSS ve OMOPSO) ile karşılaştırılmıştır. Tutarlı bir performans değerlendirmesi yapmak için ZDT, DTZL ve Viennet test paketlerini de içeren 26 test problemi üzerinde algoritmalar çalıştırılmıştır. Her bir problem için 100 bağımsız çalıştırma yapılmıştır. Algoritmalar hem 25000 hem de 50000 fonksiyon değerlendirmesi için ayrı ayrı çalıştırılmıştır. Alınan sonuçların kaliteleri 4 farklı performans göstergesi (GD, SP, MS ve HV) kullanılarak değerlendirilmiştir.

Sonuç olarak; kalite göstergelerinden alınan değerler ve IBMO tarafından bulunan Pareto-Optimal yüzeylerin diğer algoritmalar ile karşılaştırmalı analizleri dikkate alındığında, önerilen algoritmanın son derece verimli, rekabet düzeyi yüksek ve uygulanabilir nitelikte bir yöntem olduğunu görülmektedir.

Çizelge 6.2. GD göstergesi için ortanca ve çeyrekler açıklığı değerleri (25000 fonksiyon değerlendirmesi)

Problem	IBMO		OMOPSO		AbYSS		NSGA2		SPEA2		
	$\tilde{x}$	*IQR*	$\tilde{x}$	*IQR*	$\tilde{x}$	*IQR*	$\tilde{x}$	*IQR*	$\tilde{x}$	*IQR*	
Schaffer	2.267E-04	2.206E-05	2.280E-04	2.026E-05	2.307E-04	1.629E-05	4.660E-02	1.547E-01	4.791E-02	1.837E-01	+
Fonseca	1.353E-04	9.399E-06	1.239E-04	1.136E-05	2.327E-04	3.429E-05	3.530E-04	4.801E-05	2.162E-04	2.503E-05	+
Kursawe	1.317E-04	1.206E-05	5.377E-04	1.042E-04	1.538E-04	2.243E-05	2.131E-04	3.175E-05	1.575E-04	2.161E-05	+
ZDT1	9.742E-05	4.633E-05	1.427E-04	3.821E-05	1.831E-04	3.199E-05	2.192E-04	4.767E-05	2.206E-04	2.510E-05	+
ZDT2	5.775E-05	3.085E-05	7.799E-05	2.393E-05	1.103E-04	4.385E-05	1.761E-04	4.195E-05	1.814E-04	4.768E-05	+
ZDT3	1.453E-04	3.103E-05	2.245E-04	4.070E-05	1.902E-04	2.009E-05	2.136E-04	1.530E-05	2.305E-04	2.131E-05	-
ZDT4	2.558E-04	2.555E-04	7.393E-01	4.664E-01	5.737E-04	5.181E-04	4.286E-04	2.853E-04	5.209E-04	4.420E-04	+
ZDT6	1.496E-04	1.328E-05	2.650E-02	4.634E-02	1.804E-04	1.874E-05	8.903E-04	1.277E-04	1.663E-03	2.625E-04	+
Constr	1.886E-04	4.229E-05	1.762E-04	2.143E-05	2.235E-04	3.261E-05	2.955E-04	4.094E-05	2.035E-04	2.546E-05	+
Srinivas	4.783E-05	1.620E-05	5.111E-05	1.939E-05	6.257E-05	2.055E-05	1.988E-04	3.558E-05	1.158E-04	2.634E-05	+
Osycka2	5.612E-03	2.109E-03	1.895E-03	7.865E-04	1.151E-03	1.592E-02	1.049E-03	8.903E-05	1.427E-03	2.447E-04	+
Golinski	3.179E-04	3.550E-05	3.697E-04	5.605E-05	3.195E-04	3.110E-05	3.231E-04	2.673E-05	2.347E-04	2.440E-05	-
Tanaka	5.464E-04	2.059E-04	8.024E-04	1.161E-04	7.711E-04	9.256E-05	7.494E-04	1.134E-04	6.464E-04	1.128E-04	+
Binh2	6.761E-04	2.797E-05	1.245E-03	4.450E-05	1.249E-03	4.456E-05	1.387E-03	9.750E-05	1.582E-03	4.950E-05	+
DTLZ1	1.515E-02	5.174E-02	1.025E+01	1.471E+00	1.974E-03	3.543E-02	1.054E-02	5.824E-02	5.794E+00	2.410E+00	+
DTLZ2	3.006E-04	5.841E-05	8.746E-04	5.099E-05	2.001E-04	2.757E-05	1.235E-03	2.862E-04	5.608E-04	1.222E-04	+
DTLZ3	3.224E-01	1.891E-01	3.455E+01	5.756E+00	6.187E-01	7.386E-01	8.410E-01	7.291E-01	2.053E+01	3.672E+00	+
DTLZ4	7.806E-04	1.514E-04	1.398E-03	2.024E-04	7.058E-04	1.978E-04	2.048E-03	1.518E-04	9.453E-04	1.140E-04	+
DTLZ5	9.934E-02	4.844E-03	2.596E-04	5.758E-06	2.520E-04	1.704E-06	6.300E-04	5.525E-05	2.721E-04	8.066E-06	+
DTLZ6	4.570E-01	1.184E-02	2.524E-04	1.733E-06	1.014E-01	2.762E-02	1.685E-01	3.117E-02	2.977E-01	1.427E-02	+
DTLZ7	4.324E-04	2.051E-04	1.421E-03	4.009E-04	5.284E-04	2.114E-04	2.815E-03	5.203E-04	3.129E-03	7.753E-04	+
Viennet	4.672E-04	6.208E-05	7.029E-04	1.080E-04	5.287E-04	8.470E-05	3.411E-03	8.552E-04	8.982E-04	7.119E-05	+
Viennet2	1.269E-04	3.995E-05	1.422E-04	3.666E-05	1.860E-04	7.390E-05	7.250E-04	3.769E-04	1.444E-04	1.876E-05	+
Viennet3	4.377E-05	4.190E-05	3.309E-05	8.978E-06	3.509E-05	5.431E-06	2.544E-04	7.682E-05	4.270E-05	6.508E-06	+
Viennet4	1.378E-04	3.383E-05	1.001E-04	9.411E-06	9.663E-05	1.359E-05	5.280E-04	2.104E-04	1.155E-04	1.558E-05	+
Tamaki	1.452E-03	2.474E-04	7.414E-02	1.599E-04	7.236E-02	4.630E-04	1.643E-01	1.319E-03	7.363E-02	2.513E-04	+

Çizelge 6.3. SP göstergesi için ortanca ve çeyrekler açıklığı değerleri (25000 fonksiyon değerlendirmesi)

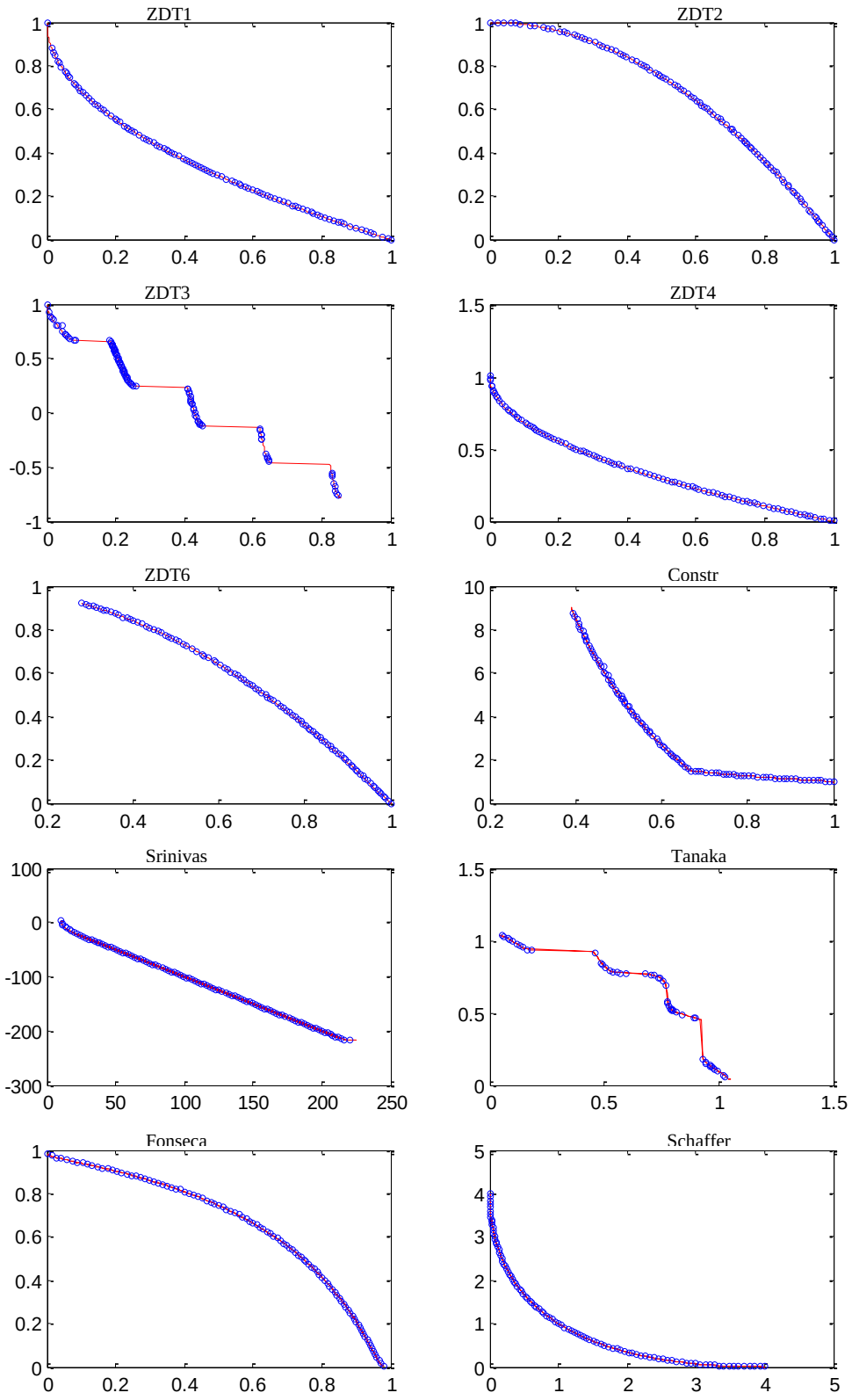
Problems	IBMO		OMOPSO		AbYSS		NSGA2		SPEA2		
	$\tilde{x}$	*IQR*	$\tilde{x}$	*IQR*	$\tilde{x}$	*IQR*	$\tilde{x}$	*IQR*	$\tilde{x}$	*IQR*	
Schaffer	1.136E-01	2.178E-02	1.114E-01	2.153E-02	1.284E-01	2.215E-02	6.450E-01	4.484E-01	4.921E-01	5.830E-01	+
Fonseca	7.055E-02	1.266E-02	5.981E-02	1.796E-02	1.032E-01	1.979E-02	3.650E-01	5.502E-02	1.236E-01	2.382E-02	+
Kursawe	2.290E-01	7.056E-03	2.987E-01	3.078E-02	2.410E-01	1.003E-02	5.424E-01	8.873E-02	2.947E-01	2.124E-02	+
ZDT1	3.102E-01	6.050E-02	8.589E-02	1.853E-02	1.128E-01	1.869E-02	3.907E-01	7.425E-02	1.322E-01	1.805E-02	+
ZDT2	2.507E-01	4.962E-02	8.697E-02	1.715E-02	1.153E-01	2.047E-02	3.920E-01	5.727E-02	1.410E-01	1.813E-02	+
ZDT3	8.122E-01	2.047E-01	1.562E-01	3.502E-02	1.406E-01	2.107E-01	3.822E-01	8.212E-02	1.668E-01	2.281E-02	+
ZDT4	6.230E-01	1.445E-01	8.837E-01	9.390E-02	1.333E-01	2.880E-02	3.985E-01	7.499E-02	2.081E-01	1.040E-01	+
ZDT6	1.084E-01	2.679E-02	9.459E-01	1.250E+00	9.501E-02	1.477E-02	3.562E-01	6.074E-02	2.045E-01	2.823E-02	+
Constr	2.787E-01	1.072E-01	1.231E-01	2.303E-02	1.601E-01	2.169E-02	3.909E-01	5.702E-02	5.233E-01	3.461E-02	+
Srinivas	8.431E-02	1.548E-02	7.759E-02	1.381E-02	7.818E-02	1.841E-02	3.551E-01	5.496E-02	1.505E-01	1.836E-02	+
Osycka2	3.315E-01	1.936E-01	7.450E-01	2.215E-01	4.279E-01	1.189E-01	6.797E-01	1.508E-01	6.547E-01	1.272E-01	+
Golinski	1.390E-01	3.754E-02	2.969E-01	5.860E-02	1.406E-01	3.241E-02	5.282E-01	8.745E-02	6.958E-01	3.771E-02	+
Tanaka	7.181E-01	1.328E-01	3.550E-01	7.258E-02	3.049E-01	4.832E-02	3.677E-01	6.166E-02	3.151E-01	5.098E-02	+
Binh2	3.917E-01	1.548E-02	3.079E-01	2.005E-02	3.207E-01	1.566E-02	5.412E-01	4.966E-02	5.559E-01	2.354E-02	+
DTLZ1	3.716E-01	3.265E-01	5.746E-01	9.561E-02	6.711E-01	3.709E-01	5.310E-01	2.029E-01	1.442E+00	1.435E-01	+
DTLZ2	4.347E-01	2.665E-02	3.681E-01	2.020E-02	5.574E-01	3.451E-02	5.132E-01	5.589E-02	9.251E-02	1.004E-02	+
DTLZ3	8.279E-01	8.487E-02	8.075E-01	1.539E-01	9.466E-01	7.809E-02	8.738E-01	1.260E-01	1.049E+00	1.140E-01	+
DTLZ4	6.322E-01	4.097E-01	9.949E-01	1.751E-01	5.913E-01	1.299E-01	5.023E-01	6.311E-02	9.455E-02	1.643E-02	+
DTLZ5	7.285E-01	3.598E-02	2.370E-01	2.438E-02	1.675E-01	1.360E-02	4.432E-01	7.211E-02	2.289E-01	2.241E-02	+
DTLZ6	7.370E-01	5.292E-02	1.553E-01	2.450E-02	8.030E-01	5.052E-02	6.503E-01	5.915E-02	6.746E-01	3.573E-02	+
DTLZ7	6.991E-01	8.143E-02	3.926E-01	2.389E-02	5.956E-01	8.279E-02	5.019E-01	7.312E-02	3.332E-01	3.862E-02	+
Viennet	4.583E-01	2.513E-02	4.265E-01	1.924E-02	5.762E-01	3.939E-02	4.557E-01	6.339E-02	1.467E-01	7.221E-03	+
Viennet2	5.055E-01	3.971E-02	4.723E-01	2.306E-02	4.462E-01	3.232E-02	4.531E-01	6.871E-02	2.151E-01	1.523E-02	+
Viennet3	3.941E-01	7.292E-02	3.912E-01	2.567E-02	3.148E-01	2.019E-02	4.049E-01	5.117E-02	5.491E-01	1.962E-02	+
Viennet4	5.634E-01	3.831E-02	5.080E-01	2.563E-02	6.808E-01	3.203E-02	5.481E-01	8.628E-02	3.298E-01	1.535E-02	+
Tamaki	5.466E-01	8.235E-02	4.933E-01	1.623E-02	6.885E-01	3.176E-02	6.898E-01	3.917E-02	3.004E-01	1.411E-02	+

Çizelge 6.4. MS göstergesi için ortanca ve çeyrekler açıklığı değerleri (25000 fonksiyon değerlendirmesi)

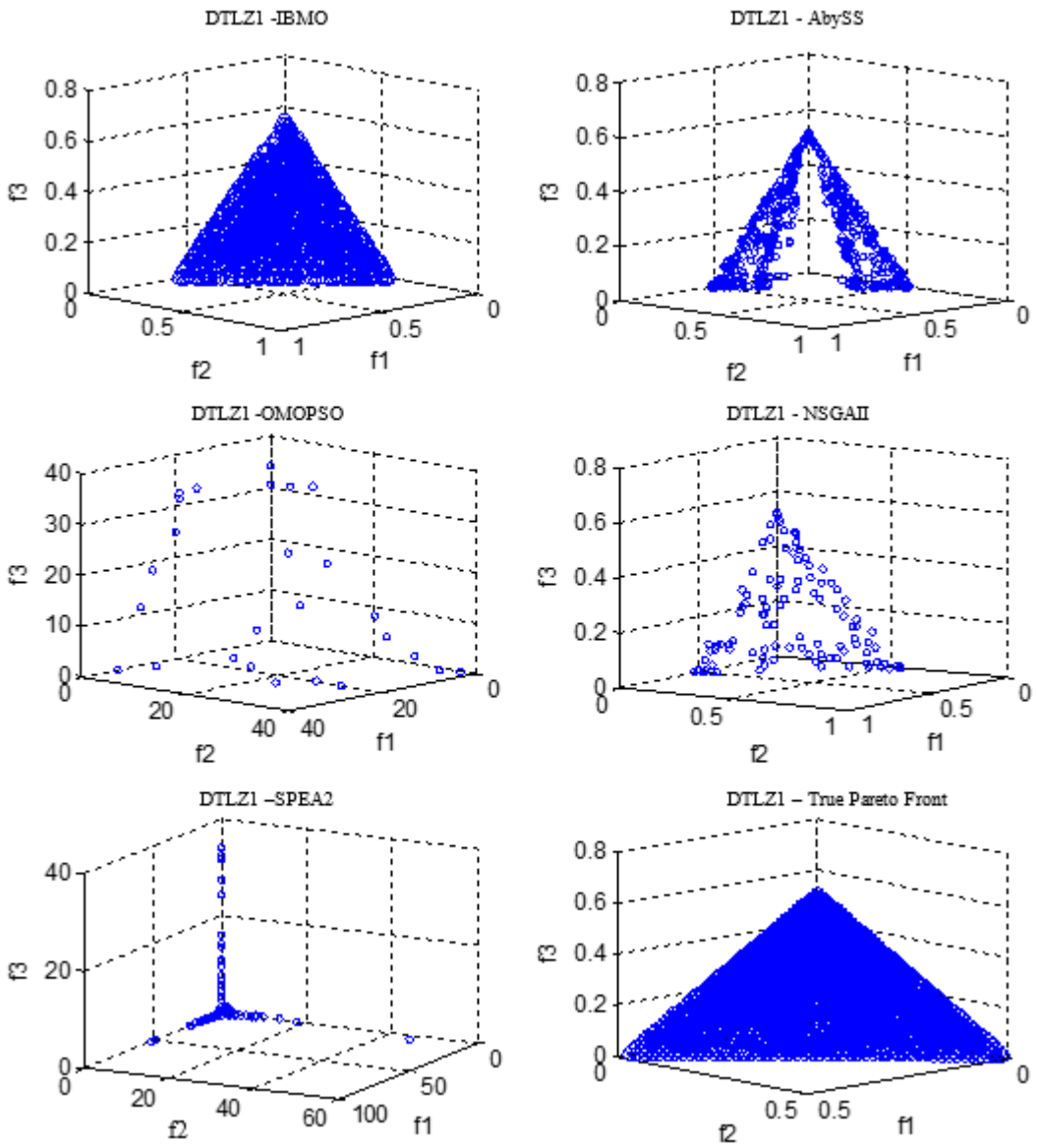
Problem	IBMO		OMOPSO		AbYSS		NSGA2		SPEA2		
	$\tilde{x}$	*IQR*	$\tilde{x}$	*IQR*	$\tilde{x}$	*IQR*	$\tilde{x}$	*IQR*	$\tilde{x}$	*IQR*	
Schaffer	1.000E+00	1.641E-04	1.000E+00	7.366E-05	1.000E+00	4.538E-05	8.711E-01	2.524E-01	8.550E-01	2.221E-01	+
Fonseca	9.995E-01	5.704E-04	9.999E-01	9.848E-05	1.000E+00	0.000E+00	1.000E+00	1.306E-05	9.996E-01	4.810E-04	+
Kursawe	9.996E-01	4.685E-04	9.907E-01	6.361E-03	1.000E+00	3.424E-05	9.999E-01	1.327E-04	9.993E-01	1.218E-03	+
ZDT1	1.000E+00	0.000E+00	1.000E+00	0.000E+00	1.000E+00	8.511E-05	9.996E-01	1.900E-04	9.990E-01	5.165E-04	+
ZDT2	1.000E+00	0.000E+00	1.000E+00	0.000E+00	1.000E+00	6.945E-05	9.992E-01	3.186E-04	9.978E-01	9.446E-04	-
ZDT3	9.973E-01	8.535E-03	9.984E-01	1.686E-03	9.998E-01	2.053E-01	9.997E-01	2.102E-04	9.993E-01	3.937E-04	+
ZDT4	9.992E-01	1.045E-03	7.071E-01	0.000E+00	9.978E-01	2.769E-03	9.977E-01	2.147E-03	9.733E-01	9.728E-02	+
ZDT6	1.000E+00	0.000E+00	1.000E+00	0.000E+00	9.994E-01	8.501E-05	9.933E-01	1.533E-03	9.863E-01	2.845E-03	+
Constr	9.035E-01	9.220E-02	9.965E-01	3.446E-03	9.990E-01	1.505E-03	9.974E-01	3.286E-03	9.960E-01	4.733E-03	+
Srinivas	9.826E-01	1.501E-02	9.867E-01	7.348E-03	9.921E-01	4.859E-03	9.926E-01	5.533E-03	9.921E-01	5.342E-03	+
Osycka2	8.189E-01	2.010E-01	8.048E-01	2.954E-01	8.137E-01	6.668E-02	9.803E-01	1.872E-01	7.904E-01	2.699E-01	+
Golinski	9.624E-01	4.478E-02	9.914E-01	1.302E-02	9.889E-01	2.606E-02	9.783E-01	3.387E-02	9.515E-01	2.851E-02	+
Tanaka	9.632E-01	2.586E-02	9.914E-01	6.635E-03	9.981E-01	5.712E-04	9.979E-01	1.963E-03	9.958E-01	3.533E-03	+
Binh2	7.694E-01	2.788E-04	8.111E-01	0.000E+00	8.111E-01	3.771E-10	8.111E-01	2.893E-10	8.111E-01	2.091E-08	+
DTLZ1	1.000E+00	0.000E+00	1.000E+00	0.000E+00	1.000E+00	0.000E+00	1.000E+00	0.000E+00	1.000E+00	6.661E-16	+
DTLZ2	1.000E+00	0.000E+00	1.000E+00	0.000E+00	1.000E+00	2.044E-12	1.000E+00	3.933E-11	1.000E+00	5.571E-06	+
DTLZ3	1.000E+00	0.000E+00	1.000E+00	0.000E+00	1.000E+00	1.110E-16	1.000E+00	0.000E+00	1.000E+00	1.022E-10	-
DTLZ4	1.000E+00	0.000E+00	1.000E+00	0.000E+00	1.000E+00	0.000E+00	1.000E+00	0.000E+00	1.000E+00	0.000E+00	+
DTLZ5	6.075E-01	1.673E-02	1.000E+00	0.000E+00	1.000E+00	3.875E-14	1.000E+00	9.466E-10	1.000E+00	2.103E-05	+
DTLZ6	8.165E-01	0.000E+00	1.000E+00	8.869E-09	1.000E+00	0.000E+00	1.000E+00	3.614E-13	1.000E+00	2.674E-06	+
DTLZ7	9.887E-01	5.866E-02	9.899E-01	8.378E-03	7.086E-01	2.929E-01	9.986E-01	6.368E-04	9.733E-01	7.641E-03	+
Viennet	9.993E-01	2.168E-03	1.000E+00	5.893E-04	1.000E+00	5.464E-06	1.000E+00	1.100E-04	9.984E-01	3.491E-03	+
Viennet2	9.772E-01	4.864E-02	9.953E-01	9.259E-03	9.974E-01	5.015E-04	9.972E-01	2.093E-03	9.928E-01	1.142E-02	+
Viennet3	9.953E-01	6.446E-03	9.994E-01	3.199E-03	1.000E+00	3.893E-05	9.999E-01	2.596E-04	9.985E-01	3.259E-03	+
Viennet4	9.744E-01	2.198E-02	9.918E-01	7.614E-03	9.999E-01	6.201E-04	9.980E-01	6.219E-03	9.837E-01	1.966E-02	+
Tamaki	8.958E-01	3.791E-02	1.020E-02	0.000E+00	1.020E-02	1.388E-17	1.020E-02	3.253E-14	1.021E-02	3.985E-06	+

Çizelge 6.5. HV göstergesi için ortanca ve çeyrekler açıklığı değerleri (25000 fonksiyon değerlendirmesi)

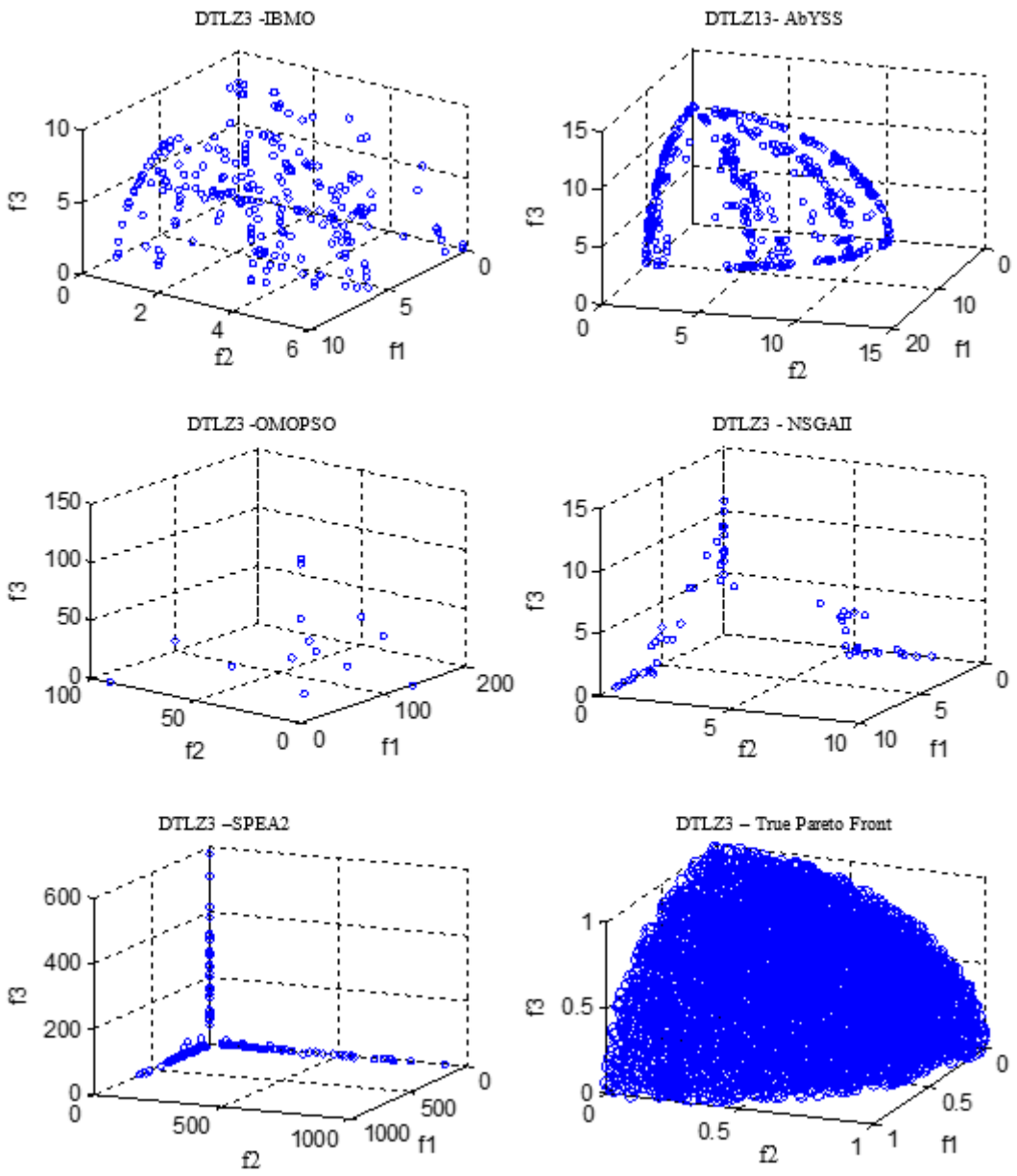
Problem	IBMO		OMOPSO		AbYSS		NSGA2		SPEA2		
	$\tilde{x}$	*IQR*	$\tilde{x}$	*IQR*	$\tilde{x}$	*IQR*	$\tilde{x}$	*IQR*	$\tilde{x}$	*IQR*	
Schaffer	8.299E-01	3.609E-05	8.299E-01	4.209E-05	8.299E-01	4.903E-05	5.732E-01	2.476E-01	5.850E-01	2.334E-01	+
Fonseca	3.119E-01	8.878E-05	3.124E-01	4.601E-05	3.104E-01	3.408E-04	3.080E-01	4.158E-04	3.107E-01	2.884E-04	+
Kursawe	4.015E-01	6.544E-05	3.972E-01	7.238E-04	4.011E-01	2.133E-04	3.996E-01	3.063E-04	4.008E-01	1.920E-04	+
ZDT1	6.607E-01	7.884E-04	6.611E-01	4.267E-04	6.614E-01	2.955E-04	6.593E-01	4.637E-04	6.600E-01	3.819E-04	+
ZDT2	3.282E-01	4.334E-04	3.280E-01	3.944E-04	3.281E-01	3.655E-04	3.261E-01	4.737E-04	3.263E-01	7.469E-04	+
ZDT3	5.130E-01	2.224E-03	5.141E-01	1.359E-03	5.158E-01	3.462E-03	5.149E-01	2.017E-04	5.142E-01	2.924E-04	+
ZDT4	6.583E-01	6.610E-03	0.000E+00	0.000E+00	6.546E-01	7.715E-03	6.552E-01	4.134E-03	6.529E-01	9.278E-03	+
ZDT6	4.014E-01	5.487E-05	4.012E-01	9.551E-05	4.003E-01	1.927E-04	3.883E-01	1.898E-03	3.786E-01	3.455E-03	+
Constr	7.732E-01	6.442E-03	7.761E-01	2.201E-04	7.761E-01	2.015E-04	7.743E-01	3.807E-04	7.756E-01	3.335E-04	+
Srinivas	5.408E-01	9.853E-05	5.408E-01	1.006E-04	5.407E-01	1.317E-04	5.380E-01	5.511E-04	5.400E-01	2.170E-04	+
Osycka2	6.976E-01	4.397E-02	7.054E-01	1.200E-02	7.437E-01	2.542E-01	7.505E-01	7.583E-03	7.358E-01	3.578E-02	+
Golinski	9.696E-01	8.209E-05	9.674E-01	5.688E-04	9.693E-01	1.814E-04	9.690E-01	1.683E-04	9.672E-01	7.323E-04	-
Tanaka	3.032E-01	2.098E-03	3.056E-01	9.505E-04	3.075E-01	5.612E-04	3.077E-01	5.319E-04	3.085E-01	6.258E-04	+
Binh2	7.990E-01	4.865E-05	8.007E-01	2.799E-05	8.006E-01	5.954E-05	7.994E-01	2.805E-04	7.999E-01	1.779E-04	+
DTLZ1	5.409E-01	5.284E-01	0.000E+00	0.000E+00	7.436E-01	7.070E-01	7.274E-01	4.474E-02	0.000E+00	0.000E+00	+
DTLZ2	4.335E-01	3.171E-03	4.165E-01	2.724E-03	4.333E-01	3.078E-03	3.751E-01	8.087E-03	4.390E-01	1.344E-03	+
DTLZ3	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	+
DTLZ4	4.377E-01	1.748E-02	3.850E-01	2.435E-02	4.357E-01	6.297E-03	3.768E-01	5.934E-03	4.354E-01	2.643E-03	+
DTLZ5	1.081E-02	1.436E-02	9.470E-02	1.252E-04	9.533E-02	1.685E-05	9.222E-02	2.360E-04	9.387E-02	1.987E-04	+
DTLZ6	0.000E+00	0.000E+00	9.537E-02	9.837E-06	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	+
DTLZ7	3.142E-01	2.147E-02	2.937E-01	1.262E-02	2.709E-01	4.654E-02	2.804E-01	5.503E-03	2.388E-01	1.595E-02	+
Viennet	7.371E-01	5.807E-04	7.376E-01	3.496E-04	7.364E-01	5.482E-04	7.115E-01	1.959E-03	7.402E-01	1.337E-04	+
Viennet2	9.293E-01	3.400E-04	9.295E-01	1.714E-04	9.287E-01	3.409E-04	9.200E-01	1.734E-03	9.300E-01	7.562E-05	+
Viennet3	8.402E-01	1.648E-04	8.398E-01	1.061E-04	8.398E-01	1.306E-04	8.326E-01	1.005E-03	8.374E-01	2.148E-04	+
Viennet4	8.675E-01	1.067E-03	8.698E-01	5.668E-04	8.683E-01	1.421E-03	8.554E-01	3.138E-03	8.718E-01	6.207E-04	+
Tamaki	3.213E-01	1.370E-02	4.184E-01	2.053E-03	4.139E-01	3.245E-03	3.572E-01	8.966E-03	4.318E-01	8.715E-04	+



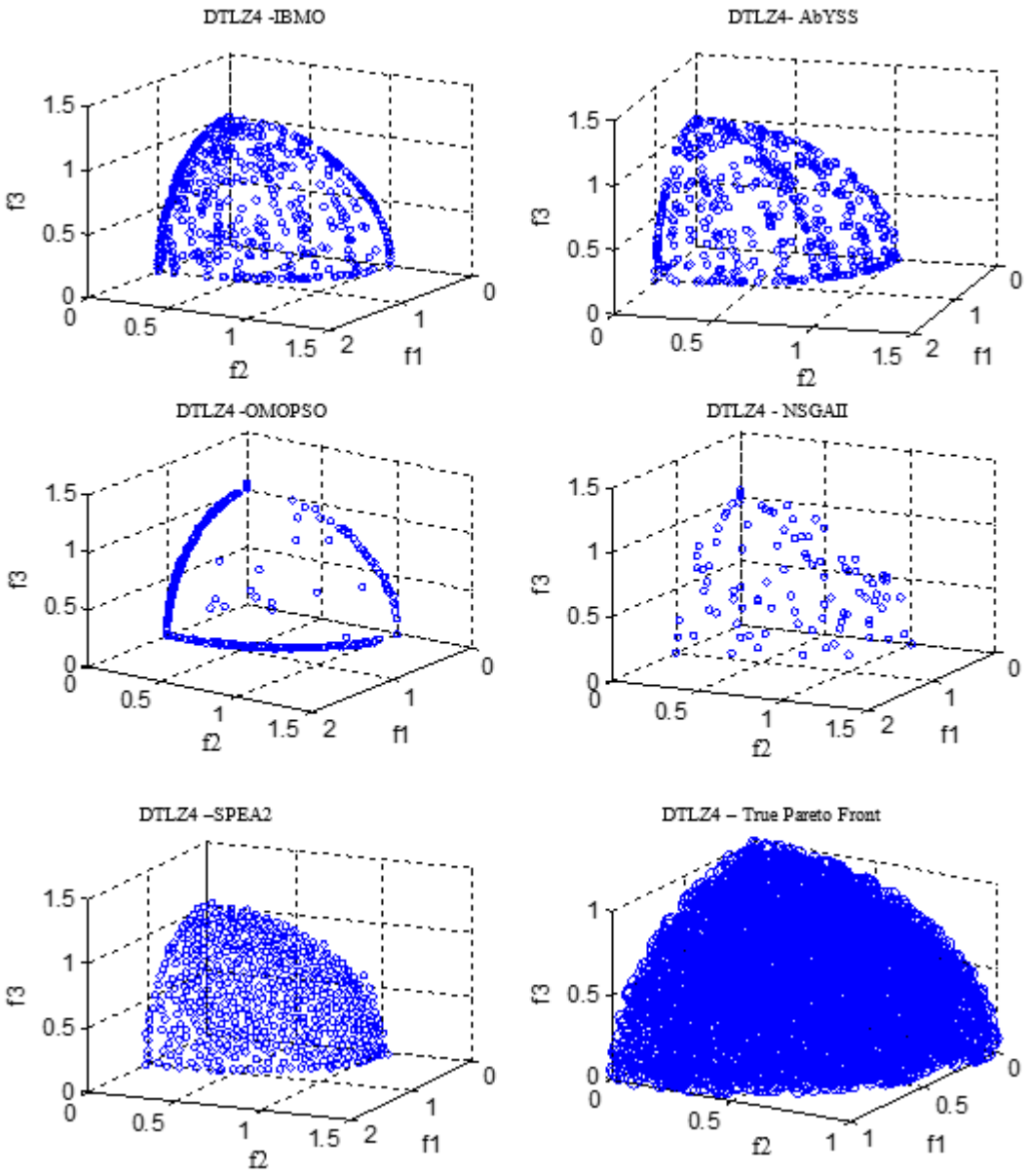
Şekil 6.7. 2-amaçlı test problemlerinin IBMO ile bulunan Pareto yüzeyleri



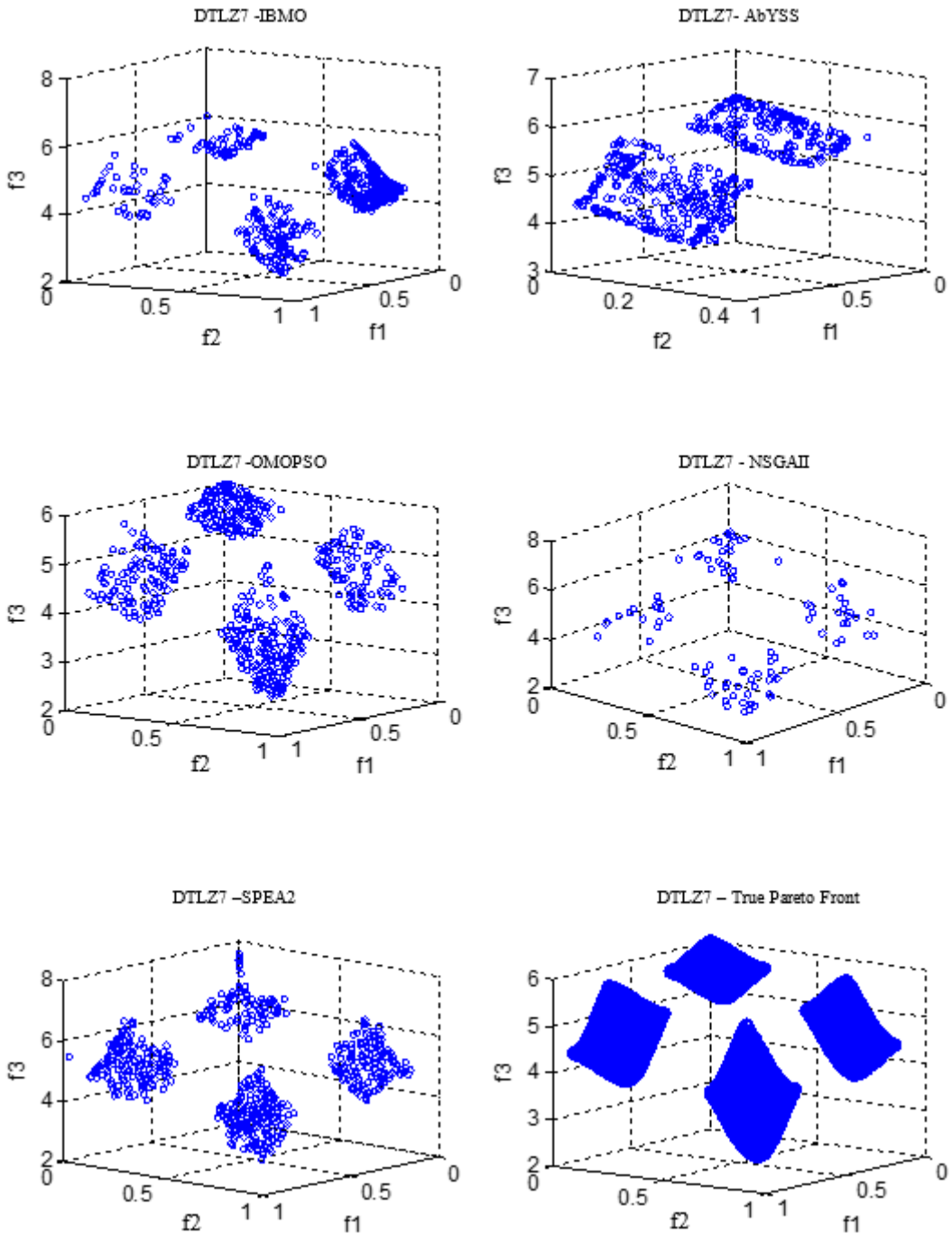
Şekil 6.8. DTLZ1 problemi için IBMO, AbySS, OMOPSO, NSGAII, SPEA2 ile elde edilmiş bastırılmamış çözümler ve doğru Pareto-Optimal yüzey



Şekil 6.9. DTLZ3 problemi için IBMO, AbySS, OMOPSO, NSGAII, SPEA2 ile elde edilmiş bastırılmamış çözümler ve doğru Pareto-Optimal yüzey



Şekil 6.10. DTLZ4 problemi için IBMO, AbySS, OMOPSO, NSGAII, SPEA2 ile elde edilmiş bastırılmamış çözümler ve doğru Pareto-Optimal yüzey



Şekil 6.11. DTLZ7 problemi için IBMO, AbySS, OMOPSO, NSGAII, SPEA2 ile elde edilmiş bastırılmamış çözümler ve doğru Pareto-Optimal yüzey

7. ÇOK AMAÇLI OPTİMİZASYON İLE RENKLİ GÖRÜNTÜ SEGMENTASYONU

Bu bölümde çok-amaçlı optimizasyona dayanan yeni bir renkli görüntü segmentasyon algoritması önerilmektedir. Burada optimizasyon algoritması olarak dördüncü bölümde anlatılan IBMO, görüntü segmentasyonu için modifiye edilmiştir. Segmentasyon işlemi görüntüdeki piksellerin belirli özelliklerine göre gruplanması olarak düşünüldüğünde bir kümeleme problemine dönüştürülebilir. Önerilen yaklaşım, IBMO algoritmasını ve çok çekirdekli bölge genişletme tekniğini kullanır. Piksel düzeyinde çıkarılan genel amaçlı öznitelikler kümelenerek segmentasyon işlemi gerçekleştirilir.

Bu bölümde IBMO algoritmasının renkli görüntü segmentasyonuna uyarlanması ve önerilen yöntem detaylı bir şekilde açıklandıktan sonra, kullanılan genel amaçlı öznitelikler anlatılacaktır. Ayrıca segmentasyon sonuçlarını değerlendirmek için kullanılan ölçütlerden bahsedilecek ve deneysel sonuçlar üzerinde durulacaktır.

7.1. Önerilen Renkli Segmentasyon Yöntemi

Önerilen segmentasyon yönteminin ana hatları Şekil 7.1'de gösterilmektedir. Bu yöntem 3 aşamadan oluşur: (1) renkli görüntüden özniteliklerin çıkarılması, (2) çekirdek noktalarının optimal pozisyonlarının ve optimal eşik değerlerinin IBMO yardımıyla belirlenmesi, (3) çok çekirdekli bölge genişletme tekniği (SRG).

Bir görüntüde bulunan toplam piksel sayısı NP ise $NP \times 7$ boyutlu bir öznitelik vektörü tanımlanır. Bu vektör renkli görüntüden çıkarılan piksel-düzeyindeki 7 adet özniteliğin her bir piksel için çıkarılmasıyla doldurulur. Kullanılan öznitelikler şu şekilde sıralanır: HSV renk uzayının her bir bileşeni için ayrı ayrı çıkarılmak üzere 3 adet yerel homojenlik değeri, Gabor filtresi ile elde edilen birleştirilmiş bir doku özelliği ve CIE Lab renk uzayının tüm renk kanalları öznitelik vektörünün birer bileşeni olarak kullanılır. Vektördeki tüm değerler [0-1] aralığında normalleştirilmiş değerlerdir.

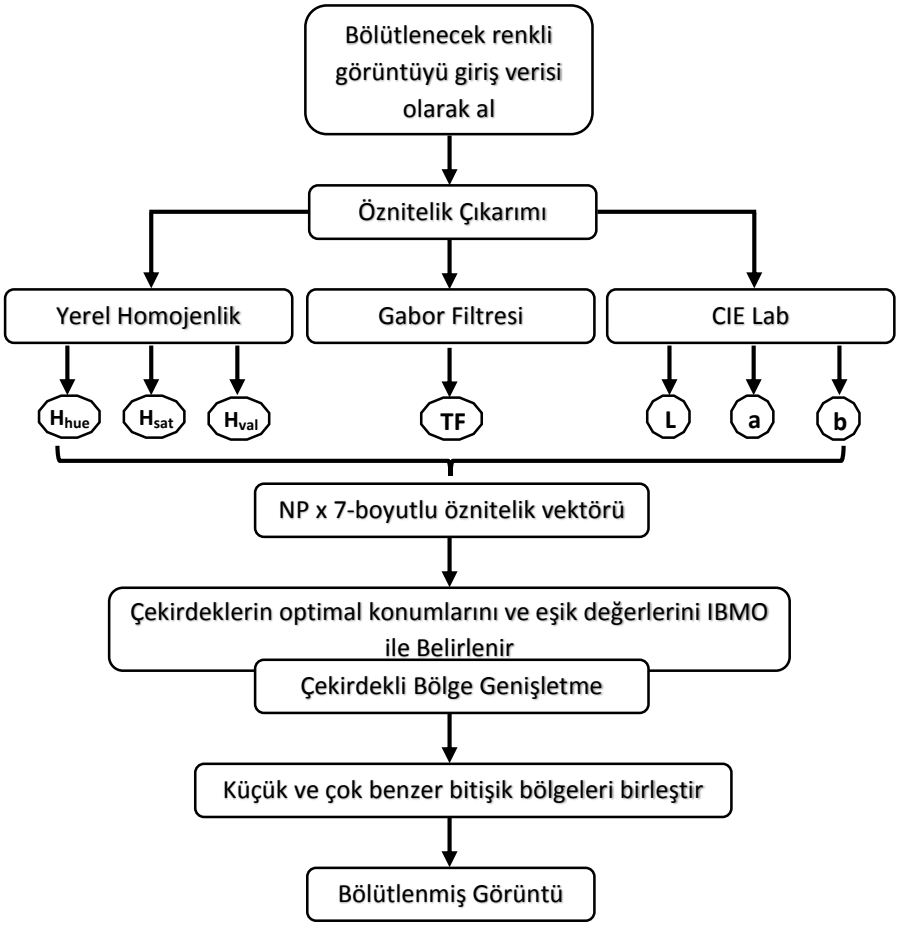
Segmentasyon işleminde modifiye edilmiş çok-çekirdekli bölge genişletme tekniği kullanılır. Bölge-genişletme teknikleri pikselleri homojen bölgeler içinde gruplandırır. Homojenlik kavramı bir benzerlik ölçütü ile temsil edilir. Pikseller arasındaki uzaklığı veya benzerliği ölçmek amacıyla denklem (7.1) kullanılır.

Görüntünün pikselleri öznitelik uzayında bir nokta olarak düşünüldüğünde, öznitelik değerleri üzerinden Öklid mesafesi ile hesaplanır.

$$dist(p_1, p_2) = \sqrt{\sum_{k=1}^{7}\big(F(p_1,k) - F(p_2,k)\big)^2} \tag{7.1}$$

Burada $dist(p_1, p_2)$ p_1 ve p_2 pikselleri arasındaki benzerliği (uzaklığı) gösterir. F öznitelik vektörü olmak üzere $F(n,k)$ $n.$ pikselin $k.$ özniteliğini gösterir.

SRG tekniği segmentasyon işlemine; bölge genişletme terminolojisinde çekirdek noktaları olarak bilinen merkez noktalarından başlar. Çekirdek pikselin komşuluğundaki piksellerden başlanarak piksellerin çekirdeğe olan benzerlikleri hesaplanır. Daha sonra benzer pikseller aynı sembolle etiketlenerek bölgeler oluşturulur. Çekirdek pikselin değeri bölgeye her yeni eklenmesiyle birlikte yeniden hesaplanır. Görüntüdeki tüm pikseller bir bölgeye ait oluncaya kadar bu etiketleme işlemine devam edilir. SRG yönteminin başarısı büyük ölçüde başlangıçta seçilen çekirdek noktalarının ve homojenlik ölçütünün seçimine bağlıdır.



NP: görüntüdeki toplam piksel sayısı; TF: Doku özniteliği
H_{hue} - H_{sat} - H_{val} : Hue, Saturation ve Value için yerel homojenlik değerleri

Şekil 7.1. Önerilen segmentasyon metodunun çalışmasını açıklayan ana hatları

Çalışmada kullanılan SRG yöntemi dört aşamada açıklanabilir. İlk olarak görüntüdeki tüm piksellerin tüm çekirdek noktalarına olan uzaklıkları hesaplanır ve bir matriste saklanır. İkinci adımda; önceki adımda hesaplanan matris kullanılarak tüm piksellerin tüm çekirdeklere ayrı ayrı ait olma olasılıkları hesaplanır. Olasılık hesaplaması için denklem (7.2)'de verilen FCM algoritmasının üyelik fonksiyonu kullanılır.

$$u_{ij} = \frac{1}{\sum_{k=1}^{C} \left(\frac{\|x_i - c_j\|}{\|x_i - c_k\|} \right)^2} \tag{7.2}$$

Burada $u_{ij} - x_i$ pikselinin j. bölgeye olan üyelik derecesini ifade eder. x_i görüntüdeki i. piksel anlamına gelir. c_j j. bölgenin çekirdek pikselidir. $||*||$ gösterimi ise ölçülen herhangi bir piksel ile çekirdek nokta arasındaki benzerliği (uzaklığı) ifade eder.

Sonraki aşamada; hesaplanan olasılık değerlerini bir eşik değeri ile filtreleyen bir konvolüsyon maskesi yardımıyla bir ikili matris oluşturulur. Eşik değeri olasılık değerinden büyükse, matris değeri sıfır olur. Aksi halde bir olarak kabul edilir. Böylece klasik SRG yönteminin aksine her adımda çekirdek güncellemesi yapılmadan bölgeler tek seferde oluşturulur. Son aşamada; çekirdek noktaları referans verilerek bağlantılı bileşen etiketleme yöntemi uygulanarak pikseller bölgelerine etiketlenir. Bu işlem tüm çekirdek noktaları için sırayla uygulanır. Bu prosedürün sonunda yine de etiketlenmeden kalan pikseller olabilir. Bir bölgeye dâhil edilmemiş olan bu pikseller son-işleme sürecinde ele alınarak kendilerine bitişik olan bölgelerden en benzer olanına eklenirler. Etiketlenmeyen piksel sayısının tüm piksel sayısına oranı 0.25'in üzerinde olması durumunda, bu oransal değer optimizasyon algoritmasında kısıt fonksiyon değeri olarak dikkate alınır. Bu oranın 0.25 değerinden küçük ya da eşit olması durumunda kısıt ihmal edilir ve sıfır olduğu kabul edilerek son-işleme sürecinde piksellerin geriye kalanlarının etiketlenmesi sağlanır.

Daha doğru bir segmentasyon sonucu elde etmek için, IBMO optimizasyon algoritması önerilen SRG-temelli segmentasyon yöntemine 3 önemli katkı sağlamaktadır. Bu katkılar aşağıdaki gibi özetlenebilir:

- Çekirdek noktalarının görüntüdeki optimal pozisyonlarını tespit eder.
- Her çekirdek noktası için kullanılacak ayrı bir eşik değerinin belirlemesini sağlar. Bu eşik değeri ilgili bölgenin çekirdek noktası ve etrafındaki pikseller için homojenlik ölçütü olarak kullanılır.

- Çok-amaçlı optimizasyon algoritması olarak segmentasyonun kalitesini birden fazla amacı hesaba katarak değerlendirebilme yeteneğine sahiptir. Böylece tek benzerlik ölçütü kullanmak ya da ölçütleri tek bir fonksiyonda birleştirmek yerine birbirinden bağımsız ve eş zamanlı olarak değerlendirilebilmesi sağlanmış olur.

Segmentasyon amacıyla çatısı yeniden şekillendirilen IBMO algoritmasında $(i = 1, 2, \dots SN)$ olmak üzere her bir x_i çözümü 2xD boyutlu bir vektördür. Burada D çekirdek sayısıdır ve çekirdeklerin görüntüde aynı bölge içine denk gelme ihtimaline karşı öngörülen çıkarılması beklenen bölge sayısından bir miktar daha fazla bir değere atanarak algoritma çalıştırılır. x_i çözümü Şekil (7.2)'de tasvir edildiği gibidir. IBMO çekirdek değerlerini piksel indeksleri olarak $\{1,2, \dots, NP\}$ aralığında üretir. Eşik değerleri ise bölgeye ait olma olasılığı olarak ele alındığı için [0, 1] aralığında oluşturulur. $NP = I_{width} \times I_{height}$ görüntüdeki toplam piksel sayısıdır.



Şekil 7.2. IBMO algoritmasında oluşturulan bir aday çözüm gösterimi

Segmentasyon görüntüdeki birbirine benzeyen bitişik piksellerin kümelenmesi olarak düşünülürse bir kümeleme problemi olarak çözüm üretilebilir. Kümelemenin kalitesini değerlendirmek için yeterli olduğu yaygın olarak kabul görmüş olan iki önemli ölçüt bulunmaktadır. Bu ölçütler eş zamanlı olarak hesaba katılarak kümeleme yapılırsa görüntü parçalarına düzgün bir şekilde ayrılabilir ve segmentasyon yapılmış olur. Bu ölçütler: sıkılık (*compactness*) ve ayrışma (*separation*)'dır. Kümelemenin amacı maksimum sıkılık yani bölge içi piksel benzerliğinin yüksek olması ve minimum ayrışma yani farklı bölgelerde bulunan piksellerin birbirlerine olan benzerliklerinin minimum olmasını sağlamaktır. Bu bağlamda; görüntü segmentlerinin doğru pikselleri ne ölçüde kapsadıklarını ölçmek için küme doğrulama göstergeleri kullanılabilir. Literatürde çok sayıda küme doğruluk göstergesi önerilmiştir. Bunlardan en fazla alıntı yapılan ve sıklıkla tercih edilenlerden bazıları şöyle sıralanabilir: Dunn, Davies-Bouldin, Silhouette, S_Dbw, CS, Sym, Turi (Maulik ve Bandyopadhyay, 2002; Halkidi, 2001). Bu göstergelerden bazılarının hesaplama karmaşıklığı yüksektir ve amaç fonksiyonu olarak kullanılması uygun değildir. Bu göstergelerin birçoğu sıkılık ve ayrışma değerlerini tek fonksiyon içinde birleştirerek kullanır. Oysa bu iki kavram çoğunlukla birbiriyle çelişen amaçlardır.

Bu noktada IBMO açıklanan segmentasyon yöntemine entegre edilir. Bölge içi benzerlik ve bölgeler arası farklılık iki ayrı amaç fonksiyon olarak ele alınır. Çalışmada S_Dbw doğruluk göstergesi üzerinde duruldu ve amaç fonksiyon olarak kullanıldı. S_Dbw doğruluk göstergesi denklem (7.3)'de verilmiştir.

$$S_Dbw = scat(NC) + Dens_bw(NC) \tag{7.3}$$

Bu doğruluk göstergesi denklem (7.3)'den anlaşıldığı gibi dağılma (scattering) ve yoğunluk (density) terimlerinin toplanması ile oluşur. Dağılma ifadesi kümelerin sıkılığını ölçer. Denklem (7.4)'teki gibi tanımlanır.

$$Objective1 = scat(NC) = \frac{1}{NC}\sum_{k=1}^{NC}\frac{\|\sigma(c_k)\|}{\|\sigma(S)\|} \tag{7.4}$$

Burada $\sigma(c_k)$ ifadesi c_k kümesinin varyansıdır ve $\sigma(S)$ ifadesi ise tüm S veri kümesinin varyansıdır. Yoğunluk ifadesi ise kümelerin ayrışıklığını ölçer ve denklem (7.5)'deki gibi tanımlanır.

$$Objective2 = Dens_{bw(K)} = \frac{1}{NC(NC-1)}\sum_{k=1}^{NC}\sum_{l=1\setminus l\neq k}^{NC}\frac{density(c_k,c_l)}{max\{density(c_k),density(c_l)\}} \tag{7.5}$$

$$density(c_k) = \sum_{s_i\in C_k} f(s_i,\bar{c_k})$$

$$density(c_k,c_l) = \sum_{s_i\in C_k\cup C_l} f\left(s_i,\frac{\bar{c_k}+\bar{c_l}}{2}\right)$$

$$f(s,c_k) = \begin{cases} 0 & if\ density(s_i,c_k) > stdev(C), \\ 1 & otherwise. \end{cases}$$

$$stdev(C) = 1/NC\sqrt{\sum_{c_k\in C}\|\sigma(c_k)\|}$$

S_Dbw doğruluk göstergesini bir toplam fonksiyonu olarak kullanmak yerine, IBMO içinde birbirinden bağımsız iki ayrı amaç fonksiyonu şeklinde ele alınır. Böylece dağılma ve yoğunluk değerleri eş zamanlı ve bağımsız olarak kısıtlı Pareto üstünlüğü kullanılarak değerlendirilirler. Yukarıda ifade edildiği gibi kısıt değeri olarak da çevrim içinde etiketlenmemiş piksellerin tüm piksel sayısına oranı kullanır. Bu işlem olabildiğince fazla etiketlenmesi yönünde seçim basıncını artırır.

Son olarak; IBMO çevrimlerini tamamlayıp optimal çekirdek pozisyonlarını ve bu çekirdeklere karşılık gelen eşik değerlerini hesapladıktan sonra bir son-işleme prosedürü uygulanır. Öncelikle etiketlenmiş görüntüye bir morfolojik kapatma işlemi uygulanarak etiketlenmemiş az sayıda pikselden oluşan küçük bölgelerinin kapanması sağlanır. Daha sonra mevcut etiketlenmiş bölgelerden küme içi varyans değeri en büyük olan kümenin varyans değeri, etiketlenmemiş bölgelerde benzerlik eşiği olarak saklanır. Görüntü etiketlenmiş ve etiketlenmemiş pikseller olarak maskelenerek ikili görüntü oluşturulur. Bu ikili görüntü üzerinde bağlantılı bileşen analizi yapılır ve etiketlenmeyen ve ayrık durumda olan bölgeler tespit edilir. Piksel sayısı önceden tanımlanmış ve görüntüye göre değişkenlik gösteren bir en-küçük-bölge-alanı değişken değerinden küçük olan bölgeler tek bir bölge olarak kabul edilir. Bu küçük bölgeler komşuluğundaki bölgelerden kendisine en benzer olan bölgeler ile birleştirilir. Geriye kalan etiketsiz bölgelere sırasıyla bölge genişletme işlemi uygulanır. Bu işlem sırasında benzerlik eşiği olarak hesaplanmış olan en büyük varyans değeri kullanılır. Bu işlem tüm pikseller etiketleninceye kadar devam eder.

7.2. Öznitelik Çıkarımı

Çoğunlukla kenar tespiti ve doku analizinden oluşan öznitelik çıkarma işlemi görüntü segmentasyonunun en önemli bileşenidir. Çalışmada segmentasyon benzer piksellerin kümelenmesi problemi şeklinde ele alındığı için kümelemede kullanılmak üzere bir veri matrisi diğer bir ifadeyle öznitelik vektörü oluşturuldu. Bu vektör her bir piksel için renk, kenar ve doku bilgilerini içeren yedi bileşenden oluşur. Öncelikle iki renk arasındaki göreceli algısal farklılıkları tanımadaki avantajından dolayı CIE L*a*b* renk uzayın tüm bileşenleri öznitelik vektörüne eklendi. Daha sonra her bir piksel için Gabor filtresi uygulanarak bir doku özelliği çıkarıldı. Son olarak yerel homojenlik değeri olarak isimlendirilen ve kenar değişimlerinde faydalanan bir öznitelik hesaplandı. Diğer bir algısal renk uzayı olan HSV'nin her bir kanalı için yerel homojenlik değerleri hesaplanarak vektöre eklendi.

Bu bölümde çalışmada ele alınan bu özniteliklerin detaylarına yer verilmektedir. Sırasıyla kullanılan renk uzayları, Gabor filtresi ve yerel homojenlik alt başlıklar halinde anlatılmaktadır.

7.2.1. Kullanılan Renk Uzayları

Renk bilgisi, görüntüyü tanımlamak için kullanılan en baskın ve ayırt edici düşük seviye görsel özelliklerin başında gelir. Renk uzayları renkler arasındaki farklılığı ölçmek için bazı ölçütler tanımlar. Bu ölçütler görüntü analizinde ve segmentasyon sürecinde önemli rol oynar. Renk uzayları karakteristiklerine göre üç kategoride toplanabilir: birincil renk uzayları, luminance-chrominance ve algısal renk uzayları (Busin ve ark., 2008).

Birincil renk uzaylarının başında gelen RGB renk modeli renk gösterimi açısından oldukça başarılıdır. Bununla birlikte luminance-chrominance ve algısal uzaylar iki renk arasındaki farklılığı ve/veya benzerliğin açıklanmasında daha iyi sonuçlar vermektedirler (Littmann ve Ritter, 1997).

Çalışmada HSV algısal renk uzayı yerel homojenlik değerlerini hesaplamak için kullanılır. HSV renkli görüntü analizinde ve renkli görüntü segmentasyonunda sıklıkla tercih edilen bir renk uzayıdır. Bunu yanı sıra Gabor filtresi için CIE L*a*b* renk uzayı kullanıldı. L* rengin açıklığını gösterirken, a* ve b* rengi oluşturan bileşenlerdir. CIE Lab renk uzayının en belirgin özelliği renk uzayının algılama yönünden düzgün değişim göstermesidir. Bu bakımdan renkler arsındaki farklılığı hesaplamak için Öklid uzaklığı kullanılabilir.

7.2.2. Gabor Filtresi

Gabor filtresinden yararlanan birçok metot çeşitli makine görüşü uygulamalarında başarılı bir şekilde uygulanmıştır. Bu uygulamalardan bazıları: doku segmentasyonu (Jain ve Farrokhnia, 1991; Teuner ve ark., 1995; Dunn ve Higgins, 1995), kenar tanıma (Namuduri, 1994), nesne tanıma (Casasent ve ark. 1992), el yazısı rakam tanıma (Hamamoto ve ark., 1998) olarak sıralanabilir. Gabor filtresi, içerdiği parametreler sayesinde memelilerin göz yapısının birincil görsel korteksinde yer alan hücre tepkilerini modelleyebilmektedir. Kullanılan boyutlara göre sürekli renk değişimlerini yakalayan bir çekirdek üretir. 2-boyutlu Gabor dalgacık gösterimi için ilk çalışmalar yine yönteme adını veren Daugman Gabor tarafından 1985 yılında yapılmıştır.

İki-boyutlu bir Gabor fonksiyonu sinüzoidal bir işaretin Gauss özellikli bir zarf ile modüle edilmesi temeline dayanan bir grup dalgacık olarak tanımlanabilir. Her bir

dalgacık özel bir yön ve frekanstaki enerjiyi göstermektedir. 2-boyutlu bir Gabor filtresi denklem (7.6)'daki gibi tanımlanır.

$$G(x, y, \theta, \lambda, \sigma_x, \sigma_y) = g(x, y, \theta, \sigma_x, \sigma_y).\, e^{i2\pi\frac{x'}{\lambda}} \tag{7.6}$$

$$g(x, y, \theta, \sigma_x, \sigma_y) = \frac{1}{2\pi\sigma_x\sigma_y} e^{-\frac{1}{2}\left[\left(\frac{x'}{\sigma_x}\right)^2 + \left(\frac{y'}{\sigma_y}\right)^2\right]}$$

$$x' = x\cos\theta + y\sin\theta$$

$$y' = -x\sin\theta + y\cos\theta$$

$$b = \frac{\log_2 \frac{\sigma}{\lambda}\pi + \sqrt{\frac{\ln 2}{2}}}{\log_2 \frac{\sigma}{\lambda}\pi - \sqrt{\frac{\ln 2}{2}}}$$

Denklem (7.6)'da verilen $g(x, y, \theta, \sigma_x, \sigma_y)$ ifadesi 2-boyutlu bir Gauss fonksiyonudur. θ parametresi 0 ile 360 derece arasında değer alabilen, filtrenin uygulanma açısıdır. λ dalga boyudur ve f = 1/λ ise kosinüs faktörünün uzamsal frekansıdır. σ_x ve σ_y x ve y doğrultusundaki varyansları gösterir ve ($\sigma_x = \sigma_y = \sigma$) genellikle eşit değer alacak şekilde kullanılırlar. σ/λ oranı ile uzamsal frekans belirlenebilir. b parametresi bant genişliğini gösterir.

$w \times h$ boyutlu bir $I(x, y)$ görüntüsü Gabor filtresi ile iki boyutlu konvolüsyon işlemine sokularak filtrelenmiş görüntü elde edilir. Bu işlem denklem (7.7) ile gösterilmiştir.

$$GI(x, y, \theta, \lambda, \sigma_x, \sigma_y) = I(x, y) * G(x, y, \theta, \lambda, \sigma_x, \sigma_y) \tag{7.7}$$

Çalışmada uygulanan Gabor filtresi için bant genişliği bir olarak kabul edilmiştir. Buna bağlı olarak standart sapma σ değeri, bant genişliği ve frekans değerleri kullanılarak hesaplanmıştır. Uygulanma açısı θ (Clausi ve Jernigan, 2000)'de tavsiye edilen değerler baz alınarak kullanılmıştır. Buna göre θ açısı $\pi/6$ ve katları şeklinde $\left(\theta = \left\{0, \frac{\pi}{6}, \frac{\pi}{3}, \frac{\pi}{2}, \frac{2\pi}{3}, \frac{5\pi}{6}\right\}\right)$ alınarak uygulanmıştır. Frekans değerleri (Clausi ve Jernigan, 2000)'de tavsiye edildiği gibi denklem (7.8) ve denklem (7.9) kullanılarak hesaplanmıştır.

$$F_H = 0.25 + 2^{(i-0.5)}/w \qquad 0.25 \le F_H < 0.5 \tag{7.8}$$

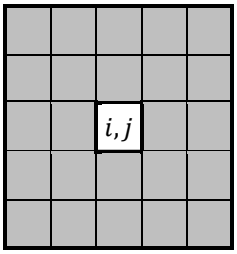
$$F_H = 0.25 - 2^{(i-0.5)}/w \qquad 0 < F_L < 0.25 \tag{7.9}$$

Burada $i = 1,2, \dots, \log_2 \frac{w}{8}$ değerinde kullanılır ve w parametresi görüntünün genişliğini gösterir. Çalışmada doku özelliğini çıkarmak için Gabor filtresi Lab renk uzayındaki görüntünün L kanalına uygulanmıştır. Çünkü a ve b bileşenleri kromatitiklik bileşenleridir ve doku verisi genellikle renge bağlı olarak oluşur. Bu bakımdan L bileşenini kullanmak yeterlidir. Filtre uygulandıktan sonra elde edilen cevap matrisinin diğer bir ifadeyle Gabor alt-bant katsayılarının yerel ortanca enerjileri alınarak her piksel için tek doku özelliğine indirgenmesi sağlanmıştır.

7.2.3. Yerel Homojenlik

Homojenlik büyük ölçüde bir görüntüden elde edilen yerel bilgiyle ilişkilidir ve görüntüde bir bölgenin ne ölçüde düzgün bir dağılım gösterdiğini, bölgedeki piksellerin birbirlerine ne kadar benzerlik gösterdiklerini ölçmek için kullanılır. Yerel homojenlik özniteliği, piksel-düzeyinde iki bileşenden oluşan renk bilgisi olarak tanımlanır. Bu bileşenler piksel renk değerlerinin standart sapması ve süreksizliktir (Chengand ve Sun, 2000).

Burada küçük renk değişimlerine karşı hassas ve algısal bir renk uzayı olan HSV kullanılarak renk bilgisi için piksel düzeyinde öznitelik çıkarılmıştır. MxN boyutlu bir görüntüde (i,j) pozisyonundaki bir pikselin üç renk bileşenleri $P_{ij}^{k} = \{P_{ij}^{H}, P_{ij}^{S}, P_{ij}^{V}\}$ notasyonu ile gösterilsin. $P_{ij}^{k}(k = H,S,V)$ pikselinin homojenlik değerinin hesaplanması için $d \geq 3$ ve tek bir tamsayı olmak üzere (i,j) merkezli ve $d \times d$ boyutunda bir pencere tanımlanır. 5×5 boyutunda örnek bir pencere Şekil 7.3'de gösterilmektedir.



Şekil 7.3. (i,j) pozisyonundaki bir piksel için yerel görüntü penceresi

Bir pikselin renk bileşenlerinin standart sapmaları, bu pencere içinde kalan komşu pikseller hesaba katılarak denklem (7.10)'a göre hesaplanır.

$$v_{ij}^{k} = \frac{v_{ij}^{k}}{v_{max}} \quad ve \quad v_{ij}^{k} = \sqrt{\frac{1}{d^2} \sum_{p=i-((d-1)/2)}^{i+((d-1)/2)} \sum_{q=j-((d-1)/2)}^{j+((d-1)/2)} \left(P_{ij}^{k} - \mu_{ij}^{k}\right)^2} \tag{7.10}$$

$$v_{max} = max\{v_{ij}\}, \; 0 \le i,p \le M-1, \quad 0 \le j,n \le N-1$$

Burada μ_{ij}^{k} pencere içindeki piksellerin k. renk bileşen değerlerinin aritmetik ortalamasıdır. Denklem (7.11)'de verildiği gibi hesaplanır.

$$\mu_{ij}^{k} = \frac{1}{d^2} \sum_{p=i-((d-1)/2)}^{i+((d-1)/2)} \sum_{q=j-((d-1)/2)}^{j+((d-1)/2)} P_{ij}^{k} \tag{7.11}$$

Bir P_{ij}^{k} renk bileşenin süreksizliği gradiyent büyüklükleri ile açıklanabilir. Gardiyent bilgisi için Sobel, Canny ve Robert gibi birçok farklı kenar çıkarma filtresi kullanılabilir. Bu amaçla çalışmada sadeliği nedeniyle 5×5 lik bir Sobel filtresi kullanılmıştır. (i,j) konumundaki bir piksel için gradiyent büyüklüğü e_{ij} denklem (7.12)'de verildiği gibi hesaplanır.

$$E_{ij} = \frac{e_{ij}}{e_{max}} \quad ve \quad e_{ij} = \sqrt{{G_x}^2 + {G_y}^2} \tag{7.12}$$

Burada $e_{max} = max\{e_{ij}\}$, G_x ve G_y ise renk bileşeninin x ve ye doğrultusundaki gradiyent büyüklükleridir. P_{ij}^{k} renk bileşeninin yerel homojenlik değeri denklem (7.13)'teki gibi hesaplanır.

$$H_{ij}^{k} = 1 - E_{ij}^{k} \times V_{ij}^{k} \tag{7.13}$$

Yerel homojenlik her noktada [0,1] aralığında bir değer alır. Bir pikseli çevreleyen bölge ne kadar homojen bir dağılım gösterirse, yerel homojenlik değeri o ölçüde bire yaklaşır.

7.3. Değerlendirme Ölçütleri ve Deneysel Sonuçlar

Segmentasyon sonuçlarının niceliksel olarak değerlendirilmesi büyük ölçüde öznel olması bakımından zor bir işlemdir. Çünkü bir görüntünün kesin doğru olarak

tanımlanan tek bir segmentasyon sonucu olmayabilir. Dolayısıyla bir algoritmanın çıktılarını karşılaştırmak için bir referans görüntü kullanılması her zaman tutarlı sonuçlar vermeyebilir. Literatürde en çok kullanılan değerlendirme yöntemleri belirlenmiş test görüntüleri için manuel oluşturulmuş bir karşılaştırma kümesinin kullanılmasına dayanır. Bu tez çalışmasında, segmentasyon sonuçlarının etkinliğini ölçmek amacıyla üç çeşit tutarlılık hatası üzerinde durulmuştur. Bu kapsamda öncelikle değerlendirme ölçütleri açık bir şekilde anlatılmaktadır. Daha sonra kullanılan kontrol parametre değerlerinin belirlenmesi için izlenen metodoloji açıklanmaktadır. Son olarak test görüntülerinden elde edilen deneysel sonuçlar ve değerlendirmelerine yer verilmektedir.

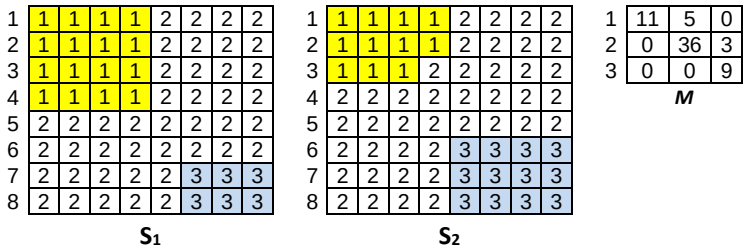
7.3.1. Değerlendirme Ölçütleri

Önerilen segmentasyon yönteminin sonuçları bölge-tabanlı tutarlılık ölçütleri kullanılarak değerlendirilmiştir. Bu ölçütler algoritmadan elde edilen sonuçların farklı el-çizimleri ile ne ölçüde tutarlılık gösterdiğini hesaplar. Martin ve ark. (2001) yerel düzeltme hatasına (LRE) dayanan iki hata ölçütü önerdiler. LRE; algoritma çıktısı ve insan algısıyla oluşturulmuş segmentasyon sonuçlarının örtüşme derecesini gösterir. Bu bakımdan doğru bir segmentasyon çıktısının LRE sonucunun sıfır ya da çok küçük bir değer olması beklenir. LRE matematiksel olarak denklem (7.14)'te verildiği gibi hesaplanır.

$$LRE(S_1, S_2, p_i) = \frac{|R(S_1, p_i) \setminus R(S_2, p_i)|}{|R(S_1, p_i)|} \tag{7.14}$$

Burada S_1 ve S_2 aynı görüntünün iki farklı segmentasyon sonucunu gösterir. $R(S, p_i)$ ifadesi S segmentasyonunda p_i pikselinin içinde bulunduğu bölgenin kapsadığı pikseller kümesini gösterir. \ sembolü küme fark operatörüdür. $|x|$ notasyonu ile x kümesinin eleman sayısı ifade edilir.

Görüntüdeki her bir pikselin LRE değerinin bulunması, hesaplama karmaşıklığı yüksek bir problemdir. Ancak n adet bölgeden oluşan S_1 ve m adet bölgeden oluşan S_2 segmentasyonları için oluşturulacak $n \times m$ boyutlu bir hata matrisi (M) yardımıyla bu durumun üstesinden gelinmiştir. Örnek bir hata matrisi hesaplanışı Şekil 7.4'te verilmiştir. Buna bağlı olarak LRE matrisinin hesaplanmasını gösteren sözde-kod Şekil 7.5'te verilmiştir.



Şekil 7.4. S1 ve S2 segmentasyonları için hata matrisi

```
for i = 1 to n
    region_i = ∑_j M_ij
    for j = 1 to m
        ind = find indices where S1equals i and S2equals j
        LRE(ind) = (region_i − M_ij)/region_i
    end for j
end for I
```

Şekil 7.5. Tüm pikseller için LRE hesaplamasını gösteren sözde-kod - $LRE(S_1, S_2)$

LRE'nin simetrik olmadığı dikkate alındığında, değerlendirme yapmak için tek başına LRE matrisinin toplanması yeterli değildir. Bu durumu her iki yönü de hesaba katarak düzeltmek için LRE'ye bağlı olarak hesaplanan iki yöntem bulunmaktadır. Bu yöntemler: yerel tutarlılık hatası (LCE) ve global tutarlılık hatası (GCE) dır. LCE görüntünün farklı yerlerinde farklı yönlerde düzeltme yapmak için uygundur. LCE matematiksel olarak denklem (7.15)'te verildiği gibi tanımlanır.

$$LCE(S_1, S_2) = \frac{1}{n}\sum_{i=1}^{n} \min\{LRE(S_1, S_2, p_i), LRE(S_2, S_1, p_i)\} \tag{7.15}$$

Global tutarlılık hatası; aynı yönde olan tüm yerel düzeltmelere odaklanır ve denklem (7.16) ile tanımlanır.

$$GCE(S_1, S_2) = \frac{1}{n}\min\left\{\sum_{i=1}^{n} LRE(S_1, S_2, p_i), \sum_{i=1}^{n} LRE(S_2, S_1, p_i)\right\} \tag{7.16}$$

Her iki ölçüt de $[0,1]$ aralığındadır ve düşük değerler daha iyi bir segmentasyon anlamına gelir. El-çizimi ile oluşturulan segmentasyon sonuçlarındaki farklılığın veya bozulmanın üstesinden gelmek için bir başka ölçüte daha ihtiyaç duyulmuştur. İki-yönlü tutarlılık hatası (BCE) olarak bilinen bu ölçüt bölgelerin örtüşme derecesiyle orantılı olarak segmentasyonlar arasındaki farklılığı cezalandırır. BCE denklem (7.17)'deki gibi tanımlanır.

$$BCE(S_{test}, \{S_k\}) = \frac{1}{n}\sum_{i=1}^{n} \min_k \{\max\{LRE(S_{test}, S_k, p_i), LRE(S_k, S_{test}, p_i)\}\} \quad (7.17)$$

Burada S_{test} bir algoritma ile elde edilen segmentasyonu gösterir. $\{S_k\}$ ise görüntünün el-çizimleriyle oluşturulmuş mevcut segmentasyon sonuçları kümesidir. Segmentasyonların karşılaştırılmasını kolaylaştırmak amacıyla anlatılan bölge-tabanlı bu üç tutarlılık hatası genellikle bir doğruluk göstergesi şeklinde kullanılabilir ve $LCI = 1 - LCE$, $GCI = 1 - GCE$, $BCI = 1 - BCE$ şeklinde tanımlanabilir. Bununla birlikte çalışmada, tutarlılık hatalarının orijinal formları kullanılmıştır. Ayrıca bu üç göstergenin toplamı ek bir sonuç olarak verilmiştir.

7.3.2. Parametre Seçimi

Önerilen segmentasyon algoritması ve karşılaştırma amacıyla ele alınan diğer tüm algoritmalar 240x160 piksel boyutlu test görüntüleri üzerinde 500 fonksiyon değerlendirmesi için çalıştırılarak sonuçlar alınmıştır. Koloni boyutu (*NP*), harici arşiv boyutu (*EXA*) ve en küçük bölge alanı gibi bazı kontrol parametreleri tüm algoritmalar için ortak kullanılmıştır. Ortak parametreler ve kullanılan değerleri Çizelge 7.1'de verilmiştir. Bu parametrelerin yanı sıra tahmini bölge sayısı ve limit parametre değerinin belirlenmesi algoritmaların performansı için önemli etkenlerdir.

Çizelge 7.1. Tüm algoritmalar için ortak kullanılan kontrol parametreleri

FEs (ön değerlendirme için)	50
FEs (değerlendirme için)	500
En küçük bölge boyutu	500
Harici arşiv boyutu *nEXA*	10
Sınıflandırılmamış piksel oranı	0.25

Bu bağlamda, segmentasyon sürecinin dört adımda tamamlandığı söylenebilir. Bu adımlar Şekil 7.6'da gösterilmektedir. Öncelikle 20 elemandan oluşan küçük bir koloni boyutu ile farklı parametre değerleri için IBMO çalıştırılmıştır. Farklı bölge sayılarını içeren $\{2, 3, 4, 5\}$ kümesinin her bir elemanı ve limit parametresinin farklı değerlerini içeren $\{0, 5, 10, 15, 20\}$ kümesinin her bir elemanı için algoritma teker teker çalıştırılmıştır. Böylece parametre değerlerinin kombinasyonlarından oluşan 20 adet bağımsız ön çalıştırma işlemi tüm test görüntülerine ayrı ayrı uygulanmıştır. Bu adımda

fonksiyon değerlendirme sayısı 50 olarak ayarlandığı için ön-çalıştırmaların zaman tüketimi çok azdır. Örneğin 3096 numaralı test görüntüsü için elde edilen ön-çalıştırma sonuçları Çizelge 7.2'de görülmektedir. Tutarlılık hataları, amaç fonksiyonlarının değerleri, tüketilen zaman ve çalışma sonunda bulunan uygun çözüm sayısı Çizelge 7.2'de verilmiştir.



Şekil 7.6. Önerilen segmentasyon metodunun akış diyagramı

İkinci adımda, ön-çalıştırma sonuçları kullanılarak tahmini bölge sayısı ve limit değerleri seçilir. İki gösterge dikkate alınarak parametre değerleri belirlenir. Bu göstergeler, segmentasyon kalitesini ifade eden toplam tutarlılık hatası ve küme doğruluk indeksi (*S_Dbw*)'dir. Çizelge 7.2'den görülebileceği gibi 3096 numaralı test görüntüsü için iki tane en iyi çalışma sonucu elde edilmiştir. Birinci en iyi çalıştırma sonucunda, en düşük toplam tutarlılık hatası 0.099622, bölge sayısı dört ve limit değeri 5'dir. İkinci en iyi çalıştırma sonucunda ise en iyi S_Dbw değeri 0.067871, bölge sayısı 3 ve limit değeri 5'dir. Tahmini bölge sayısı için iki çalıştırma arasında bölge sayısı daha büyük olanın değeri seçilir. Bu şekilde oluşabilecek bölge sayısı öngörülür. Bölge sayısını, olduğundan çok az veya çok fazla seçmek yakınsamayı yavaşlatır. Öngörülen değerin makul bir aralıkta seçilmesiyle segmentasyon süreci hızlandırılmış olur. Son-işleme aşamasında bölge birleştirme ve yeni bölgelerin eklenmesiyle nihai çıktı elde edilir. Limit değeri ise iki çalışmadaki limit değerlerinin ortalaması alınarak hesaplanır. Çalışmada kullanılan tüm test görüntüleri için seçilen kontrol parametrelerinin değerleri Çizelge 7.3'te görülmektedir.

Akış diyagramının üçüncü adımı algoritmaların 500 fonksiyon değerlendirmesi için çalıştırılmasıdır. Bu adımda her algoritma her bir test görüntüsü için 10 defa bağımsız olarak çalıştırıldı. Son olarak, son-işleme adımı gerçekleştirilir ve tüm piksellerin bir bölgeye atanması sağlanarak parçalanmış görüntü elde edilir. IBMO'nun 3096 numaralı test görüntüsü için elde edilen 10 bağımsız çalıştırmanın sonuçları Çizelge 7.4'te verilmektedir.

Çizelge 7.2. 3096 numaralı test görüntüsü için alınan ön-çalıştırma sonuçları

Koloni boyutu =20										
Bölge Sayısı	Limit	Uygun çözüm	LCE	GCE	BCE	Toplam Tutarlılık Hatası	Dağılma (S)	Yoğunluk (Dbw)	S_Dbw	Geçen Süre (s)
	0	1	0,034198	0,024060	0,047180	0,105438	0,914569	0,010385	0,924953	1,305310
	5	1	0,033655	0,022781	0,048486	0,104921	0,306425	0,001675	0,308100	0,918317
2	10	2	0,040758	0,030900	0,055720	0,127379	0,926912	0,026795	0,953708	1,048002
	15	3	0,033701	0,022547	0,049751	0,105999	0,686253	0,005141	0,691393	4,098121
	20	0								0,953178
	0	1	0,035481	0,031272	0,038274	0,105028	0,158427	0,243995	0,402422	1,468747
	5	1	0,028786	0,028371	0,303048	0,360205	0,067871	0,000000	0,067871	3,968405
3	10	1	0,033771	0,019680	0,060419	0,113869	0,380538	0,233965	0,614503	1,087864
	15	0								1,535403
	20	1	0,030549	0,030129	0,046495	0,107172	0,202062	0,088580	0,290642	1,589586
	0	0								1,768662
	5	1	0,031508	0,018884	0,049230	0,099622	0,491916	0,544024	1,035941	1,530317
4	10	0								1,552668
	15	0								1,412377
	20	0								1,705617
	0	0								1,827194
	5	0								1,696940
5	10	0								1,818752
	15	1	0,033145	0,017487	0,074201	0,124833	0,533543	0,337834	0,871377	2,130782
	20	1	0,030164	0,029318	0,161644	0,221125	0,067158	0,379882	0,447040	2,758318

Çizelge 7.3. Test görüntüleri için seçilen kontrol parametrelerinin değerleri

	3096	**42049**	**62096**	**86016**	**113016**	**135069**	**299091**
Tahmini Bölge sayısı	4	9	4	2	5	4	3
Limit	5	10	10	15	10	10	10

Çizelge 7.4. Önerilen segmentasyon yöntemi IBMO'nun 3096 numaralı test görüntüsüne 500 fonksiyon değerlendirmesi için uygulanması ile alınan sonuçlar

Koloni boyutu = 50										
	BÇS	BBS	LCE	GCE	BCE	Toplam Tutarlılık Hatası	Dağılma (S)	Yoğunluk (Dbw)	S_Dbw	Geçen süre (s)
Çalışma-1	6	2	0,031536	0,022609	0,041679	0,095824	0,369667	0,000000	0,369667	28,146870
Çalışma-2	3	2	0,028850	0,024367	0,029078	0,082295	0,189290	0,211740	0,401030	22,261440
Çalışma-3	5	2	0,033778	0,029902	0,034255	0,097935	0,836958	0,252941	1,089899	17,576708
Çalışma-4	1	2	0,033364	0,019479	0,058516	0,111359	0,304323	0,313017	0,617340	28,537790
Çalışma-5	6	2	0,026904	0,022389	0,026580	0,075873	0,189680	0,452645	0,642325	21,999283
Çalışma-6	4	2	0,029992	0,022558	0,035460	0,088010	0,905638	0,001843	0,907482	18,238793
Çalışma-7	4	2	0,030907	0,024931	0,033433	0,089271	0,348370	0,105275	0,453644	17,896590
Çalışma-8	3	2	0,037470	0,028424	0,049357	0,115250	0,158221	0,247190	0,405410	17,787398
Çalışma-9	4	2	0,029114	0,020498	0,036771	0,086383	0,496398	0,000044	0,496443	21,481115
Çalışma-10	7	2	0,029865	0,026297	0,029680	0,085842	0,484033	0,344284	0,828317	18,518198

BÇS : bastırılmamış çözüm sayısı
BBS : bulunan bölge sayısı

7.3.3. Deneysel Sonuçlar

Denetimsiz segmentasyon algoritmalarının değerlendirilmesi test görüntüleri için manuel olarak oluşturulmuş sonuçların kullanılmasıyla mümkündür. Bu nedenle önerilen segmentasyon yöntemi, Berkeley görüntü segmentasyonu veri tabanından (BSDS) (Martin ve ark. 2001) temin edilen çeşitli test görüntülerine uygulanmıştır. BSDS literatürde çokça kullanılan bir değerlendirme paketidir. BSDS veritabanı; görüntü segmentasyonu ve kenar tespiti ile ilgilenen araştırmacılar için hem gri-tonlamalı hem de renkli resimlerden oluşan 300 adet test görüntüsü içerir. Bunun yanı sıra, 30 insan denek yardımıyla oluşturulmuş el-çizimlerini içerir.

Çalışma kapsamında, segmentasyon çıktıları bölge-tabanlı tutarlılık ölçütleri kullanılarak değerlendirilmiştir. Bu ölçütler denekler tarafından çizilen segmentasyonlar ile algoritmaların bulduğu sonuçların bölgesel olarak ne kadar örtüşüp örtüşmediğini ortaya koymaktadır. Buna ek olarak, önerilen metottan başka üç farklı optimizasyon algoritması karşılaştırma amacıyla aynı test görüntüleri üzerinde çalıştırılmıştır. Bunlar NSGA-II, NSPSO ve FCM algoritmalarıdır. Tüm algoritmalar aynı öznitelik vektörü ve ortak kontrol parametreleri kullanılarak çalıştırılmıştır. Çalıştırılan üç algoritma çok-amaçlı optimizasyon algoritmasıdır. Bunlar IBMO, NSGA-II ve NSPSO olmak üzere görüntüde çekirdek noktalarını ve bu noktalara karşılık gelen bölgenin eşik benzerlik değerini tespit etmek amacıyla çalıştırılır. Daha sonra önerilen yöntem için geliştirilen aynı son-işleme aşaması atanmamış piksel kalmayıncaya kadar uygulanır. Diğer taraftan FCM için son işleme aşaması uygulanmaz. Çünkü herhangi bir bölgeye atanmayan piksel olmaz. FCM algoritması için tahmini bölge sayısı ve öznitelik vektörü görüntüyü bölütlemek için yeterlidir.

IBMO ve diğer tüm algoritmaların segmentasyon sonuçları 11 sütundan oluşan Çizelge 7.5'de verilmiştir. İlk sütun test görüntüsünün veri tabanındaki numarasını, ikinci sütun algoritmayı göstermektedir. Sonraki üç sütun sırasıyla LCE, GCE ve BCE tutarlılık hatalarını göstermektedir. Ayrıca toplam tutarlık hatasına altıncı sütunda yer verilmiştir. Amaç fonksiyonlarının (scattering ve density) hesaplanan değerleri sırasıyla sütun 7 ve sütun 8'de verilmektedir. Küme doğruluk indeksi S_Dbw bu değerlerin toplamıdır ve sütun 9'da ayrıca gösterilmiştir. Bu değerlerin tamamı son-işleme aşamasından önce elde edilmiştir.

Çok-amaçlı optimizasyon algoritmalarının çalışma prensipleri gereğince birden fazla sonuç üretmeleri beklenen bir durumdur. Bu durumda segmentasyon problemi için ifade edilen en iyi çözüm, bulunan çözüm kümesi içindeki en düşük toplam tutarlılık

hatasına sahip olan çözüm demektir. Böylece bir çalıştırmanın sonunda en iyi olarak seçilen çözüm hafızaya alınır. Çizelge 7.5'deki değerler algoritmaların her bir görüntü için 10 defa çalıştırılması sonucunda elde edilen en iyi çıktıyı temsil etmektedir. Bu çözümün elde edilmesi için harcanan süre ve tüm 10 çalışmanın ortalama süreleri sırasıyla çizelgenin 10. ve 11. sütunlarda gösterilmiştir. IBMO temelli önerilen segmentasyon metodu ve diğer karşılaştırma algoritmalarının tamamı Matlab ortamında geliştirilmiştir. Tüm algoritmalar Intel core i7-2600 3.40Ghz CPU, 10GB hafıza ve Windows işletim sistemi üzerinde çalıştırılarak çıktılar üretilmiştir.

Her bir görüntü için elde edilen en düşük toplam tutarlılık hatası çizelgede gri arkaplan rengi ile vurgulanmıştır. Önerilen metodun, bölge-tabanlı tutarlılık göstergelerine göre önemli derecede bir doğruluğa sahip olduğu anlaşılmaktadır. Üstelik IBMO, kullanılan tüm test görüntülerinde tüm algoritmalar arasında en iyi toplam hata değerine sahiptir. IBMO ve NSPSO algoritmaları, küme doğruluğu ve manuel etiketlemeli referans görüntülerle tutarlılığı dikkate alındığında makul bir çalışma süresi içinde başarılı segmentasyon sonuçları elde edebilmiş başarılı algoritmadır. Bununla birlikte, NSPSO her ne kadar NSGA-II'ye göre daha kararlı bir çalışma karakteristiğine sahip olsa da tüm çalıştırma sonuçları dikkate alındığında IBMO'nun daha başarılı ve daha yeknesak sonuçlar ürettiği görülür. Algoritmaların ürettiği sonuçların yeknesaklığı kutu diyagramları ile rahatça gözlemlenebilir. Bu amaçla, 10 bağımsız çalışma için toplam tutarlılık hatalarının kutu diyagramları Şekil 7.7'de gösterilmektedir. IBMO yerel arama prosedürü sayesinde kararlılığını koruyabilmiştir. Diğer taraftan, NSGA2 kısıt fonksiyonunu sağlayabilmek için çok fazla zaman tüketmek zorunda kalmıştır. Öyle ki bazı çalıştırmaların sonunda hiçbir uygun çözüme ulaşılamamıştır. Kıyaslanan diğer algoritmaların aksine FCM popülasyon-temelli olmayan bir metottur. Performansı büyük ölçüde başlangıç merkez noktalarına ve öznitelik vektörüne bağlı olarak değişebilen FCM algoritması, bu sayede tek-düze bir dağılım gösterebilmeyi başaran hızlı bir tekniktir. Fakat FCM'nin bir bölgede bulunan piksellerin bitişik olması gerekliliğiyle ilgili önemli bir dezavantajı vardır. Çünkü FCM, özellik olarak benzer ancak konum olarak ayrık olan pikselleri aynı bölgenin elemanları olarak kabul edebilir.

BCE değerlendirmesinde tüm resimler için en az beş farklı manuel referans çizim kullanılmıştır. Bu referans çizimler Şekil 7.8'de görülmektedir. LCE ve GCE hesaplamalarında bu şekilde ilk satırda gösterilen referans çizimler kullanılmıştır. Ayrıca Şekil 7.9'da önerilen metodun ve kıyaslanan tüm algoritmaların ürettiği segmentasyon sonuçları görülmektedir.

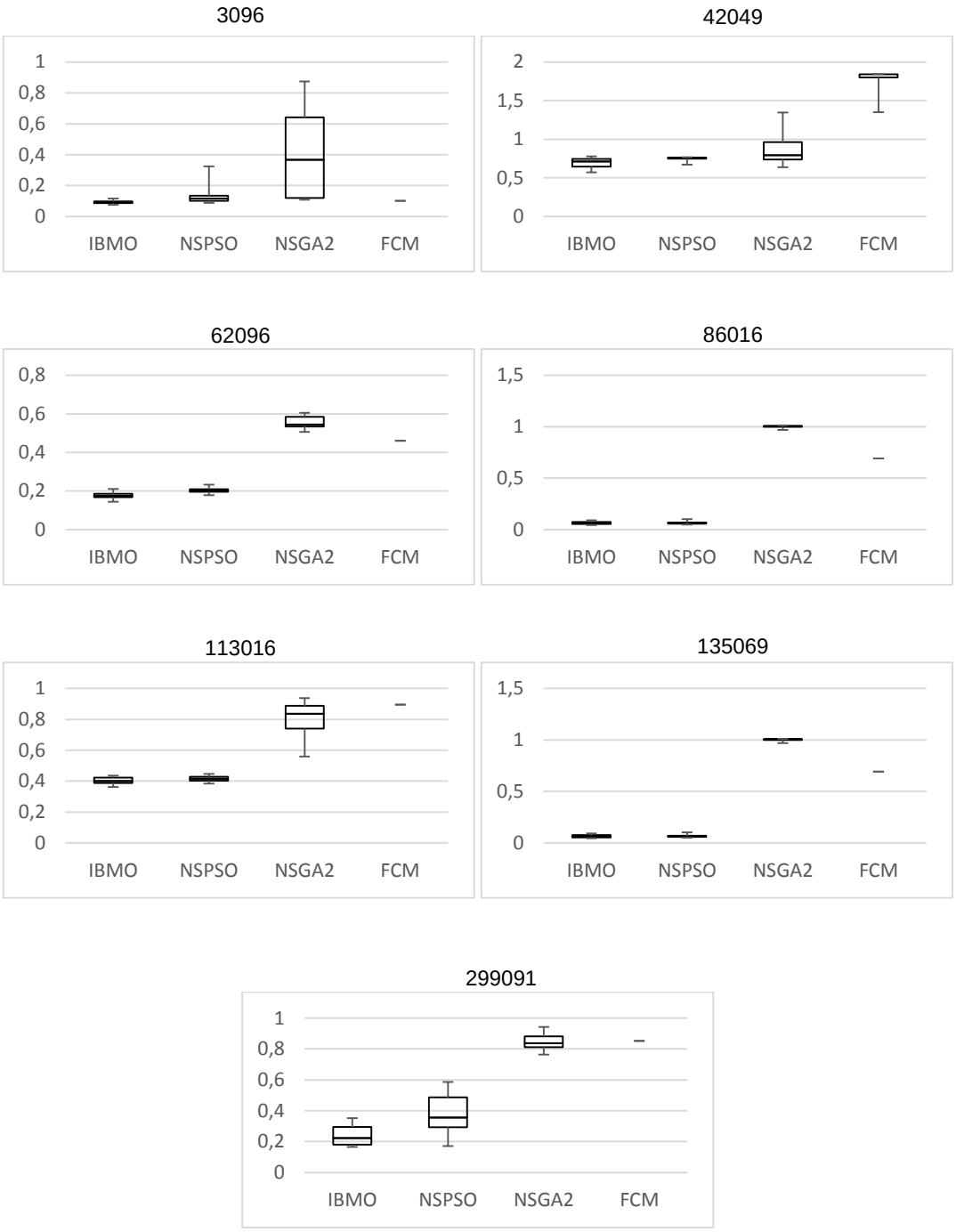
Çizelge 7.5. Önerilen yöntem ve diğer kıyaslama algoritmaları için hesaplanan değerlendirme ölçütlerinin sonuçları

BSDS Test Görüntü Numarası	Alg.	LCE	GCE	BCE	Toplam Tutarlılık Hatası	scattering	density	S_Dbw	Harcanan süre (s)	Ortalama süre (s)
3096	IBMO	0,026904	0,022389	0,026580	0,075873	0,189680	0,452645	0,642325	21,999283	21,244419
	NSPSO	0,029631	0,025557	0,032188	0,087375	0,250551	0,000000	0,250551	31,713330	24,772340
	NSGA2	0,032267	0,031382	0,044001	0,107650	0,017324	0,252256	0,269581	7,951744	224,787879
	FCM	0,035146	0,031644	0,034811	0,101600				4,297040	4,270320
42049	IBMO	0,127600	0,126887	0,315882	0,570370	0,204968	0,748724	0,953692	33,277100	29,966519
	NSPSO	0,129081	0,123707	0,416388	0,669176	0,193324	0,639812	0,833136	30,056306	30,924711
	NSGA2	0,180821	0,163823	0,291493	0,636137	0,000129	0,000000	0,000129	76,098348	59,246045
	FCM	0,377183	0,338840	0,632877	1,348900				18,689864	18,635147
62096	IBMO	0,050142	0,041711	0,052910	0,144763	0,080334	0,000000	0,080334	19,334017	18,580924
	NSPSO	0,068731	0,048576	0,061188	0,178495	0,022984	0,000000	0,022984	21,884963	20,111907
	NSGA2	0,067914	0,054074	0,384713	0,506700	0,000119	0,000000	0,000119	50,745152	62,421526
	FCM	0,102107	0,094006	0,264986	0,461098				10,385511	10,759900
86016	IBMO	0,020326	0,020326	0,273596	0,314249	0,182435	0,502460	0,684896	9,260838	8,257934
	NSPSO	0,023244	0,023244	0,272545	0,319032	0,209643	0,002889	0,212532	9,852829	9,069223
	NSGA2	0,029167	0,029167	0,273544	0,331877	0,123098	0,000000	0,123098	33,468635	236,781059
	FCM	0,048687	0,048686	0,296424	0,393797				4,271056	4,257568
113016	IBMO	0,090185	0,088903	0,183136	0,362224	0,204651	0,020822	0,225473	17,257824	16,756211
	NSPSO	0,096012	0,095509	0,191842	0,383363	0,192109	0,020362	0,212472	15,285761	16,801116
	NSGA2	0,152569	0,115728	0,291876	0,560174	0,012346	0,000000	0,012346	7,806258	1285,950599
	FCM	0,192610	0,183654	0,519677	0,895942				10,231946	10,431204
135069	IBMO	0,013899	0,013807	0,015201	0,042907	0,354371	0,436605	0,790976	23,437318	29,523659
	NSPSO	0,010745	0,009592	0,029849	0,050186	0,258097	0,000000	0,258097	51,923066	47,590460
	NSGA2	0,050848	0,038165	0,880366	0,969380	0,001322	0,729614	0,730936	314,049001	286,549701
	FCM	0,032415	0,020859	0,637633	0,690908				8,269594	8,252442
299091	IBMO	0,014711	0,014707	0,136737	0,166155	0,214982	0,060953	0,275935	15,233427	14,751328
	NSPSO	0,015828	0,015819	0,139554	0,171201	0,402444	0,002326	0,404770	16,336937	16,601441
	NSGA2	0,076569	0,076569	0,609592	0,762729	0,012849	0,000000	0,012849	4734,349180	4456,687256
	FCM	0,292942	0,140141	0,418282	0,851365				6,379537	6,349791

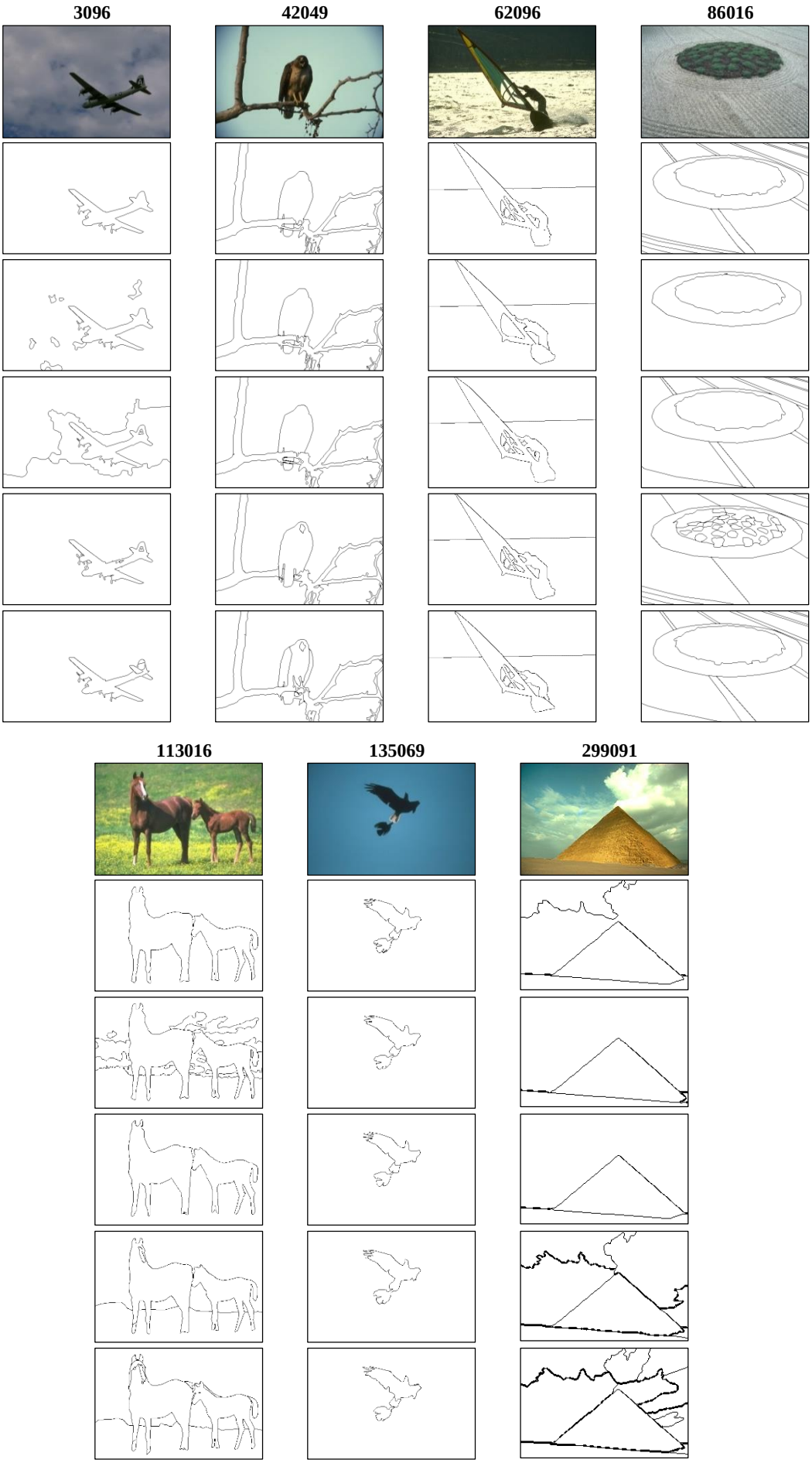
7.4. Bölüm Değerlendirmesi

Bu bölümde renkli görüntüler için genel amaçlı yeni bir segmentasyon algoritması önerilmiştir. Segmentasyon işlemi, renkli görüntüden çıkarılan özniteliklerin gruplanması yoluyla gerçekleştirilen çok-amaçlı bir kümeleme problemi olarak modellenmiştir. Belirleyici bir öznitelik çıkarma işlemi, görüntüde ilgilenilen bölgeleri elde etmek için son derece önemlidir. Doğa görüntüleri için önemli olan özelliklerden bazıları, her bir piksel için yerel homojenlik ve Gabor filtresi kullanılarak çıkartılmıştır. Önerilen yöntem dördüncü bölümde açıklanan çok-amaçlı optimizasyon algoritması IBMO ve çekirdekli bölge genişletme tekniğine dayanır. IBMO, önerilen çekirdekli bölge genişletme tekniğine üç önemli katkıda bulunur. Segmentasyon işlemine katkıları; çekirdek noktalarının optimal koordinatlarını belirlemek, optimal benzerlik eşik değerlerini tespit etmek ve birden fazla amacı eşzamanlı ve birbirinden bağımsız bir şekilde dikkate alarak değerlendirmek şeklinde özetlenebilir. Bu noktada; literatürde sıklıkla kullanılan küme doğrulama göstergelerinden biri olan S_Dbw ele alınmıştır. Bu gösterge normalde dağılma ve yoğunluk terimlerini birleştiren bir toplam fonksiyonu olarak çalışır. IBMO bu göstergeyi parçalayarak; iki bağımsız amaç fonksiyonu şeklinde değerlendirir.

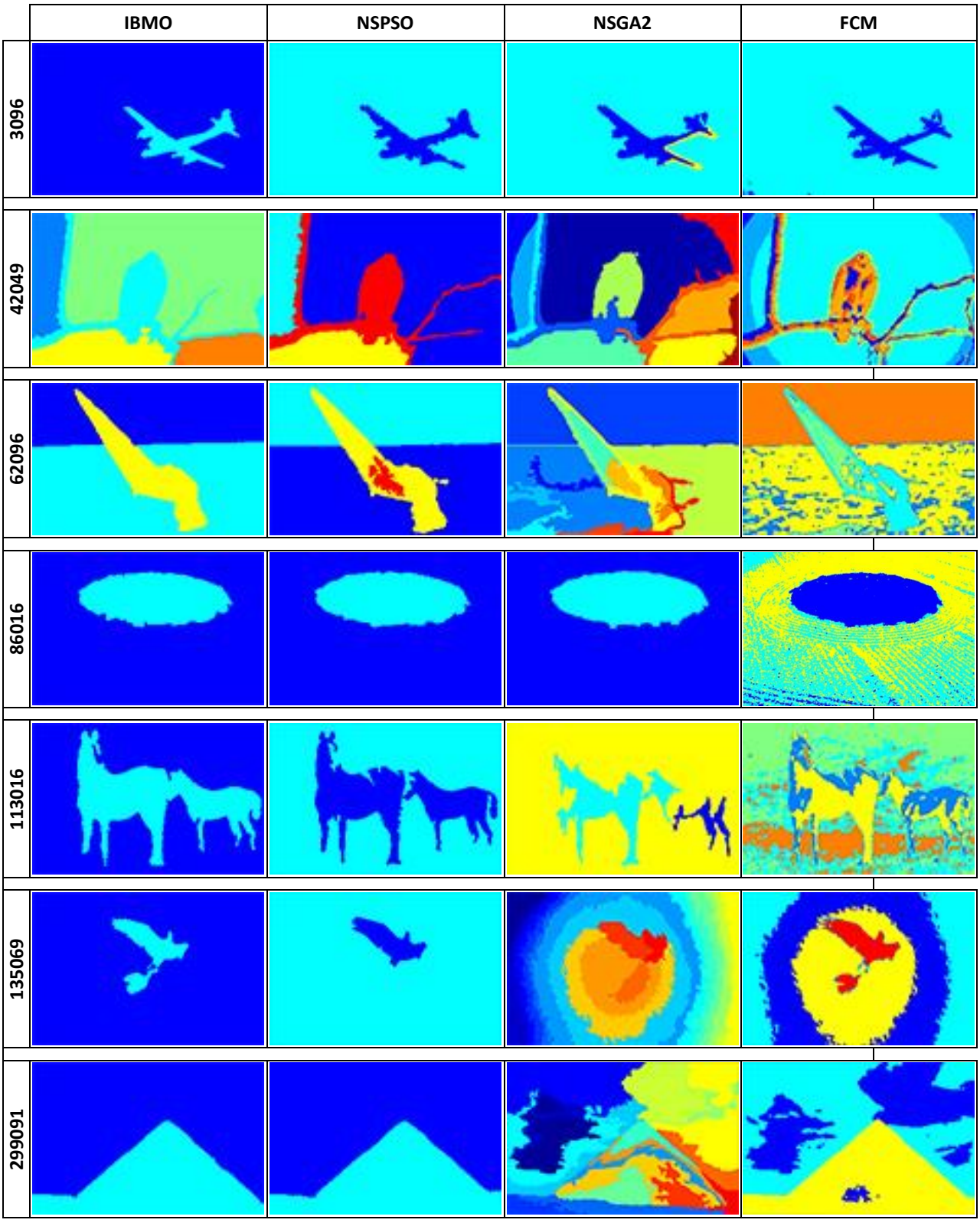
Böylece maksimum bölge-içi benzerlik ve minimum bölgeler-arası benzerlik şeklinde iki amaç fonksiyonu IBMO tarafından değerlendirmeye alınmıştır. Optimal çekirdek pozisyonları çok-maçlı optimizasyon algoritması ile arama uzayında araştırılırken; değerlendirme modifiye edilmiş çok çekirdekli bölge genişletme tekniği yardımıyla yapılmıştır. Bu işlem sırasında etiketlenmeden kalan pikseller son-işleme aşamasında değerlendirilmiş ve komşuluklarında bulunan ve kendilerine en fazla benzerlik gösteren bölgelere atanmaları sağlanmıştır. Önerilen algoritma Berkeley görüntü segmentasyonu veri tabanındaki test görüntülerine uygulanmıştır. Elde edilen segmentasyon sonuçlarının bölge-tabanlı tutarlılık göstergeleri LCE, GCE ve BCE değerleri hesaplanmıştır. Ayrıca önerilen yöntem üç başarılı algoritma ile karşılaştırılmıştır. Bu algoritmalar NSGA-II, NSPSO ve FCM'dir. Elde edilen başarılı sonuçlar önerilen yöntemin genel amaçlı renkli görüntü segmentasyonu için tutarlı bir doğruluğa sahip olduğunu göstermiştir. Bununla birlikte önerilen yöntem, çoklu çalıştırmalar sonucunda tüm algoritmalar içinde en kararlı sonuçları üretebilmiştir. Sonuç olarak; çok-amaçlı optimizasyon algoritması olan IBMO'nun görüntü segmentasyonuna uyarlanmasıyla, bu alan için iyi-tasarlanmış başarılı bir araç olarak kullanılabileceği açık bir şekilde ifade edilebilir.



Şekil 7.7. Toplam tutarlılık hatalarının 10 çalışma için kutu diyagramları



Şekil 7.8. Test görüntüleri ve denekler tarafından oluşturulan el-çizimi segmentasyon sonuçları



Şekil 7.9. Algoritmaların bulduğu segmentasyon sonuçları

8. SONUÇLAR

Görüntü segmentasyonu, birçok örüntü tanıma ve makine görüşü uygulamasının en temel ve kritik öneme sahip olan adımıdır. En basit tanımıyla bazı ölçütlere göre görüntüdeki piksellerin kümelenmesi şeklinde ele alınabilir. Bu bağlamda; başarılı bir segmentasyon algoritması için birbiriyle çelişen ve eş-zamanlı olarak sağlanması gereken birden fazla amaç fonksiyonu tanımlanabilir. Örneğin bir bölgeyi oluşturan piksellerin konumsal uyumluluğu, özniteliklerinin benzerliği ile birlikte düşünülmeli; bölgeler-arası farklılığa karşı bölge-içi benzerlik birlikte ele alınmalıdır. Aynı zamanda, bölütlenecek görüntü için birçok farklı öznitelik çıkarılabilir. Segmentasyonun kalitesini belirlemek için birçok doğruluk ve değerlendirme ölçütü birlikte kullanılabilir. Sürecin farklı aşamalarında birden fazla farklı optimizasyon metodu dâhil edilebilir. Uygulamanın türüne göre birden fazla gösterim biçimi üzerinde durulabilir. Bu etkenlerin hepsi segmentasyon işlemini çok-amaçlı optimizasyon problemi biçiminde tanımlamak için yeterlidir.

Bu tez çalışmasında görüntü segmentasyonu, bir optimizasyon problemi olarak ele alınmıştır. Yapay arı kolonisi optimizasyon algoritması üzerinde yoğunlaşılmış ve renkli görüntüler için yeni bir segmentasyon tekniğinin geliştirilmesi hedeflenmiştir. Bu amaçla, çalışmada segmentasyon süreci için bölge-tabanlı teknikler kullanılmıştır ve görüntü segmentasyonu için üç yeni algoritma önerilmiştir. İlk olarak tek-amaçlı temel ABC algoritması kümeleme yoluyla segmentasyona uyarlanmıştır. Daha sonra segmentasyon işleminin pek çok yönüyle çok-amaçlı bir optimizasyon problemi olarak sunulabilmesi bakımından; Pareto-temelli yeni bir çok-amaçlı optimizasyon algoritması geliştirilmiştir. İlkesel olarak temel ABC algoritmasına dayalı olarak geliştirilen bu algoritma "Çok Amaçlı Optimizasyon için İyileştirilmiş Yapay Arı Kolonisi Algoritması" (IBMO) olarak isimlendirilmiştir. Son aşamada; IBMO renkli görüntü segmentasyonu için uyarlanmış ve yeni melez bir yaklaşım geliştirilmiştir.

İlk aşamada tek-amaçlı ABC algoritması ile gerçekleştirilen segmentasyon işlemi, görüntüdeki benzer piksellerin en iyi şekilde gruplanmasını sağlayacak bir kümeleme problemi şeklinde tanımlanmıştır. Bu yaklaşımda, benzerlik özelliği olarak gri-tonlamalı görüntünün yoğunluk değerleri kullanılmıştır. Tek-amaçlı temel ABC algoritması uygun küme merkezlerinin optimal değerlerini bulmak için kullanılmıştır.

Uygunluk fonksiyonu olarak niceleme hatası seçilmiştir. Bu göstergeye ek olarak elde edilen sonuçlar üzerinden kümelemenin başarısını anlamak amacıyla d_{min} ve d_{max} olarak adlandırılan iki ölçüt değeri daha hesaplanmıştır. d_{min} farklı kümeler arasındaki minimum mesafeyi gösterirken; d_{max} ise tüm kümeler arasında maksimum küme-içi uzaklığı verir. Ayrıca elde edilen sonuçlar üç farklı kümeleme algoritması ile karşılaştırılmıştır. Bu algoritmalardan ikisi klasik danışmansız kümeleme algoritmaları olan k-ortalamalar ve bulanık c-ortalamalar algoritmalarıdır. Üçüncüsü ise PSO-temelli bir kümeleme algoritmasıdır. Önerilen yaklaşımın tüm gösterge değerleri bakımından daha başarılı sonuçlar elde ettiği görülmüştür.

İkinci aşamada; çok amaçlı optimizasyon problemlerinin çözümü için yapay arı kolonisini temel alan yeni bir melez algoritma geliştirilmiştir. Bu algoritma, başlangıç popülasyonunu üretmek için bir çeşitlendirme fonksiyonu kullanarak başlar. Pareto-üstünlüğü, kalabalıklık mesafesi ve haricen arşivleme gibi çok-amaçlı optimizasyon kavramlarından faydalanır. NSGA-II ve ABC algoritmalarının temel çalışma mekanizmalarını birleştirir. Ayrıca Pareto yüzeye yakınsamayı hızlandıran bir iyileştirme prosedürü içerir. IBMO olarak isimlendirilen algoritmanın performansı literatürdeki en iyi dört sezgisel algoritma ile karşılaştırılmıştır. Bunlar sırasıyla: NSGA-II, SPEA2, OMOPSO ve AbYSS algoritmalarıdır. Karşılaştırmaların doğruluğunu ortaya koymak açısından 26 farklı test problemi kullanılmıştır. Performans değerlendirmesi amacıyla sonuçlar dört farklı performans göstergesi (GD, SP, MS, HV) kullanılmıştır. Bütün bu değerlendirmelerin sonunda; IBMO, test problemlerinin büyük çoğunluğunda istatistiksel güven aralığında kalarak en iyi değerlere ulaşmayı başarmıştır.

Son aşamada çok-amaçlı optimizasyona dayanan yeni bir renkli görüntü segmentasyon algoritması önerilmiştir. Segmentasyon işlemi, renkli görüntüden çıkarılan özniteliklerin gruplanması yoluyla gerçekleştirilen çok-amaçlı bir kümeleme problemi olarak modellenmiştir. Çalışmada kullanılan piksel düzeyinde önemli öznitelikler yerel homojenlik ve Gabor filtresi kullanılarak çıkartılmıştır. Önerilen yöntem çok-amaçlı optimizasyon algoritması IBMO ve çekirdekli bölge genişletme tekniğine dayanır. Piksel düzeyinde elde edilen renk, doku ve gradiyent büyüklüklerine bağlı olarak hesaplanan yedi elemanlı bir öznitelik vektörü oluşturulur. Daha sonra, çekirdekli bölge genişletme tekniği ile benzerlik gösteren pikseller konumsal olarak sınıflandırılır. IBMO algoritması; çekirdek noktalarının optimal koordinatlarını

atanması, optimal benzerlik eşik değerlerini belirlenmesi ve segmentasyon kalitesini birden fazla küme doğrulama göstergesi ile değerlendirmesini sağlamak amacıyla renkli görüntü segmentasyonuna uyarlanmıştır. Ayrıca önerilen algoritma, Berkeley görüntü segmentasyon veri kümesindeki test görüntüleri üzerinde uygulanmıştır. Elde edilen segmentasyon sonuçlarının bölge-tabanlı tutarlılık göstergeleri hesaplanmıştır. Ayrıca sonuçlar NSGA-II, NSPSO ve FCM algoritmalarından aynı şartlar altında alınan segmentasyon sonuçlarıyla karşılaştırılmıştır. Elde edilen sonuçlardan, önerilen yöntemin genel amaçlı renkli görüntü segmentasyonu için tutarlı bir doğruluğa sahip olduğu ve performansının diğer algoritmalara göre daha iyi olduğu görülmüştür.

Yapılan tez çalışmasında; geliştirilen çok-amaçlı optimizasyon algoritması IBMO'nun performansı, belirli bir uygulama alanı için özelleştirilmemiş genel test görüntüleri üzerinde denenmiştir. Bu nedenle renkli görüntülerden çıkarılabilecek genel özellikler ile genel amaçlı bir öznitelik vektörü oluşturulmuştur. Öznitelik vektörünün uygulamaya yönelik olarak kurulması, kuşkusuz algoritmaların performansını olumlu yönde etkileyen önemli bir etkendir. Sonraki olası çalışmalarda, renkli medikal görüntüler gibi daha özel bir uygulama alanı için belirlenecek bir öznitelik vektörü kullanarak IBMO'nun performansı, klasik kümeleme tabanlı segmentasyon algoritmaları ile kıyaslanabilir. Bunula birlikte; bu tez çalışmasında bölge-tabanlı bir segmentasyon yaklaşımı benimsenmiştir. Bunun dışında, IBMO çoklu–eşikleme tekniklerine adapte edilerek de kullanılabilir. Bu amaçla klasik ve sezgisel yaklaşımlar arasındaki avantaj ve dezavantajlar ortaya konularak yeni çalışmalar yapılabilir.

KAYNAKLAR

Akay, B., Karaboğa, D., 2012, A Modified Artificial Bee Colony Algorithm for Real-Parameter Optimization, *Information Sciences*, 192, 120-142.

Alipoor, M., Imandoost, S., and Haddadnia, J., 2010, Designing Edge Detection Filters using Particle Swarm Optimization, *In Proceeding of 18th Iranian Conference on Electrical Engineering*, Iran, 548-552.

Aydın, D., 2010, An Efficient Ant-Based Edge Detector, *LNCS Transactions on Computational Collective Intelligence*, 1, 39-55.

Aydın, D., 2011, Sürü Zekâsı Yaklaşımlarının Renkli Görüntü Kesimlemeye Uyarlanması ve Tanıma Sistemleri Üzerinde Gerçekleştirimi, Doktora Tezi, *Ege Üniversitesi Fen Bilimleri Enstitüsü*, İzmir.

Beni G.. 1988, The concept of cellular robotic systems. *In Proceedings of the IEEE International Symposium on Intelligent Systems*, IEEE Press, Piscataway, NJ, 57–62.

Bezdek, J.C., 1981, Pattern Recognition With Fuzzy Objective Function Algorithms. *New York: Plenum Press*.

Bhandarkar, S.M. and Zhang, 1999, H., Image segmentation using evolutionary computation. *IEEE Transactions on Evolutionary Computation*, 3(1), 1–21.

Bhanu, B., Lee, S., and Ming, J., 1995, Adaptive image segmentation using a genetic algorithm. *IEEE Transactions on Systems Man and Cybernetics*, 25, 1543–1567.

Binh, T.T., Ulrich, K., 1997, Multiobjective evolution strategy with linear and nonlinear constraints, *15th IMACS World Congr Sci. Comp Modeling Appl. Math*, 357–362.

Blum, C., Li, X., 2008, Swarm intelligence in optimization, *Swarm Intelligence Natural Computing Series*, France, 43-85.

Bong, C.W., Mandava, R., 2010, Multiobjective Optimization Approaches in Image Segmentation – The Directions and Challenges, *Int. J. Advance. Soft Comput. Appl.*, 2 (1), 676-688.

Busin, L., Vandenbroucke, N. and Macaire, L., 2008, Color Spaces and Image Segmentation, *Advances in Imaging and Electron Physics*, 151, 66-169.

Cagnoni, S., Lutton, E., and Olague, G., 2007, Genetic and Evolutionary Computation for Image Processing and Analysis, *Hindawi Publishing*.

Casasent, D.P., Smokelin J.S., and Ye, A., 1992, Wavelet and Gabor transforms for detection, *Optical Engineering* 31, 1893-1898.

Cheng H.D., Jiang X.H., Sun Y., 2001, Wang Jingli, Color image segmentation: advances and prospects, *Pattern Recognition,* 34, 2259-2281.

Chengand, H.D., Sun, Y., 2000, A Hierarchical Approach to Color Image Segmentation Using Homogeneity, *IEEE Transactions on Image Processing,* 9(12), 2071-2082.

Clausi, D., Ed Jernigan, M., 2000, Designing Gabor filters for optimal texture separability, *Pattern Recognition*, 33(1) 1835-1849.

Coello A.C., 1998, A Comprehensive Survey of Evolutionary-Based Multiobjective Optimization Techniques, *Knowledge and Information Systems*, 1, 269-308.

Coello C., Carlos A.(Editor)., 2005, Application of Multi-Objective Evolionary Algorithms, *World Scientific Publishing Company*, River Edge, NJ, USA.

Dass, R., Priyanka, Devi, S., 2012, Image segmentation techniques, *International Journal of Electronics & Communication Technology*, 3 (1), 66-67.

Deb, K., Thiele, L., Laumanns, M., Zitzler, E., 2001, Scalable Test Problems for Evolutionary Multi-Objective Optimization, *Kanpur, India: Kanpur Genetic Algorithms Lab. (KanGAL), Indian Inst. Technol.*

Deb, K., 2002, A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II, *IEEE Transactions On Evolutionary Computation*, 6, 182-197.

Dias, A.H.F., Joao, A. de V., 2002, Multiobjective Genetic Algorithms Applied to Solved Optimization Problems, *IEEE Transactions On Magnetics*, 38(2), 1133-1136.

Dorigo, M., Maniezzo, V., Colorni A., 1996, The Ant System: Optimization by a colony of cooperating agents, *IEEE Transactions on Systems*, Man, and Cybernetics-Part B.

Dunn, D., and Higgins, W.E., 1995, Optimal Gabor filters for texture segmentation, *IEEE Trans. Image Processing* 4, 947-964.

Erkut, E., Tjandra, S.A., and Verter, V., 2007, Hazardous material transportation, Handbooks in, Operations Research and Management Science, G. Laporte and C. Barnhart (Eds.), *Elsevier Science*, North-Holland, 539-621.

Fan, S.-K.S. and Lin, Y., 2007, A Multi-level Thresholding Approach Using a Hybrid Optimal Estimation Algorithm, *Pattern Recognition Letters*, 28, 662-669.

Farmer, M.E., and Shugars, D., 2006, Application of genetic algorithms for wrapper-based image segmentation and classification. *IEEE Congress on Evolutionary Computation*, Vancouver BC, 1300–1307.

Feitosa, R.Q., Costa, G.A.O.P., and Cazes, T.B., 2006, A genetic approach for the automatic adaptation of segmentation parameters, *OBIA06*, Salzburg – Austria.

Fernandes, C., Ramos, V., and Rosa, A.C., 2005, Self-Regulated Artificial Ant Colonies on Digital Image Habitats, *International Journal of Lateral Computing*, 1(2), 1-8.

Fonseca, C., Flemming, P., 1998, Multiobjective optimization and multiple constraint handling with evolutionary algorithms - part II: Application example. *IEEE T SYST MAN CYB*, 28, 38–47.

Fu, K.S., Mui J.K., 1981, A survey on image segmentation, *Pattern Recognition*, 13, 3-16.

Gauch, J., Hsia Chi-Wan, 1992, A comparison of three color image segmentation algorithm in four color spaces, *Visual Communications and Image Processing '92*, SPIE Vol. 1818, 1168-1181.

Glover, F., Laguna, M., Martí, R., 2003, Scatter search, Theory and Applications, A. Ghosh and S. Tsutsui, Eds. *Advances in Evolutionary Computing, Springer-Verlag*, Berlin-Germany, 519–537.

Goldberg, D.E., and Richardson, J.J., 1987, Genetic Algorithms with sharing for multimodal function optimization. Genetic Algorithms and Their Applications, *Proceedings of the Second ICGA, Lawrence Erlbaum Associates*, Hillsdale, NJ, 41–49.

Gonzales, R.C., and Woods, R.E., 2002, Digital Image Processing 2nd Edition, *Prentice Hall*, Upper Saddle River - New Jersey, 567-642.

Halkidi, M., 2001, Clustering Validity Assessment: Finding the optimal partitioning of a data set, *Data Mining, 2001 ICDM 2001, Proceedings IEEE International Conference on*, San Jose CA, 187–194.

Hamamoto, Y., Uchimura, S., Watanabe, M., Yasuda, T., Mitani, Y., and Tomita, S., 1998, A Gabor filter-based method for recognizing handwritten numerals, *Pattern Recognition*, 31, 395-400.

Han, Y., and Shi, P., 2007, An Efficient Approach for Fish Bone Detection Based on Image Processing and Particle Swarm Clustering, D.-S. Huang, L. Heutte, and M. Loog (Eds.), *ICIC CCIS* 2, 940–948.

Huang, P., Cao, H. and Luo, S., 2008, An Artificial Ant Colonies Approach to Medical Image Segmentation, *Computer Methods and Programs in Biomedicine*, 92, 267-273.

Hoover, A., Kouznetsova, V., and Goldbaum, M., 2000, Locating Blood Vessels in Retinal Images by Piecewise Threshold Probing of a Matched Filter Response, *IEEE Transactions on Medical Imaging*, 19(3), 203–210.

Höppner, F., Klawonn, F., Kruse, R., and Runkler, T., 2000, Fuzzy Cluster Analysis, *John Wiley&Sons*, Chichester.

Hung, C.C. and Sun, M., 2010, Ant Colony Optimization for the K-means Algorithm in Image Segmentation, *The 48th Annual Southeast Regional Conference*, Oxford, MS, USA, 1-4.

Jain, A.K. and Farrokhnia, F., 1991, Unsupervised texture segmentation using Gabor filters, *Pattern Recognition*, 24, 1167-1186.

Jain, A.K., Murty, M.N., Flynn, P.J., 1999, Data Clustering: A Review, *ACM Computing Surveys*, 31(3), 264-323.

Jesmin, F., Khan, Sharif, M.A., Bhuiyan, and Reza R., Adhami, 2011, Image Segmentation and Shape Analysis for Road-Sign Detection, *IEEE Transactions On Intelligent Transportation Systems*, 12(1), 83-96.

Kang, W.X., Yang, Q.Q., Liang R.R., 2009, The Comparative Research on Image Segmentation Algorithms, *IEEE Conference on ETCS*, 703-707.

Karaboğa, D., 2005, An Idea Based On Honey Bee Swarm for Numerical Optimization, Technical Report *TR06, Erciyes University, Engineering Faculty, Computer Engineering Department*.

Karaboğa, D., Akay, B., 2009, A comparative study of Artificial Bee Colony algorithm, *Applied Mathematics and Computation* 214, 108–132.

Kennedy, J., and Eberhart, R.C., 1995, Particle swarm optimization. *IEEE int'l conf. on neural networks*, 4, 1942-1948.

Kim, K.M., Lee, C.S., Ha, Y.H., 1996, Color image quantization using weighted distortion measure of HVS color activity, *IEEE International Conference on Pattern Recognition*, 1035-1039.

Knowles, J.D., Corne D.W., 2000, Approximating the nondominated front using the Pareto archived evolution strategy. *IEEE T Evolut Comput*, 8(2), 149–172.

Koschan, A., and Abidi, M., 2008, Digital Color Image Processing, *John Wiley & Sons Inc*, New Jersey.

Kumar, R., 2010, Theory Of Automata, Languages and Computation, *Tata McGraw Hill Education Private Limited*, 390-411.

Kurpati, A., Azarm, S., Wu, J., 2002, Constraint handling improvements for multi-objective genetic algorithms, *Struct Multidiscip O*, 23(3), 204–213.

Kursawe, F., 1990, A variant of evolution strategies for vector optimization, in Parallel Problem Solving for Nature, H. Schwefel and R. Manner, (Eds.), *Springer-Verlag*, Berlin-Germany, 193–197.

Laptik, R., and Navakauskas, D., 2007, Application of Ant Colony Optimization for Image Segmentation, *Electronics and Electrical Engineering*, 8(80), 13-18.

Li, X., 2003, A Non-dominated Sorting Particle Swarm Optimizer for Multiobjective Optimization. *Genetic and Evolutionary Computation*, 2723, 37–48.

Liang, Y.C., Chen, A.H.L., and Chyu, C.C., 2006, Application of a Hybrid Ant Colony Optimization for the Multilevel Thresholding in Image Processing, King et al. (Eds.): *ICONIP Part II, LNCS 4233*, 1183–1192.

Lin, Z., Jin, J., and Talbot, H., 2000, Unseeded region growing for 3D image segmentation, *ACM International Conference Proceeding Series*, 9, 31-37.

Littmann, E., Ritter, H., 1997, Adaptive color segmentation a comparison of neural and statistical methods, *IEEE Trans. Neural Network*, 8(1), 175-185.

Lu, D.S., and Chen, C.C., 2008, Edge Detection Improvement by Ant Colony Optimization, *Pattern Recognition Letters*, 29(4), 416-425.

Lucchese, L., and Mitra, S.K., 2001, Color image segmentation: A state- of-the-art survey. *The Indian National Science Academy (INSA-A)*, 67(A2), 207.

Lucchese, L., Mitra, S.K., 1999, Advances in color image segmentation. *Global Telecommunications Conference Globecom*, Rio de Janeireo, 4, 2038-2044.

Macaire, L., Ultre, V., and Postaire, J.G., 1996, Determination of Compatibility Coefficients for Color Edge Detection by Relaxation, *International Conference on Image Processing*, C, 1045–1048.

MacQueen, J. B., 1967, Some Methods for classification and Analysis of Multivariate Observations. *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability 1. University of California Press*., 281–297

Mahmoud, K.R., El-Adawy, M., and Ibrahem, S.M.M., 2007, A comparison between circular and hexagonal array geometries for smart antenna systems using particle swarm optimization algorithm, *Progress In Electromagnetics Research*, 72(1), 75-90.

Maulik, U., Saha, I., 2009, Modified differential evolution based fuzzy clustering for pixel classification in remote sensing imagery, *Pattern Recognition*, 42(9), 2135-2149.

Maulik, U., Bandyopadhyay, S., 2002, Performance Evaluation of Some Clustering Algorithms and Validity Indices, 24(12), 1650–1654.

Martin, D., Fowlkes, C., Tal, D., and Malik, J., 2001, A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. *In Proceedings of the 8th International Conference Computer Vision*, 2(1), 416–423.

Marjan, M., Liub, S.H., Karaboğa D., Črepinšeka, M., 2015, On clarifying misconceptions when comparing variants of the Artificial Bee Colony Algorithm by offering a new implementation, *Information Sciences*, 291, 115–127.

Mobahi, H., Ahmadabadi, M.N., and Araabi, B.N., 2006, Swarm Contours: A Fast Self-Organization Approach for Snake Initialization, *Complexity*, 12, 41-52.

Mohsen, F.M.A., Hadhoud, M.M., and Amin K.M., 2010, Particle Swarm Optimization Method for Very Low Bit Rate Segmented Image Compression, *International Conference on Computer Engineering and Systems ICCES 2010*, Cairo, 15-20.

Mukhopadhyay A., Maulik U., Bandyopadhyay S., 2009, Multiobjective genetic clustering with ensemble among pareto front solutions: application to MRI brain image segmentation, *7th International Conference on Advances in Pattern Recognition*, Kolkata, 236-239.

Nakib, A., Oulhadj, H., Siarry, P., 2009, Image thresholding based on Pareto multiobjective optimization, *Engineering Applications of Artificial Intelligence*, 22, 236-249.

Namuduri, K. R., Mehrotra, R., and Ranganathan, N., 1994, Efficient computation of Gabor filter based multiresolution responses, *Pattern Recognition* 27, 925-938.

Narkhede, H.P., 2013, Review of Image Segmentation Techniques, *International Journal of Science and Modern Engineering (IJISME)*, 1(8), 2319-6386.

Nebro, A.J., Luna, F., Alba, E., Dorronsoro, B., Durillo, J.J., Beham, A., 2008, AbYSS: Adapting scatter search to multiobjective optimization. *IEEE Trans. Evol. Comput*, 12, 439–457.

Nevatia, A., 1977, Color Edge Detector and Its Use in Scene Segmentation, *IEEE Transaction on System Man Cybernetics SMC*, 7(11), 820–826.

Nezamabadi-pour, H., Saryazdi, S., Rashedi, E., 2006, Edge Detection Using Ant Algorithms, *Soft Computing*, 10, 623-628.

Niu, Y., Shen, L., 2007, The Optimal Multi-objective Optimization Using PSO in Blind Color Image Fusion, *International Conference on Multimedia and Ubiquitous Engineering MUE '07*, Seoul, 970 – 975.

Noor, A.I., Mokhtar, M.H., Rafiqul, Z.K., Pramod, K.M., 2012, Understanding Color Models: A Review, *ARPN Journal of Science and Technology*, 2 (3), 265-275.

Omran, M.G.H., 2004, PSO methods for pattern recognition and image processing, PhD thesis, *University Pretorial*.

Omran, M.G.H., 2005, Salman, A., and Engelbrecht, A.P., 2005, Dynamic Clustering Using Particle Swarm Optimization with Application in Image Segmentation, *Pattern Anal Applic*, 8(1), 332-344.

Omran, M.G.H., Engelbrecht, A.P., and Salman, A., 2005, Particle Swarm Optimization Method for Image Clustering, *International Journal of Pattern Recognition and Artificial Intelligence*, 19(3), 297–322.

Osyczka, A., 2002. Evolutionary algorithms for single and multicriteria design optimization. *Springer Verlag*, Berlin-Germany.

Osyczka, A., Kundo, S., 1995, A new method to solve generalized multicriteria optimization problems using a simple genetic algorithm. *Struct Optimization*, 10, 94–99.

Öztürk, C., 2011, Yapay Sinir Ağlarının Yapay Arı Kolonisi Algoritması ile Eğitilmesi, Doktora Tezi, *Erciyes Üniversitesi Fen Bilimleri Enstitüsü*, Kayseri, 29-41.

Pal, N.R., Pal, S.K., 1993, A review on image segmentation techniques, *Pattern Recognition* 29, 1277-1294.

Peng, H., Long, F., Chi, Z., and Su, W., 2000, A hierarchical distributed genetic algorithm for image segmentation. *The 2000 Congress on Evolutionary Computation*, California, USA, 1, 272–276.

Pignalberi, G., Cucchiara, R., Cinque, L., and Levialdi, S., 2003, Tuning range image segmentation by genetic algorithm, *EURASIP Journal on Applied Signal Processing*, 8, 780–790.

Ramos, V., and Almeida, F., 2000, Artificial Ant Colonies in Digital Image Habitats - A Mass Behavious Effect Study on Pattern Recognition, *Second International Workshop on Ant Algorithms*, Brussels-Belgium, 113-116.

Ramos, V., and Muge, F., 2000, Image colour segmentation by genetic algorithms. *The 11th Portuguese Conference on Pattern Recognition*, Porto, 125-129.

Rezaee, A., 2008, Extracting Edge of Images with Ant Colony, *Journal of Electrical Engineering*, 59(1), 57-59.

Rogowska, J., 2000, Overview and Fundemantals of Medical Image Segmentation, Handbook of Medical Imaging, *Academic Press*, 69-85.

Russ, J.C., 2007, The Image Processing Handbook, *CRC Press*, Boca Raton.

Sankur, B., Sezgin, M., 2004, A Survey Over Image Thresholding Techniques And Quantitative Performance Evaluation, *Journal of Electronic Imaging*, 13(1), 146-165.

Sağ, T. and Çunkaş, M., 2009, A tool for multiobjective evolutionary algorithms, *Advances in Engineering Software*, 40(9), 902–912.

Saha, U., Maulik, S., Bandyopadhyay, 2009, An improved multi-objective technique for fuzzy clustering with application to IRS image segmentation, *EvoWorkshops 2009 on Applications of Evolutionary Computing LNCS, Springer – Heidelberg*, 5484, 426-431.

Saha, S., Bandyopadhyay, S., 2010, A new symmetry based multiobjective clustering technique for automatic evolution of clusters, *Pattern Recognition*, 43(4), 738–751.

Saraswathi, S., Allirani, A., 2013, Survey on Image Segmentation via Clustering, *International Conference on Information Communication and Embedded Systems (ICICES)*, 331–335.

Sarker, R., 2002, Evolutionary Optimization, *Kluwer Academic Publishers*, Secasucus, NJ, USA.

Schaffer, J.D., 1985, Multiple objective optimization with vector evaluated genetic algorithms. *1st International Conference on Genetic Algorithms*, Hillsdale-New Jersey, 93–100.

Setayesh, M., Zhang, M., and Johnston, M., 2009, A New Homoheneity-based Approach to Edge Detection using PSO, *24th International Conference Image and Vision Computing*, New Zealand, 231-236.

Shapiro, Linda, G., and Stockman, G.C., 2002, Computer Vision, *Prentice Hall*.

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

Shi, Y., Eberhart, R.C., 2001, Fuzzy adaptive particle swarm optimization, The Congress on *Evolutionary Computation*, Seoul- Republic of Korea, 1, 101-106.

Shirakawa, S., Nagao, T., 2009, Evolutionary image segmentation based on multiobjective clustering, *The Congress on Evolutionary Computation (CEC '09),* Trondheim - Norway, 2466-2473.

Skarbek, W. and Koschan, A., 1994, Colour image segmentation: A survey. TechnicalReport94-32, *Computer Science Department, Berlin Technical University*.

Souza, J.G., and Costa, A.F., 2009, Unsupervised Data Clustering and Image Segmentation Using Natural Computing Techniques, *IEEE International Conference on Systems, Man, and Cybernetics*, 5045-5050.

Srinivas, N., Deb, K., 1995, Multiobjective function optimization using nondominated sorting genetic algorithms, *IEEE T Evolut Comput*, 2(3), 221–248.

Swagatam, D., Ajith A., and Amit K., 2008, Particle Swarm Optimization and Differential Evolution Algorithms: Technical Analysis, Applications and Hybridization Perspectives, Advances of Computational Intelligence in Industrial Systems, YingLiu et al. (Eds.), *Studies in Computational Intelligence, SpringerVerlag*, Germany.

Tamaki, H., Kita, H., Kobayashi, S., 1996, Multiobjective optimization by genetic algorithms: A review. *3rd IEEE Conf. Evol. Comput ICEC'96*, Nagoya-Japan, 517–522.

Tanaka, M., Watanabe, H., Furukawa, Y., Tanino, T., 1995, GA-based decision support system for multicriteria optimization. *IEEE International Conference on Systems, Man, and Cybernetics,* Piscataway-NJ, 2, 1556–1561.

Tang, L., Wang, X., 2013, A Hybrid Multiobjective Evolutionary Algorithm for Multiobjective Optimization Problems. *IEEE T Evolut Comput,* 17(1), 20–45.

Tao, W., Jin, H. and Liu, L., 2007, Object Segmentation Using Ant Colony Optimization and Fuzzy Entropy, *Pattern Recognition Letters*, 28, 788-796.

Teuner, A., Pichler, O. and Hosticka, B. J., 1995, Unsupervised texture segmentation of images using tuned matched Gabor filters, *IEEE Trans. Image Processing,* 4, 863-870.

Thilagamani, S., and Shanthi, N., 2011, A Survey on image segmentation through Clustering, *International Journal of Research and Reviews in Information Sciences*, 1(1), 14-17.

Tseng, C.C., Hsieh, J.G., and Jeng, J.H., 2009, Active Contour Model via Multi-population Particle Swarm Optimization, *Expert Systems with Applications*, 36, 5348-5352.

Turi, R.H., 2001, Clustering-Based Colour Image Segmentation, Ph.D Thesis, *Monash University School of Computer Science and Software Engineering*, Australia.

Viennet, R., Fontiex C., Marc, I., 1996, Multicriteria Optimization Using a Genetic Algorithm for Determining a Pareto Set. *J Syst Sci,* 27(2), 255–260.

Wang, Y.H., 2009, Tutorial: Image Segmentation, *National Taiwan University, Taipei, Taiwan, ROC.*

Wikipedia, http://en.wikipedia.org/wiki/Pareto_efficiency

Ye, Z., Zheng, Z., Yu, X., and Ning, X., 2005, Automatic Threshold Selection Based on Ant Colony Optimization Algorithm, *International Conference on Neural Networks and Brain*, Beijing, 2, 728-732.

Yu, Z., Au, O.C., Zou, R., Yu, W., and Tian J., 2010, An Adaptive Unsupervised Approach Toward Pixel Clustering and Color Image Segmentation, *Pattern Recognition*, 43, 1889-1905.

Zang, J., 2009, Image Segmentation Using Possibilistic C-Means Based on Particle Swarm Optimization, *WRI Global Congress on Intelligent Systems GCIS '09*, 1, 119-123.

Zhao, X., Lee, M.E., and Kim, S.H., 2008, Improved Image Thresholding using Ant Colony Optimization Algorithm, *International Conference on Advanced Language Processing and Web Information Technology 08*, Liaoning-China, 210-215pp.

Zingaretti, P., Tascini, G., and Regini. L., 2002, Optimising the colour image segmentation. *VIII Convegno dell'Associazione Italiana per l'Intelligenza Artificiale,* 1, 272-276.

Zitzler, E., Deb, K., Thiele, L., 2000, Comparison of multiobjective evolutionary algorithms: Empirical results. *IEEE T Evolut Comput 2000*; 8(2), 173–195.

Zitzler, E., Thiele, L., 1999, Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. *IEEE T Evolut Comput 1999*, 3, 257–271.

Zitzler, E., Laumanns, M. and Thiele, L., 2001, SPEA 2: Improving the Strength Pareto Evolutionary algorithm. *Technical report 103, Computer engineering and Networks Laboratory (TIK),* Swiss Federal Institute of Technology (ETH) Zurich.

EKLER

EK-1

Çok-Amaçlı Test Problemlerinin Tanımlamaları

Çizelge Ek-1.1. Kısıtsız iki-amaçlı test problemleri

<	D	Değişken Sınırları	Amaç Fonksiyonlar
Schaffer	1	[-10^3 , 10^3]	$f_1(x) = x^2$ $f_2(x) = (x-2)^2$
Fonseca	3	[-4 , 4]	$f_1(x) = 1 - exp\left(-\sum_{i=1}^{3}\left(x_i - \frac{1}{\sqrt{3}}\right)^2\right)$ $f_2(x) = 1 - exp\left(-\sum_{i=1}^{3}\left(x_i + \frac{1}{\sqrt{3}}\right)^2\right)$
Kursawe	3	[-5 , 5]	$f_1(x) = \sum_{i=1}^{n-1}\left(-10exp\left(-0.2\sqrt{{x_i}^2 + {x_{i+1}}^2}\right)\right)$ $f_2(x) = \sum_{i=1}^{n}(\lvert x_i\rvert^{0.8} + 5\sin x_i^3)$
ZDT1	30	[0 , 1]	$f_1(x) = x_1$ $f_2(x) = g(x)\left(1 - \sqrt{x_1/g(x)}\right)$ $g(x) = 1 + 9\sum_{i=2}^{D} x_i/(D-1)$
ZDT2	30	[0 , 1]	$f_1(x) = x_1$ $f_2(x) = g(x)\left(1 - \left(x_1/g(x)\right)^2\right)$ $g(x) = 1 + 9\sum_{i=2}^{D} x_i/(D-1)$
ZDT3	30	[0 , 1]	$f_1(x) = x_1$ $f_2(x) = g(x)\left(1 - \sqrt{\frac{x_1}{g(x)}} - \left(x_1/g(x)\right)sin(10\pi x_i)\right)$ $g(x) = 1 + 9\sum_{i=2}^{D} x_i/(D-1)$
ZTD4	10	$x_1 \in [0\,,1]$ $x_i \in [-5\,,5]$ $i = 2, \dots, D$	$f_1(x) = x_1$ $f_2(x) = g(x)\left(1 - \sqrt{x_1/g(x)}\right)$ $g(x) = 1 + 10\,(D-1) + \sum_{i=2}^{D}[x_i^2 - 10\cos(4\pi x_i)]$
ZDT6	10	[0 , 1]	$f_1(x) = 1 - exp(-4x_1)\,sin^6(6\pi x_1)$ $f_2(x) = g(x)\left(1 - \left(x_1/g(x)\right)^2\right)$ $g(x) = 1 + 9\left[\sum_{i=2}^{D} x_i/(D-1)\right]^{0.25}$

Çizelge Ek-1.2. Kısıtlı iki-amaçlı test problemleri

Problem	D	Değişken Sınırları	Amaç Fonksiyonlar
Constr	2	$x_1 \in [0.1 \quad , \quad 1]$ $x_2 \in [0 \quad , \quad 5]$	$f_1(x) = x_1$ $f_2(x) = (1 + x_2)/x_1$ $g_1(x) = x_2 + 9x_1 \geq 6$ $g_2(x) = -x_2 + 9x_1 \geq 1$
Srinivas	2	x_1, x_2 $\in [-20 \quad , \quad 20]$	$f_1(x) = 2 + (x_1 - 2)^2 + (x_2 - 1)^2$ $f_2(x) = 9x_1 - (x_2 - 1)^2$ $g_1(x) = x_1^2 + x_2^2 \leq 225$ $g_2(x) = x_1 - 3x_2 + 10 \leq 0$
Osyczka2	6	$x_1, x_2 \in [0 \quad , \quad 10]$ $x_3,\ x_5 \in [1 \quad , \quad 5]$ $x_4 \in [0 \quad , \quad 6]$ $x_6 \in [1 \quad , \quad 10]$	$f_1(x) = -[25(x_1 - 2)^2 + (x_2 - 2)^2 + (x_3 - 1)^2 + (x_4 - 4)^2 + (x_5 - 1)^2]$ $f_2(x) = x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 + x_6^2$ $g_1(x) = x_1 + x_2 - 2 \geq 0$ $g_2(x) = 6 - x_1 - x_2 \geq 0$ $g_3(x) = 2 - x_2 + x_1 \geq 0$ $g_4(x) = 2 - x_1 + 3x_2 \geq 0$ $g_5(x) = 4 - (x_3 - 3)^2 - x_4 \geq 0$ $g_6(x) = (x_5 - 3)^2 + x_6 - 4 \geq 0$
Golinski	7	$x_1 \in [2.6 \quad , \quad 3.6]$ $x_2 \in [0.7 \quad , \quad 0.8]$ $x_3 \in [17 \quad , \quad 28]$ $x_4 \in [7.3 \quad , \quad 8.3]$ $x_5 \in [7.3 \quad , \quad 8.3]$ $x_6 \in [2.9 \quad , \quad 3.9]$ $x_7 \in [5.0 \quad , \quad 5.5]$	$f_1(x) = 0.7854x_1{x_2}^2(10{x_3}^2/3 + 14.933x_3 - 43.0934) - 1.508x_1({x_6}^2 + {x_7}^2) + 7.477({x_6}^3 + {x_7}^3) + 0.7854(x_4{x_6}^2 + x_5{x_7}^2)$ $f_2(x) = \sqrt{(745x_4/x_2x_3)^2 + 1.69 \times 10^7}/0.1{x_6}^2$ $g_1(x) = \frac{1.0}{(x_1{x_2}^2x_3)} - \frac{1}{27} \leq 0$ $g_2(x) = \frac{1.0}{(x_1{x_2}^2{x_3}^2)} - \frac{1}{397.5} \leq 0$ $g_3(x) = \frac{{x_4}^3}{(x_2{x_3}^2{x_6}^4)} - \frac{1}{1.93} \leq 0$ $g_4(x) = \frac{{x_5}^3}{(x_2x_3{x_7}^4)} - \frac{1}{1.93} \leq 0$ $g_5(x) = x_2x_3 - 40 \leq 0$ $g_6(x) = x_1/x_2 - 12 \leq 0$ $g_7(x) = 5 - x_1/x_2 \leq 0$ $g_8(x) = 1.9 - x_4 + 1.5x_6 \leq 0$ $g_9(x) = 1.9 - x_5 + 1.1x_7 \leq 0$ $g_{10}(x) = f_2(x) \leq 1300$ $a = 745x_5/x_2x_3$ $b = 1.575 \times 10^8$ $g_{11}(x) = \frac{\sqrt{a^2 + b}}{0.1x_7^3} \leq 1100$
Tanaka	2	$x_1, x_2 \in [0\,, \pi]$	$f_1(x) = x_1$ $f_2(x) = x_2$ $g_1(x) = x_1^2 + x_2^2 - 1 - 0.1\cos\left(16\arctan\left(\frac{x_1}{x_2}\right)\right) \geq 0$ $g_2(x) = (x_1 - 0.5)^2 + (x_2 - 0.5)^2 \leq 0.5$
Binh2	2	$x_1 \in [0 \quad , \quad 5]$ $x_2 \in [0 \quad , \quad 3]$	$f_1(x) = 4x_1^2 + 4x_2^2$ $\quad f_2(x) = (x_1 - 5)^2 + (x_2 - 5)^2$ $g_1(x) = (x_1 - 0.5)^2 + {x_2}^2 - 25 \leq 0$ $g_2(x) = (x_1 - 8)^2 + (x_2 - 3)^2 - 7.7 \geq 0$

Çizelge Ek-1.3. DTLZ test problemleri

Problem	D	Değişken Sınırları	Amaç Fonksiyonlar
DTLZ1	7	$x_i \in [0\ ,1]$ $i = 1, \dots, D$	$f_1(x) = 0.5x_1x_2\big(1+g(x)\big)$ $f_2(x) = 0.5x_1(1-x_2)\big(1+g(x)\big)$ $f_3(x) = 0.5(1-x_1)\big(1+g(x)\big)$ $g(x) = 100\left[5+\sum_{i=3}^{D}\big((x_i-0.5)^2-\cos\big(20\pi(x_i-0.5)\big)\big)\right]$
DTLZ2	12	$x_i \in [0\ ,1]$ $i = 1, \dots, D$	$f_1(x) = \big(1+g(x)\big)\cos(0.5\pi x_1)\cos(0.5\pi x_2)$ $f_2(x) = \big(1+g(x)\big)\cos(0.5\pi x_1)\sin(0.5\pi x_2)$ $f_3(x) = \big(1+g(x)\big)\sin(0.5\pi x_1)$ $g(x) = \sum_{i=3}^{D}(x_i-0.5)^2$
DTLZ3	12	$x_i \in [0\ ,1]$ $i = 1, \dots, D$	$f_1(x) = \big(1+g(x)\big)\cos(0.5\pi x_1)\cos(0.5\pi x_2)$ $f_2(x) = \big(1+g(x)\big)\cos(0.5\pi x_1)\sin(0.5\pi x_2)$ $f_3(x) = \big(1+g(x)\big)\sin(0.5\pi x_1)$ $g(x) = 100\left[10+\sum_{i=3}^{D}\big((x_i-0.5)^2-\cos\big(20\pi(x_i-0.5)\big)\big)\right]$
DTLZ4	12	$x_i \in [0\ ,1]$ $i = 1, \dots, D$	$f_1(x) = \big(1+g(x)\big)\cos(0.5\pi {x_1}^{100})\cos(0.5\pi {x_2}^{100})$ $f_2(x) = \big(1+g(x)\big)\cos(0.5\pi {x_1}^{100})\sin(0.5\pi {x_2}^{100})$ $f_3(x) = \big(1+g(x)\big)\sin(0.5\pi {x_1}^{100})$ $g(x) = \sum_{i=3}^{D}(x_i-0.5)^2$
DTLZ5	12	$x_i \in [0\ ,1]$ $i = 1, \dots, D$	$f_1(x) = \big(1+g(x)\big)\cos(0.5\pi\theta_1)\cos(0.5\pi\theta_2)$ $f_2(x) = \big(1+g(x)\big)\cos(0.5\pi\theta_1)\sin(0.5\pi\theta_2)$ $f_3(x) = \big(1+g(x)\big)\sin(0.5\pi\theta_1)$ $g(x) = \sum_{i=3}^{D}(x_i-0.5)^2$ $\theta_i = \pi\big(1+2x_ig(x)\big)/\big(4\big(1+g(x)\big)\big)$
DTLZ6	12	$x_i \in [0\ ,1]$ $i = 1, \dots, D$	$f_1(x) = \big(1+g(x)\big)\cos(0.5\pi\theta_1)\cos(0.5\pi\theta_2)$ $f_2(x) = \big(1+g(x)\big)\cos(0.5\pi\theta_1)\sin(0.5\pi\theta_2)$ $f_3(x) = \big(1+g(x)\big)\sin(0.5\pi\theta_1)$ $g(x) = \sum_{i=3}^{D} {x_i}^{0.1}$ $\theta_i = \pi\big(1+2x_ig(x)\big)/\big(4\big(1+g(x)\big)\big)$
DTLZ7	22	$x_i \in [0\ ,1]$ $i = 1, \dots, D$	$f_1(x) = x_1$ $f_2(x) = x_2$ $f_3(x) = \big(1+g(x)\big)h(x)$ $g(x) = 1+9\sum_{i=3}^{D}x_i/20$ $h(x) = 3-\sum_{i=1}^{2}f_i.\big(1+\sin(3\pi f_i)\big)/\big(1+g(x)\big)$

Çizelge Ek-1.4. Viennet test ailesi ve Tamaki test problemleri

Problem	D	Değişken Sınırları	Amaç Fonksiyonlar
Viennet	2	$x_1, x_2 \in [-2, 2]$	$f_1(x) = {x_1}^2 + (x_2 - 1)^2$ $f_2(x) = {x_1}^2 + (x_2 + 1)^2 + 1$ $f_3(x) = (x_1 - 1)^2 + {x_2}^2 + 2$
Viennet2	2	$x_1, x_2 \in [-4, 4]$	$f_1(x) = (x_1 - 2)^2/2 + (x_2 + 1)^2/13 + 3$ $f_2(x) = (x_1 + x_2 - 3)^2/36 + (2x_2 - x_1)^2/8 - 17$ $f_3(x) = (x_1 + 2x_2 - 1)^2/175 + (2{x_2}^2 - x_1)^2/17 - 13$
Viennet3	2	$x_1, x_2 \in [-3, 3]$	$f_1(x) = ({x_1}^2 + {x_2}^2)/2 + \sin({x_1}^2 + {x_2}^2)$ $f_2(x) = (3x_1 - 2x_2 + 4)^2/8 + (x_1 - x_2 + 1)^2/27 - 15$ $f_3(x) = 1/({x_1}^2 + {x_2}^2 + 1) - 1.1exp(-{x_1}^2 - {x_2}^2)$
Viennet4	2	$x_1, x_2 \in [-4, 4]$	$f_1(x) = (x_1 - 2)^2/2 + (x_2 + 1)^2/13 + 3$ $f_2(x) = (x_1 + x_2 - 3)^2/175 + (2x_2 - x_1)^2/17 - 13$ $f_3(x) = (3x_1 + 2x_2 + 4)^2/8 + (x_1 - x_2 + 1)^2/27 + 15$ $g_1(x) = 4x_1 + x_2 - 4 < 0$ $g_2(x) = x_1 + 1 > 0$ $g_3(x) = x_1 - x_2 - 2 < 0$
Tamaki	3	$x_1, x_2, x_3 \in [0, 1]$	$f_1(x) = x_1$ $f_2(x) = x_2$ $f_3(x) = x_3$ $g(x) = {x_1}^2 + {x_2}^2 + {x_3}^2 - 1 \leq 0$

EK-2

50.000 Fonksiyon Değerlendirmesi için Hesaplanan Performans Göstergeleri

Çizelge Ek-2.1. GD göstergesi için ortanca ($\tilde{x}$) ve çeyrekler açıklığı (*IQR*) değerleri (50000 fonksiyon değerlendirmesi)

Problem	IBMO		OMOPSO		AbYSS		NSGA2		SPEA2		
	$\tilde{x}$	*IQR*	$\tilde{x}$	*IQR*	$\tilde{x}$	*IQR*	$\tilde{x}$	*IQR*	$\tilde{x}$	*IQR*	
Schaffer	2,255E-04	2,510E-05	2,260E-04	2,647E-05	2,337E-04	1,858E-05	1,691E-02	4,640E-02	1,507E-02	4,364E-02	+
Fonseca	1,281E-04	9,087E-06	1,212E-04	8,787E-06	1,924E-04	2,555E-05	3,475E-04	3,970E-05	2,164E-04	2,757E-05	+
Kursawe	1,288E-04	1,109E-05	4,039E-04	7,752E-05	1,381E-04	1,234E-05	2,081E-04	3,264E-05	1,533E-04	1,815E-05	+
ZDT1	6,561E-05	2,787E-05	1,256E-04	3,455E-05	1,546E-04	2,318E-05	1,923E-04	4,928E-05	1,730E-04	1,745E-05	+
ZDT2	4,658E-05	2,893E-06	5,487E-05	7,645E-06	5,324E-05	1,307E-05	1,294E-04	3,862E-05	6,739E-05	2,746E-05	+
ZDT3	1,592E-04	2,342E-05	1,943E-04	1,548E-05	1,985E-04	1,644E-05	2,102E-04	1,299E-05	2,147E-04	1,542E-05	+
ZDT4	7,272E-05	3,019E-05	5,651E-01	4,885E-01	2,435E-04	1,333E-04	1,759E-04	7,532E-05	1,882E-04	5,833E-05	+
ZDT6	1,497E-04	1,112E-05	1,732E-02	3,782E-02	1,633E-04	1,196E-05	1,597E-04	1,841E-05	1,672E-04	1,703E-05	+
Constr	1,428E-04	2,436E-05	1,315E-04	1,688E-05	1,797E-04	3,062E-05	2,987E-04	4,231E-05	1,957E-04	2,215E-05	+
Srinivas	4,299E-05	1,893E-05	4,419E-05	1,624E-05	4,571E-05	1,517E-05	1,948E-04	4,105E-05	1,140E-04	2,731E-05	+
Osycka2	4,738E-03	3,496E-03	1,819E-03	2,287E-03	1,138E-03	1,697E-02	1,014E-03	7,722E-05	1,344E-03	1,540E-04	+
Golinski	3,121E-04	2,913E-05	3,524E-04	5,040E-05	3,081E-04	2,423E-05	3,262E-04	3,013E-05	2,256E-04	2,109E-05	+
Tanaka	6,653E-04	3,983E-04	7,743E-04	9,179E-05	7,813E-04	6,853E-05	8,115E-04	9,049E-05	7,159E-04	8,757E-05	+
Binh2	6,732E-04	2,530E-05	1,253E-03	3,337E-05	1,251E-03	3,795E-05	1,386E-03	7,615E-05	1,586E-03	5,363E-05	+
DTLZ1	2,668E-03	1,987E-03	9,276E+00	1,354E+00	5,503E-04	6,508E-05	1,343E-03	1,747E-04	4,786E-01	9,615E-01	+
DTLZ2	1,644E-04	1,613E-05	6,973E-04	6,129E-05	1,604E-04	1,396E-05	1,218E-03	2,494E-04	3,034E-04	4,006E-05	+
DTLZ3	5,964E-02	8,691E-02	3,438E+01	6,029E+00	4,176E-03	7,461E-02	6,292E-03	1,849E-02	8,352E+00	2,948E+00	+
DTLZ4	6,342E-04	6,125E-05	1,236E-03	1,637E-04	6,207E-04	1,477E-05	2,049E-03	2,050E-04	7,651E-04	5,406E-05	+
DTLZ5	1,110E-01	2,428E-03	2,524E-04	1,439E-06	2,517E-04	1,273E-06	6,114E-04	4,204E-05	2,502E-04	2,646E-06	+
DTLZ6	4,292E-01	1,117E-02	2,526E-04	1,098E-06	4,670E-03	2,560E-03	2,005E-03	2,369E-03	1,336E-01	8,580E-03	+
DTLZ7	4,267E-04	1,201E-04	1,066E-03	1,537E-04	4,060E-04	2,327E-04	2,665E-03	5,604E-04	8,379E-04	9,706E-05	+
Viennet1	4,342E-04	7,385E-05	6,809E-04	8,280E-05	4,957E-04	6,756E-05	3,292E-03	7,982E-04	8,980E-04	7,123E-05	+
Viennet2	1,270E-04	3,828E-05	1,483E-04	3,264E-05	1,713E-04	6,093E-05	7,205E-04	3,641E-04	1,435E-04	1,448E-05	+
Viennet3	3,160E-05	1,049E-05	3,115E-05	5,285E-06	3,243E-05	5,606E-06	2,380E-04	5,885E-05	4,171E-05	6,207E-06	+
Viennet4	1,003E-04	2,006E-05	9,072E-05	8,155E-06	8,952E-05	1,037E-05	4,709E-04	1,287E-04	1,078E-04	1,273E-05	+
Tamaki	1,428E-03	1,997E-04	7,394E-02	1,414E-04	7,287E-02	3,151E-04	1,642E-01	1,112E-03	7,333E-02	1,510E-04	+

Çizelge Ek-2.2. SP göstergesi için ortanca ($\tilde{x}$) ve çeyrekler açıklığı (*IQR*) değerleri (50000 fonksiyon değerlendirmesi)

Problem	IBMO		OMOPSO		AbYSS		NSGA2		SPEA2		
	$\tilde{x}$	*IQR*	$\tilde{x}$	*IQR*	$\tilde{x}$	*IQR*	$\tilde{x}$	*IQR*	$\tilde{x}$	*IQR*	
Schaffer	1,112E-01	2,865E-02	1,084E-01	2,437E-02	1,162E-01	3,507E-02	7,611E-01	3,435E-01	7,470E-01	2,923E-01	+
Fonseca	6,211E-02	1,822E-02	5,457E-02	1,488E-02	8,698E-02	1,607E-02	3,741E-01	6,559E-02	1,242E-01	1,994E-02	+
Kursawe	2,242E-01	5,995E-03	2,731E-01	3,417E-02	2,302E-01	6,644E-03	5,437E-01	9,784E-02	2,890E-01	1,449E-02	+
ZDT1	1,541E-01	2,121E-02	7,763E-02	1,677E-02	8,170E-02	1,823E-02	3,871E-01	5,823E-02	1,275E-01	1,819E-02	+
ZDT2	1,364E-01	1,605E-02	7,356E-02	1,858E-02	8,592E-02	1,920E-02	4,074E-01	7,563E-02	1,263E-01	1,992E-02	+
ZDT3	2,821E-01	8,374E-02	1,207E-01	1,926E-02	1,104E-01	2,169E-01	3,844E-01	6,655E-02	1,578E-01	2,362E-02	+
ZDT4	2,392E-01	5,043E-02	8,461E-01	1,319E-01	8,899E-02	2,553E-02	4,165E-01	7,331E-02	1,085E-01	1,859E-02	+
ZDT6	8,128E-02	1,263E-02	8,824E-01	1,224E+00	6,469E-02	1,365E-02	4,122E-01	7,221E-02	1,217E-01	1,856E-02	+
Constr	2,498E-01	9,214E-02	9,666E-02	2,041E-02	1,431E-01	1,978E-02	3,950E-01	6,017E-02	5,236E-01	3,794E-02	+
Srinivas	7,456E-02	2,183E-02	7,069E-02	1,542E-02	6,973E-02	1,379E-02	3,591E-01	5,079E-02	1,482E-01	1,431E-02	+
Osycka2	3,123E-01	2,033E-01	6,587E-01	1,784E-01	3,757E-01	2,797E-01	6,445E-01	1,243E-01	5,970E-01	1,298E-01	+
Golinski	1,185E-01	3,877E-02	2,270E-01	4,492E-02	9,871E-02	1,913E-02	5,252E-01	9,539E-02	6,926E-01	3,967E-02	+
Tanaka	7,426E-01	1,695E-01	2,636E-01	4,739E-02	2,403E-01	4,495E-02	3,566E-01	5,173E-02	2,327E-01	4,012E-02	+
Binh2	3,888E-01	1,501E-02	3,122E-01	1,909E-02	3,127E-01	2,107E-02	5,415E-01	4,507E-02	5,531E-01	2,457E-02	+
DTLZ1	1,805E-01	9,150E-02	5,371E-01	1,082E-01	5,536E-01	3,115E-02	4,939E-01	6,106E-02	1,707E+00	4,271E-01	+
DTLZ2	4,207E-01	2,891E-02	3,730E-01	2,184E-02	5,522E-01	2,909E-02	5,134E-01	5,199E-02	8,150E-02	7,548E-03	+
DTLZ3	8,015E-01	2,051E-01	7,842E-01	1,111E-01	7,275E-01	2,428E-01	5,800E-01	9,881E-02	1,324E+00	1,728E-01	+
DTLZ4	4,173E-01	6,182E-02	9,295E-01	4,138E-01	5,121E-01	3,297E-02	4,972E-01	6,418E-02	8,120E-02	5,898E-03	+
DTLZ5	6,926E-01	3,824E-02	1,511E-01	1,029E-02	1,329E-01	1,080E-02	4,323E-01	6,491E-02	1,918E-01	1,329E-02	+
DTLZ6	7,228E-01	3,582E-02	1,128E-01	6,332E-03	4,917E-01	9,090E-02	4,917E-01	1,105E-01	6,442E-01	3,783E-02	+
DTLZ7	5,095E-01	6,265E-02	4,007E-01	2,239E-02	5,866E-01	1,379E-01	4,792E-01	5,570E-02	2,461E-01	1,255E-02	+
Viennet1	4,645E-01	2,496E-02	4,305E-01	2,705E-02	5,823E-01	2,991E-02	4,593E-01	6,091E-02	1,465E-01	7,754E-03	+
Viennet2	4,517E-01	3,407E-02	4,537E-01	2,457E-02	4,302E-01	3,194E-02	4,711E-01	7,230E-02	2,003E-01	1,429E-02	+
Viennet3	3,325E-01	3,821E-02	3,447E-01	2,059E-02	2,942E-01	2,409E-02	4,124E-01	6,276E-02	5,602E-01	1,484E-02	+
Viennet4	5,851E-01	2,937E-02	5,462E-01	2,624E-02	6,703E-01	2,801E-02	5,288E-01	7,945E-02	3,281E-01	1,800E-02	+
Tamaki	5,607E-01	1,087E-01	4,894E-01	2,154E-02	6,517E-01	2,574E-02	7,064E-01	4,212E-02	2,728E-01	9,564E-03	+

Çizelge Ek-2.3. MS göstergesi için ortanca ($\tilde{x}$) ve çeyrekler açıklığı (*IQR*) değerleri (50000 fonksiyon değerlendirmesi)

Problem	IBMO		OMOPSO		AbYSS		NSGA2		SPEA2		
	$\tilde{x}$	*IQR*	$\tilde{x}$	*IQR*	$\tilde{x}$	*IQR*	$\tilde{x}$	*IQR*	$\tilde{x}$	*IQR*	
Schaffer	1,000E+00	7,837E-05	1,000E+00	4,499E-05	1,000E+00	1,824E-05	9,442E-01	9,164E-02	9,291E-01	1,092E-01	+
Fonseca	1,000E+00	2,488E-04	1,000E+00	4,106E-05	1,000E+00	0,000E+00	1,000E+00	0,000E+00	1,000E+00	1,081E-04	+
Kursawe	9,998E-01	2,483E-04	9,926E-01	4,550E-03	1,000E+00	1,311E-05	1,000E+00	4,358E-05	9,999E-01	1,007E-04	+
ZDT1	1,000E+00	0,000E+00	1,000E+00	0,000E+00	1,000E+00	1,274E-06	1,000E+00	2,251E-05	9,999E-01	1,381E-04	+
ZDT2	1,000E+00	0,000E+00	1,000E+00	0,000E+00	1,000E+00	8,286E-07	1,000E+00	1,390E-05	9,999E-01	5,808E-05	+
ZDT3	9,999E-01	2,464E-03	9,995E-01	6,975E-04	9,999E-01	2,053E-01	9,999E-01	5,727E-05	9,998E-01	1,744E-04	+
ZDT4	9,997E-01	2,943E-04	7,071E-01	0,000E+00	9,993E-01	8,246E-04	9,995E-01	4,602E-04	9,993E-01	6,389E-04	+
ZDT6	1,000E+00	0,000E+00	1,000E+00	0,000E+00	9,994E-01	9,038E-05	9,996E-01	1,487E-04	9,994E-01	1,941E-04	+
Constr	9,110E-01	8,802E-02	9,976E-01	2,618E-03	9,995E-01	5,934E-04	9,986E-01	1,870E-03	9,973E-01	3,709E-03	+
Srinivas	9,810E-01	1,260E-02	9,910E-01	5,351E-03	9,938E-01	3,996E-03	9,935E-01	3,471E-03	9,924E-01	4,234E-03	+
Osycka2	8,128E-01	1,968E-01	8,759E-01	2,947E-01	8,207E-01	3,777E-01	9,950E-01	1,854E-01	8,127E-01	2,321E-01	+
Golinski	9,684E-01	4,090E-02	9,952E-01	8,048E-03	1,000E+00	2,121E-03	9,974E-01	1,353E-02	9,855E-01	2,990E-02	+
Tanaka	9,665E-01	5,070E-02	9,943E-01	3,236E-03	9,982E-01	3,187E-04	9,981E-01	5,481E-04	9,976E-01	1,702E-03	+
Binh2	7,695E-01	1,419E-04	8,111E-01	0,000E+00	8,111E-01	4,235E-10	8,111E-01	1,988E-10	8,111E-01	5,356E-10	+
DTLZ1	1,000E+00	0,000E+00	1,000E+00	0,000E+00	1,000E+00	0,000E+00	1,000E+00	0,000E+00	1,000E+00	0,000E+00	-
DTLZ2	1,000E+00	0,000E+00	1,000E+00	0,000E+00	1,000E+00	0,000E+00	1,000E+00	2,220E-16	1,000E+00	4,421E-06	-
DTLZ3	1,000E+00	0,000E+00	1,000E+00	0,000E+00	1,000E+00	0,000E+00	1,000E+00	0,000E+00	1,000E+00	0,000E+00	-
DTLZ4	1,000E+00	0,000E+00	1,000E+00	0,000E+00	1,000E+00	0,000E+00	1,000E+00	0,000E+00	1,000E+00	0,000E+00	-
DTLZ5	6,104E-01	1,311E-02	1,000E+00	0,000E+00	1,000E+00	1,439E-08	1,000E+00	1,998E-15	1,000E+00	1,099E-05	+
DTLZ6	8,171E-01	0,000E+00	1,000E+00	1,110E-16	1,000E+00	0,000E+00	1,000E+00	1,774E-08	1,000E+00	4,589E-09	+
DTLZ7	9,984E-01	7,952E-03	9,957E-01	3,671E-03	7,075E-01	4,078E-01	9,998E-01	7,590E-05	9,947E-01	1,598E-03	+
Viennet1	9,997E-01	8,670E-04	9,999E-01	7,233E-04	1,000E+00	2,185E-07	1,000E+00	5,530E-05	9,987E-01	3,927E-03	+
Viennet2	9,809E-01	2,253E-02	9,972E-01	5,491E-03	9,975E-01	2,324E-04	9,973E-01	9,509E-04	9,952E-01	7,998E-03	+
Viennet3	9,980E-01	4,372E-03	9,999E-01	1,760E-03	1,000E+00	1,248E-05	1,000E+00	5,939E-05	9,996E-01	1,643E-03	+
Viennet4	9,868E-01	1,630E-02	9,947E-01	4,593E-03	1,000E+00	7,993E-05	9,988E-01	4,537E-03	9,877E-01	1,217E-02	+
Tamaki	8,937E-01	4,965E-02	1,020E-02	0,000E+00	1,020E-02	1,388E-17	1,020E-02	1,388E-17	1,020E-02	1,976E-07	+

Çizelge Ek-2.4. HV göstergesi için ortanca ($\tilde{x}$) ve çeyrekler açıklığı (*IQR*) değerleri (50000 fonksiyon değerlendirmesi)

Problem	IBMO		OMOPSO		AbYSS		NSGA2		SPEA2		
	$\tilde{x}$	*IQR*	$\tilde{x}$	*IQR*	$\tilde{x}$	*IQR*	$\tilde{x}$	*IQR*	$\tilde{x}$	*IQR*	
Schaffer	8,299E-01	4,796E-05	8,299E-01	4,443E-05	8,299E-01	4,373E-05	7,097E-01	8,992E-02	7,051E-01	7,526E-02	+
Fonseca	3,121E-01	6,856E-05	3,125E-01	2,367E-05	3,110E-01	2,776E-04	3,080E-01	5,364E-04	3,107E-01	2,337E-04	+
Kursawe	4,016E-01	6,182E-05	3,985E-01	4,520E-04	4,014E-01	1,378E-04	3,996E-01	3,046E-04	4,011E-01	1,879E-04	+
ZDT1	6,620E-01	5,394E-05	6,617E-01	1,679E-04	6,620E-01	7,801E-05	6,602E-01	4,030E-04	6,616E-01	1,508E-04	+
ZDT2	3,287E-01	4,366E-05	3,285E-01	1,297E-04	3,287E-01	8,112E-05	3,272E-01	2,817E-04	3,284E-01	1,079E-04	+
ZDT3	5,155E-01	5,493E-04	5,154E-01	5,209E-04	5,160E-01	3,492E-03	5,154E-01	1,253E-04	5,157E-01	9,789E-05	+
ZDT4	6,637E-01	1,138E-03	0,000E+00	0,000E+00	6,596E-01	2,659E-03	6,593E-01	1,397E-03	6,605E-01	1,398E-03	+
ZDT6	4,014E-01	1,850E-05	4,013E-01	6,992E-05	4,007E-01	1,350E-04	3,981E-01	6,831E-04	4,004E-01	2,869E-04	+
Constr	7,744E-01	5,585E-03	7,767E-01	1,990E-04	7,765E-01	1,774E-04	7,742E-01	3,650E-04	7,758E-01	3,537E-04	+
Srinivas	5,409E-01	8,779E-05	5,408E-01	8,297E-05	5,408E-01	1,045E-04	5,380E-01	4,697E-04	5,400E-01	2,159E-04	+
Osycka2	7,095E-01	5,144E-02	7,088E-01	1,386E-02	7,453E-01	3,462E-01	7,525E-01	7,725E-03	7,445E-01	2,284E-02	+
Golinski	9,696E-01	6,661E-05	9,683E-01	1,879E-04	9,695E-01	9,083E-05	9,691E-01	1,937E-04	9,676E-01	4,594E-04	-
Tanaka	3,036E-01	2,831E-03	3,073E-01	4,831E-04	3,083E-01	3,527E-04	3,078E-01	3,185E-04	3,094E-01	2,916E-04	+
Binh2	7,990E-01	3,267E-05	8,008E-01	2,445E-05	8,006E-01	4,604E-05	7,994E-01	2,505E-04	7,999E-01	1,808E-04	+
DTLZ1	7,402E-01	3,828E-02	0,000E+00	0,000E+00	7,952E-01	7,750E-03	7,577E-01	6,764E-03	6,431E-01	2,094E-01	+
DTLZ2	4,372E-01	1,984E-03	4,222E-01	2,164E-03	4,351E-01	2,568E-03	3,738E-01	5,607E-03	4,454E-01	6,933E-04	+
DTLZ3	0,000E+00	2,167E-02	0,000E+00	0,000E+00	2,966E-01	3,829E-01	3,206E-01	5,172E-02	0,000E+00	0,000E+00	+
DTLZ4	4,389E-01	2,794E-03	4,041E-01	2,824E-02	4,364E-01	1,660E-03	3,762E-01	7,384E-03	4,422E-01	7,397E-04	+
DTLZ5	2,119E-02	5,949E-03	9,512E-02	4,373E-05	9,538E-02	7,127E-06	9,232E-02	2,544E-04	9,498E-02	6,099E-05	+
DTLZ6	2,657E-02	0,000E+00	9,536E-02	7,142E-06	5,385E-02	1,942E-02	8,111E-02	1,755E-02	0,000E+00	0,000E+00	+
DTLZ7	3,224E-01	3,648E-03	3,035E-01	5,500E-03	2,724E-01	5,593E-02	2,841E-01	5,172E-03	3,082E-01	1,887E-03	+
Viennet1	7,371E-01	4,748E-04	7,376E-01	4,059E-04	7,363E-01	6,108E-04	7,106E-01	2,796E-03	7,402E-01	1,920E-04	+
Viennet2	9,291E-01	3,132E-04	9,294E-01	2,135E-04	9,286E-01	4,183E-04	9,198E-01	1,713E-03	9,300E-01	7,607E-05	+
Viennet3	8,403E-01	1,211E-04	8,401E-01	7,993E-05	8,399E-01	1,229E-04	8,327E-01	9,393E-04	8,374E-01	2,602E-04	+
Viennet4	8,685E-01	6,997E-04	8,701E-01	4,280E-04	8,682E-01	9,228E-04	8,554E-01	2,908E-03	8,727E-01	2,841E-04	+
Tamaki	3,192E-01	1,508E-02	4,197E-01	1,782E-03	4,184E-01	2,353E-03	3,567E-01	8,989E-03	4,346E-01	7,595E-04	+

EK-3

Geliştirilen Renkli Segmentasyon Yöntemi ile Test Görüntüleri için Alınan Ön-Çalıştırma Sonuçları

Çizelge EK-3.1. 42049 numaralı test görüntüsü için alınan ön-çalıştırma sonuçları

Koloni boyutu =20										
Bölge Sayısı	Limit	Uygun çözüm	LCE	GCE	BCE	Toplam Tutarlılık Hatası	Dağılma (S)	Yoğunluk (Dbw)	S_Dbw	Geçen Süre (s)
	0	1	0,101291	0,101276	0,557316	0,759882	0,179115	1,501568	1,680682	2,486629
	5	1	0,036265	0,019110	0,708394	0,763769	0,229748	4,836968	5,066716	2,215534
6	10	0								2,047909
	15	3	0,103779	0,095889	0,548501	0,748169	0,283217	1,218437	1,501654	2,841490
	20	1	0,090355	0,089656	0,681119	0,861131	0,200472	2,213665	2,414137	2,111410
	0	1	0,059887	0,055402	0,729702	0,844991	0,390555	1,557209	1,947764	2,363065
	5	1	0,023288	0,011644	0,736013	0,770944	0,479973	1,395964	1,875938	2,333651
7	10	0								1,656412
	15	0								1,774000
	20	0								2,113510
	0	0								2,351833
	5	1	0,056733	0,050151	0,698641	0,805525	0,347545	0,292488	0,640033	2,659221
8	10	0								2,561565
	15	0								2,887161
	20	1	0,054956	0,046345	0,721239	0,822539	0,173829	5,590487	5,764316	2,679795
	0	0								2,737004
	5	0								2,423886
9	10	2	0,124774	0,113673	0,411368	0,649815	0,402970	0,039074	0,442044	2,955582
	15	0								2,560662
	20	0								2,486628

Çizelge EK-3.2. 62096 numaralı test görüntüsü için alınan ön-çalıştırma sonuçları

Koloni boyutu =20										
Bölge Sayısı	Limit	Uygun çözüm	LCE	GCE	BCE	Toplam Tutarlılık Hatası	Dağılma (S)	Yoğunluk (Dbw)	S_Dbw	Geçen Süre (s)
	0	1	0,243258	0,036206	0,494302	0,773766	0,115660	0,022355	0,138015	1,505657
	5	1	0,142270	0,062411	0,191733	0,396413	0,041439	0,000000	0,041439	0,919402
2	10	1	0,081279	0,036117	0,231637	0,349032	0,098606	0,006964	0,105570	0,765161
	15	2	0,073709	0,050569	0,197802	0,322079	0,093783	0,022056	0,115839	0,811129
	20	1	0,114875	0,091176	0,194369	0,400420	0,081048	0,013976	0,095024	0,923917
	0	1	0,108554	0,040268	0,277976	0,426798	0,156962	0,047419	0,204382	1,427377
	5	1	0,097450	0,031633	0,244397	0,373480	0,049072	0,000000	0,049072	1,301139
3	10	3	0,066295	0,066286	0,212675	0,345256	0,152287	0,043797	0,196084	1,364822
	15	3	0,056291	0,056284	0,209251	0,321826	0,114051	0,395681	0,509731	1,186165
	20	2	0,140947	0,083855	0,174497	0,399299	0,044336	0,005058	0,049394	2,023922
	0	1	0,097055	0,053177	0,229938	0,380170	0,080110	0,003520	0,083629	1,603045
	5	1	0,154661	0,079115	0,221861	0,455637	0,038479	0,000516	0,038995	1,691011
4	10	2	0,083837	0,037722	0,253074	0,374634	0,088742	0,522814	0,611556	1,740294
	15	3	0,045474	0,041584	0,197312	0,284371	0,109093	0,294449	0,403542	2,119922
	20	3	0,103205	0,086207	0,159803	0,349215	0,171674	0,015789	0,187463	1,678403
	0	1	0,130001	0,094650	0,182376	0,407027	0,184960	0,454077	0,639037	1,720985
	5	0								1,754370
5	10	3	0,064265	0,064256	0,212117	0,340639	0,183099	0,055840	0,238939	2,217255
	15	1	0,140721	0,082425	0,181116	0,404262	0,101890	0,021123	0,123013	1,846725
	20	1	0,104566	0,089805	0,133774	0,328145	0,046887	0,000643	0,047530	1,323540

Çizelge EK-3.3. 86016 numaralı test görüntüsü için alınan ön-çalıştırma sonuçları

Koloni boyutu =20										
Bölge Sayısı	Limit	Uygun çözüm	LCE	GCE	BCE	Toplam Tutarlılık Hatası	Dağılma (S)	Yoğunluk (Dbw)	S_Dbw	Geçen Süre (s)
	0	1	0,045039	0,045039	0,276402	0,366480	0,391288	0,071223	0,462511	1,288846
	5	1	0,042310	0,025329	0,313049	0,380688	0,228637	0,039130	0,267767	1,084859
2	10	3	0,023210	0,023210	0,272929	0,319349	0,290721	0,064128	0,354848	1,299449
	15	1	0,019009	0,019009	0,272277	0,310295	0,175449	0,019504	0,194953	1,309770
	20	1	0,021795	0,020627	0,276324	0,318746	0,298414	0,064788	0,363202	1,168922
	0	1	0,027295	0,023484	0,281502	0,332280	0,167267	0,328240	0,495507	1,385944
	5	2	0,023310	0,023310	0,272916	0,319536	0,240459	0,001230	0,241689	1,613889
3	10	1	0,020388	0,019761	0,275011	0,315160	0,283608	0,061287	0,344895	1,058096
	15	3	0,030793	0,030793	0,273482	0,335068	0,235715	0,136516	0,372232	1,856774
	20	1	0,032480	0,032480	0,273901	0,338861	0,271031	0,019529	0,290560	1,047796
	0	3	0,029160	0,029160	0,273303	0,331622	0,159674	0,433977	0,593651	2,345614
	5	2	0,018696	0,017764	0,274544	0,311004	0,224542	0,007664	0,232205	1,966138
4	10	1	0,032250	0,032250	0,273928	0,338427	0,206985	0,164184	0,371170	1,288659
	15	1	0,023554	0,023554	0,275134	0,322242	0,177624	0,041573	0,219197	1,095861
	20	1	0,071413	0,044293	0,355661	0,471366	0,274786	0,036076	0,310862	0,900068
	0	1	0,032219	0,018068	0,301470	0,351757	0,166919	0,349774	0,516693	1,919263
	5	2	0,029018	0,020371	0,289524	0,338913	0,220262	0,032774	0,253036	1,936276
5	10	0								1,261295
	15	3	0,027554	0,027554	0,273194	0,328302	0,318621	0,073719	0,392340	1,934767
	20	0								1,470175

Çizelge EK-3.4. 113016 numaralı test görüntüsü için alınan ön-çalıştırma sonuçları

Koloni boyutu =20										
Bölge Sayısı	Limit	Uygun çözüm	LCE	GCE	BCE	Toplam Tutarlılık Hatası	Dağılma (S)	Yoğunluk (Dbw)	S_Dbw	Geçen Süre (s)
	0	0								1,440759
	5	1	0,112797	0,082449	0,309231	0,504477	0,336664	0,051615	0,388280	0,874418
2	10	1	0,137113	0,128493	0,271029	0,536636	0,345769	0,061064	0,406833	0,794166
	15	1	0,082413	0,041207	0,368447	0,492067	0,394257	0,084986	0,479244	0,866967
	20	1	0,134655	0,131842	0,256333	0,522829	0,337077	0,059376	0,396454	0,960172
	0	1	0,112524	0,111951	0,209955	0,434430	0,267219	0,038971	0,306190	1,171142
	5	1	0,138741	0,128053	0,279178	0,545972	0,267093	0,030667	0,297759	1,045276
3	10	2	0,122571	0,122045	0,220859	0,465475	0,286551	0,005181	0,291731	1,457098
	15	2	0,122987	0,122460	0,221635	0,467082	0,386076	0,144104	0,530180	1,090121
	20	1	0,140556	0,139734	0,243885	0,524175	0,311451	0,340398	0,651849	0,839812
	0	2	0,146620	0,105125	0,255281	0,507027	0,208717	0,563735	0,772451	2,014346
	5	2	0,090237	0,048089	0,321282	0,459608	0,289740	0,040150	0,329890	1,430964
4	10	1	0,089003	0,045367	0,337637	0,472007	0,360018	0,067923	0,427941	0,907657
	15	3	0,077816	0,045152	0,263126	0,386095	0,248652	0,066451	0,315103	1,830667
	20	0								1,187861
	0	1	0,106629	0,106195	0,203179	0,416003	0,336655	0,105936	0,442591	1,672619
	5	1	0,110871	0,109920	0,204163	0,424953	0,166027	0,014319	0,180346	1,703921
5	10	3	0,137181	0,077682	0,235922	0,450785	0,202065	0,287484	0,489549	2,327449
	15	1	0,130205	0,128389	0,220596	0,479190	0,289348	0,142595	0,431943	1,580364
	20	1	0,134821	0,134369	0,241533	0,510723	0,347348	0,073323	0,420671	1,467215

Çizelge EK-3.5. 135069 numaralı test görüntüsü için alınan ön-çalıştırma sonuçları

Koloni boyutu =20										
Bölge Sayısı	Limit	Uygun çözüm	LCE	GCE	BCE	Toplam Tutarlılık Hatası	Dağılma (S)	Yoğunluk (Dbw)	S_Dbw	Geçen Süre (s)
	0	2	0,015767	0,008057	0,044990	0,068813	0,401769	0,000047	0,401816	5,501613
	5	1	0,039644	0,029692	0,494335	0,563670	0,052227	0,772755	0,824982	5,274936
2	10	1	0,040734	0,030378	0,472792	0,543905	0,048511	0,734882	0,783394	7,631666
	15	2	0,010635	0,008623	0,030312	0,049569	0,230558	0,830842	1,061400	4,087418
	20	0								0,567351
	0	0								1,389411
	5	3	0,011853	0,011150	0,029869	0,052872	0,210109	0,000000	0,210109	12,266390
3	10	2	0,012495	0,007159	0,037154	0,056808	0,262018	0,278969	0,540988	15,414238
	15	2	0,013741	0,007786	0,039514	0,061041	0,254010	0,556993	0,811003	1,580608
	20	2	0,030430	0,018022	0,044336	0,092788	0,051035	1,111352	1,162387	6,423113
	0	1	0,011616	0,007842	0,035024	0,054481	0,406560	1,127016	1,533576	1,751208
	5	2	0,014331	0,014048	0,032289	0,060667	0,139479	0,299670	0,439149	1,629298
4	10	2	0,011636	0,011492	0,029349	0,052477	0,910716	0,338147	1,248863	1,934814
	15	2	0,016151	0,015849	0,016446	0,048446	0,381273	0,613525	0,994799	1,792477
	20	1	0,039723	0,029771	0,543623	0,613117	0,032832	0,512925	0,545757	6,273645
	0	1	0,038395	0,027773	0,493831	0,559999	0,028360	0,735423	0,763783	2,418316
	5	1	0,043422	0,032289	0,669083	0,744794	0,018931	0,499789	0,518719	19,012056
5	10	1	0,043764	0,032652	0,709051	0,785466	0,014648	0,462184	0,476833	16,991266
	15	0								2,406384
	20	0								1,767421

Çizelge EK-3.6. 299091 numaralı test görüntüsü için alınan ön-çalıştırma sonuçları

Koloni boyutu =20										
Bölge Sayısı	Limit	Uygun çözüm	LCE	GCE	BCE	Toplam Tutarlılık Hatası	Dağılma (S)	Yoğunluk (Dbw)	S_Dbw	Geçen Süre (s)
	0	0								1,456267
	5	2	0,071208	0,071025	0,183923	0,326155	0,479555	0,243759	0,723314	1,426204
2	10	1	0,073616	0,046764	0,466014	0,586394	0,518026	0,287475	0,805500	1,306086
	15	0								1,465914
	20	0								0,721013
	0	0								1,215068
	5	1	0,061543	0,053832	0,167094	0,282469	0,415645	0,356061	0,771706	1,176032
3	10	1	0,179574	0,097341	0,521018	0,797933	0,333069	0,538594	0,871663	1,578182
	15	2	0,116769	0,054820	0,242927	0,414515	0,220795	0,047047	0,267843	2,193606
	20	0								1,606990
	0	2	0,181339	0,105329	0,248647	0,535315	0,260138	0,374377	0,634516	1,923301
	5	1	0,182854	0,098717	0,520855	0,802426	0,197160	0,359179	0,556339	2,134949
4	10	2	0,310881	0,189155	0,320844	0,820880	0,191243	0,270542	0,461785	1,938755
	15	0								1,279539
	20	2	0,076367	0,060311	0,178549	0,315227	0,333285	0,216337	0,549622	2,074390
	0	0								2,140679
	5	0								1,788724
5	10	0								1,783861
	15	1	0,076623	0,076175	0,188566	0,341364	0,277280	0,144511	0,421791	2,302576
	20	0								1,928774

EK-4

Algoritmaların 500 Fonksiyon Değerlendirmesi için 10 Bağımsız Çalıştırma Sonunda Aldıkları Segmentasyon Sonuçları

Çizelge EK-4.1. IBMO'nun 42049 numaralı test görüntüsü için aldığı çalıştırma sonuçları

	BÇS	BBS	LCE	GCE	BCE	Toplam Tutarlılık Hatası	Dağılma (S)	Yoğunluk (Dbw)	S_Dbw	Geçen Süre (s)
Çalışma-1	1	4	0,118986	0,113559	0,405716	0,638261	0,148839	1,476732	1,625572	29,193099
Çalışma-2	1	2	0,031288	0,015644	0,731169	0,778101	0,407679	1,473818	1,881497	27,370196
Çalışma-3	2	4	0,124119	0,124026	0,463675	0,711820	0,168731	0,885177	1,053909	27,938191
Çalışma-4	7	4	0,125861	0,120981	0,418056	0,664898	0,137791	3,636158	3,773950	30,997351
Çalışma-5	7	5	0,133836	0,133749	0,324706	0,592290	0,211014	0,639071	0,850085	32,136803
Çalışma-6	1	5	0,130884	0,123938	0,456238	0,711060	0,180954	0,483044	0,663998	30,338495
Çalışma-7	2	5	0,127600	0,126887	0,315882	0,570370	0,204968	0,748724	0,953692	33,277100
Çalışma-8	2	3	0,104302	0,104267	0,553096	0,761664	0,279829	0,380976	0,660805	26,491366
Çalışma-9	2	3	0,099588	0,094456	0,554132	0,748177	0,208453	2,883909	3,092362	28,891926
Çalışma-10	7	3	0,105176	0,100623	0,534943	0,740742	0,191617	1,245091	1,436708	33,030661

Çizelge EK-4.2. IBMO'nun 62069 numaralı test görüntüsü için aldığı çalıştırma sonuçları

	BÇS	BBS	LCE	GCE	BCE	Toplam Tutarlılık Hatası	Dağılma (S)	Yoğunluk (Dbw)	S_Dbw	Geçen Süre (s)
Çalışma-1	1	3	0,053193	0,037387	0,065598	0,156178	0,035851	0,000000	0,035851	18,714446
Çalışma-2	2	5	0,064595	0,051388	0,070996	0,186979	0,031157	0,000983	0,032140	20,257447
Çalışma-3	1	6	0,066929	0,052511	0,091242	0,210682	0,023792	0,000000	0,023792	18,279331
Çalışma-4	4	5	0,069727	0,058173	0,046122	0,174022	0,070067	0,002495	0,072562	18,455866
Çalışma-5	2	3	0,050142	0,041711	0,052910	0,144763	0,080334	0,000000	0,080334	19,334017
Çalışma-6	3	4	0,068767	0,039583	0,069524	0,177875	0,061476	0,000000	0,061476	18,711713
Çalışma-7	2	3	0,055626	0,046687	0,063802	0,166115	0,112068	0,000000	0,112068	18,414553
Çalışma-8	2	3	0,058462	0,046812	0,066487	0,171761	0,059495	0,354868	0,414363	17,779720
Çalışma-9	1	4	0,072210	0,032091	0,076560	0,180862	0,027041	0,000000	0,027041	17,180586
Çalışma-10	1	4	0,070064	0,054962	0,069181	0,194208	0,041328	0,000000	0,041328	18,681561

Çizelge EK-4.3. IBMO'nun 86016 numaralı test görüntüsü için aldığı çalıştırma sonuçları

	BÇS	BBS	LCE	GCE	BCE	Toplam Tutarlılık Hatası	Dağılma (S)	Yoğunluk (Dbw)	S_Dbw	Geçen Süre (s)
Çalışma-1	6	2	0,020326	0,020326	0,273596	0,314249	0,182435	0,502460	0,684896	9,260838
Çalışma-2	1	2	0,033105	0,033105	0,274269	0,340480	0,120442	0,000000	0,120442	9,158403
Çalışma-3	1	3	0,046014	0,046014	0,275499	0,367528	0,117442	0,000000	0,117442	8,239775
Çalışma-4	3	2	0,022307	0,022307	0,273814	0,318428	0,205475	0,000000	0,205475	5,980172
Çalışma-5	3	2	0,027044	0,027044	0,273285	0,327373	0,151263	0,025297	0,176560	8,263891
Çalışma-6	3	2	0,024699	0,024699	0,272824	0,322221	0,262480	0,000000	0,262480	9,043194
Çalışma-7	1	3	0,030893	0,030893	0,280238	0,342025	0,161914	0,000000	0,161914	9,736336
Çalışma-8	4	2	0,025853	0,025853	0,272977	0,324683	0,268993	0,000408	0,269401	9,447518
Çalışma-9	1	2	0,031002	0,031002	0,273573	0,335576	0,176819	0,000000	0,176819	7,092424
Çalışma-10	1	2	0,032471	0,032471	0,274018	0,338960	0,180600	0,000000	0,180600	6,356789

Çizelge EK-4.4. IBMO'nun 113016 numaralı test görüntüsü için aldığı çalıştırma sonuçları

	BÇS	BBS	LCE	GCE	BCE	Toplam Tutarlılık Hatası	Dağılma (S)	Yoğunluk (Dbw)	S_Dbw	Geçen Süre (s)
Çalışma-1	2	2	0,096677	0,096208	0,193806	0,386691	0,150722	0,017281	0,168003	15,025289
Çalışma-2	4	2	0,101973	0,101452	0,198127	0,401552	0,302850	0,060722	0,363572	18,169795
Çalışma-3	3	2	0,098951	0,097504	0,191050	0,387505	0,259048	0,000000	0,259048	18,034021
Çalışma-4	6	2	0,099498	0,077816	0,252874	0,430187	0,248975	0,141923	0,390898	16,323850
Çalışma-5	2	2	0,097697	0,097084	0,192194	0,386974	0,169928	0,005358	0,175286	15,159579
Çalışma-6	5	2	0,090185	0,088903	0,183136	0,362224	0,204651	0,020822	0,225473	17,257824
Çalışma-7	4	2	0,084688	0,044469	0,294435	0,423592	0,229799	0,033268	0,263067	17,522551
Çalışma-8	2	2	0,108781	0,107767	0,200717	0,417264	0,158131	0,051302	0,209433	16,070287
Çalışma-9	3	2	0,101642	0,100925	0,196096	0,398663	0,186184	0,030660	0,216844	17,931938
Çalışma-10	1	2	0,114366	0,113554	0,208365	0,436285	0,201606	0,000000	0,201606	16,066975

Çizelge EK-4.5. IBMO'nun 135069 numaralı test görüntüsü için aldığı çalıştırma sonuçları

	BÇS	BBS	LCE	GCE	BCE	Toplam Tutarlılık Hatası	Dağılma (S)	Yoğunluk (Dbw)	S_Dbw	Geçen Süre (s)
Çalışma-1	5	3	0,015510	0,012603	0,019651	0,047763	0,153215	0,678048	0,831262	17,605874
Çalışma-2	6	2	0,016924	0,008693	0,047703	0,073320	0,211985	0,299680	0,511665	54,048801
Çalışma-3	4	2	0,016989	0,014813	0,036003	0,067805	0,125402	0,202266	0,327669	50,392755
Çalışma-4	2	4	0,023559	0,012467	0,028798	0,064824	0,222594	0,633739	0,856333	17,800219
Çalışma-5	4	2	0,019417	0,009824	0,055057	0,084298	0,322536	0,000000	0,322536	29,966535
Çalışma-6	2	3	0,029931	0,017723	0,043773	0,091428	0,035470	0,375261	0,410731	20,017617
Çalışma-7	3	2	0,016491	0,009887	0,043590	0,069968	0,152128	0,308839	0,460968	21,023431
Çalışma-8	5	2	0,013899	0,013807	0,015201	0,042907	0,354371	0,436605	0,790976	23,437318
Çalışma-9	6	2	0,018049	0,011381	0,045984	0,075415	0,128837	0,345258	0,474095	38,278976
Çalışma-10	4	2	0,010426	0,007356	0,032237	0,050019	0,231889	0,257955	0,489844	22,665061

Çizelge EK-4.6. IBMO'nun 299091 numaralı test görüntüsü için aldığı çalıştırma sonuçları

	BÇS	BBS	LCE	GCE	BCE	Toplam Tutarlılık Hatası	Dağılma (S)	Yoğunluk (Dbw)	S_Dbw	Geçen Süre (s)
Çalışma-1	1	2	0,054041	0,054027	0,170544	0,278612	0,213625	0,093957	0,307582	15,775130
Çalışma-2	1	2	0,059704	0,058978	0,195226	0,313908	0,258517	0,123920	0,382437	14,653810
Çalışma-3	2	2	0,069269	0,068138	0,214182	0,351589	0,251582	0,236361	0,487942	14,681759
Çalışma-4	6	2	0,029681	0,029521	0,158633	0,217835	0,293730	0,141178	0,434908	15,863545
Çalışma-5	5	2	0,018844	0,018827	0,141117	0,178788	0,253886	0,273445	0,527331	15,059429
Çalışma-6	2	2	0,014711	0,014707	0,136737	0,166155	0,214982	0,060953	0,275935	15,233427
Çalışma-7	4	2	0,017710	0,017708	0,138360	0,173778	0,348505	0,001010	0,349515	15,130439
Çalışma-8	4	2	0,033584	0,033427	0,158653	0,225664	0,254365	0,227406	0,481771	14,060967
Çalışma-9	1	2	0,020810	0,020773	0,141799	0,183382	0,215078	0,068799	0,283877	13,856671
Çalışma-10	3	2	0,060281	0,059810	0,180072	0,300163	0,330410	0,028459	0,358869	13,198104

Çizelge EK-4.7. NSPSO'nun 3096 numaralı test görüntüsü için aldığı çalıştırma sonuçları

	BÇS	BBS	LCE	GCE	BCE	Toplam Tutarlılık Hatası	Dağılma (S)	Yoğunluk (Dbw)	S_Dbw	Geçen Süre (s)
Çalışma-1	2	5	0,029444	0,025368	0,269168	0,323980	0,055993	0,000000	0,055993	36,210093
Çalışma-2	4	2	0,033910	0,031660	0,032780	0,098350	0,154790	0,445020	0,599810	22,828942
Çalışma-3	4	2	0,037246	0,035037	0,036331	0,108615	0,311008	0,000000	0,311008	33,095358
Çalışma-4	5	2	0,034044	0,019159	0,066548	0,119752	0,312785	0,348265	0,661050	23,627899
Çalışma-5	3	3	0,039571	0,038813	0,056630	0,135014	0,181702	0,000000	0,181702	20,980360
Çalışma-6	3	2	0,036226	0,021940	0,071530	0,129695	0,639564	0,006338	0,645902	32,036112
Çalışma-7	1	6	0,029436	0,028981	0,246411	0,304827	0,069655	0,022198	0,091853	16,301162
Çalışma-8	6	2	0,033813	0,027203	0,038286	0,099302	1,001267	0,000861	1,002128	16,652776
Çalışma-9	4	2	0,029631	0,025557	0,032188	0,087375	0,250551	0,000000	0,250551	31,713330
Çalışma-10	3	2	0,033197	0,019396	0,057461	0,110055	0,336659	0,130171	0,466830	14,277366

Çizelge EK-4.8. NSPSO'nun 42049 numaralı test görüntüsü için aldığı çalıştırma sonuçları

	BÇS	BBS	LCE	GCE	BCE	Toplam Tutarlılık Hatası	Dağılma (S)	Yoğunluk (Dbw)	S_Dbw	Geçen Süre (s)
Çalışma-1	10	3	0,099438	0,099363	0,552245	0,751046	0,248112	0,853469	1,101581	31,580638
Çalışma-2	5	3	0,103536	0,103517	0,559365	0,766418	0,247461	0,435818	0,683279	26,846159
Çalışma-3	4	4	0,129081	0,123707	0,416388	0,669176	0,193324	0,639812	0,833136	30,056306
Çalışma-4	4	3	0,106486	0,106442	0,552205	0,765133	0,375936	0,011520	0,387456	30,849395
Çalışma-5	7	3	0,102260	0,102211	0,554593	0,759065	0,166476	4,154437	4,320913	31,200240
Çalışma-6	7	3	0,100720	0,100705	0,558368	0,759793	0,163522	2,067152	2,230674	30,930983
Çalışma-7	9	3	0,098055	0,097990	0,549221	0,745267	0,307896	0,129123	0,437019	31,962273
Çalışma-8	7	3	0,104950	0,104918	0,554357	0,764225	0,292577	0,439067	0,731643	30,112334
Çalışma-9	8	4	0,109684	0,084183	0,550675	0,744542	0,381970	0,034982	0,416952	32,387618
Çalışma-10	7	3	0,101720	0,101703	0,556406	0,759829	0,191001	1,405297	1,596298	33,321161

Çizelge EK-4.9. NSPSO'nun 62096 numaralı test görüntüsü için aldığı çalıştırma sonuçları

	BÇS	BBS	LCE	GCE	BCE	Toplam Tutarlılık Hatası	Dağılma (S)	Yoğunluk (Dbw)	S_Dbw	Geçen Süre (s)
Çalışma-1	3	5	0,070240	0,057343	0,066479	0,194062	0,019585	0,800069	0,819653	21,308288
Çalışma-2	1	4	0,068731	0,048576	0,061188	0,178495	0,022984	0,000000	0,022984	21,884963
Çalışma-3	1	5	0,073863	0,052672	0,073747	0,200282	0,023237	0,000000	0,023237	20,182012
Çalışma-4	1	4	0,084995	0,063217	0,084994	0,233206	0,027713	0,000000	0,027713	19,537472
Çalışma-5	1	5	0,070569	0,058938	0,080824	0,210331	0,025292	0,000000	0,025292	19,947283
Çalışma-6	1	4	0,074979	0,061738	0,079994	0,216710	0,022861	0,000000	0,022861	19,469592
Çalışma-7	1	5	0,072499	0,059484	0,069998	0,201980	0,035412	0,000000	0,035412	18,855475
Çalışma-8	1	5	0,068443	0,056557	0,075549	0,200548	0,024140	0,000000	0,024140	18,741090
Çalışma-9	1	4	0,080748	0,051156	0,076192	0,208096	0,025888	0,000000	0,025888	20,244250
Çalışma-10	1	6	0,061290	0,050224	0,080676	0,192190	0,022412	0,000000	0,022412	20,948644

Çizelge EK-4.10. NSPSO'nun 86016 numaralı test görüntüsü için aldığı çalıştırma sonuçları

	BÇS	BBS	LCE	GCE	BCE	Toplam Tutarlılık Hatası	Dağılma (S)	Yoğunluk (Dbw)	S_Dbw	Geçen Süre (s)
Çalışma-1	4	2	0,026603	0,026603	0,274961	0,328168	0,221667	0,000151	0,221818	9,452543
Çalışma-2	1	4	0,112222	0,059713	0,326525	0,498460	0,106910	0,000000	0,106910	9,013720
Çalışma-3	2	2	0,031531	0,031531	0,273689	0,336751	0,167249	0,000000	0,167249	8,824278
Çalışma-4	1	2	0,035585	0,035585	0,275603	0,346773	0,134526	0,000000	0,134526	8,557171
Çalışma-5	1	3	0,044504	0,044504	0,275356	0,364363	0,109093	0,000000	0,109093	9,262848
Çalışma-6	1	3	0,042232	0,042232	0,275272	0,359737	0,116840	0,000000	0,116840	9,278984
Çalışma-7	1	3	0,033163	0,033163	0,280038	0,346363	0,139198	0,000000	0,139198	8,276341
Çalışma-8	1	2	0,033804	0,033804	0,274239	0,341847	0,139623	0,000000	0,139623	9,184500
Çalışma-9	6	2	0,023244	0,023244	0,272545	0,319032	0,209643	0,002889	0,212532	9,852829
Çalışma-10	1	2	0,027988	0,027988	0,274127	0,330103	0,136322	0,000000	0,136322	8,989015

Çizelge EK-4.11. NSPSO'nun 113016 numaralı test görüntüsü için aldığı çalıştırma sonuçları

	BÇS	BBS	LCE	GCE	BCE	Toplam Tutarlılık Hatası	Dağılma (S)	Yoğunluk (Dbw)	S_Dbw	Geçen Süre (s)
Çalışma-1	5	2	0,102579	0,102001	0,198027	0,402608	0,140792	0,034231	0,175023	18,141133
Çalışma-2	3	2	0,117152	0,116539	0,213470	0,447161	0,250977	0,000000	0,250977	16,678674
Çalışma-3	5	2	0,096012	0,095509	0,191842	0,383363	0,192109	0,020362	0,212472	15,285761
Çalışma-4	3	2	0,109518	0,108925	0,205974	0,424416	0,209175	0,000000	0,209175	18,376434
Çalışma-5	7	2	0,106027	0,104485	0,197290	0,407802	0,221740	0,000526	0,222267	17,966731
Çalışma-6	3	2	0,100092	0,098955	0,192048	0,391095	0,156788	0,036851	0,193639	16,821510
Çalışma-7	4	2	0,115897	0,114908	0,209236	0,440041	0,200913	0,000415	0,201329	13,149561
Çalışma-8	3	2	0,109234	0,108676	0,205908	0,423818	0,211249	0,007148	0,218396	18,427585
Çalışma-9	3	2	0,110589	0,110119	0,207839	0,428547	0,181447	0,088853	0,270300	16,088813
Çalışma-10	2	2	0,104368	0,102860	0,195660	0,402888	0,185254	0,000000	0,185254	17,074960

Çizelge EK-4.12. NSPSO'nun 135069 numaralı test görüntüsü için aldığı çalıştırma sonuçları

	BÇS	BBS	LCE	GCE	BCE	Toplam Tutarlılık Hatası	Dağılma (S)	Yoğunluk (Dbw)	S_Dbw	Geçen Süre (s)
Çalışma-1	7	2	0,017892	0,017518	0,034647	0,070057	0,188185	0,000000	0,188185	49,328700
Çalışma-2	7	2	0,015620	0,007868	0,044558	0,068045	0,297029	0,000000	0,297029	65,027874
Çalışma-3	5	2	0,010745	0,009592	0,029849	0,050186	0,258097	0,000000	0,258097	51,923066
Çalışma-4	4	2	0,016071	0,015832	0,033324	0,065226	0,075263	0,280108	0,355370	26,366686
Çalışma-5	5	2	0,012721	0,006874	0,038120	0,057715	0,280981	0,000000	0,280981	44,095345
Çalışma-6	6	2	0,015253	0,015020	0,032288	0,062561	0,076109	0,284342	0,360450	55,359046
Çalışma-7	4	3	0,033886	0,023227	0,044303	0,101416	0,093483	0,000000	0,093483	46,262757
Çalışma-8	6	3	0,028217	0,020514	0,035171	0,083901	0,118158	0,197830	0,315988	49,725162
Çalışma-9	6	2	0,014873	0,007552	0,042826	0,065252	0,133817	0,244450	0,378268	37,704871
Çalışma-10	5	2	0,011164	0,009910	0,030481	0,051555	0,114239	0,288020	0,402259	50,111092

Çizelge EK-4.13. NSPSO'nun 299091 numaralı test görüntüsü için aldığı çalıştırma sonuçları

	BÇS	BBS	LCE	GCE	BCE	Toplam Tutarlılık Hatası	Dağılma (S)	Yoğunluk (Dbw)	S_Dbw	Geçen Süre (s)
Çalışma-1	3	2	0,069403	0,068217	0,215987	0,353608	0,200408	0,139390	0,339798	16,357721
Çalışma-2	1	3	0,147220	0,081800	0,258229	0,487249	0,184340	0,010695	0,195035	15,198598
Çalışma-3	3	3	0,147484	0,085839	0,248451	0,481774	0,203423	0,208695	0,412117	17,203302
Çalışma-4	6	2	0,052334	0,052324	0,171938	0,276596	0,235032	0,153166	0,388198	17,004937
Çalışma-5	1	5	0,180646	0,105619	0,300823	0,587088	0,204320	0,128112	0,332432	15,874911
Çalışma-6	4	2	0,037203	0,018601	0,473866	0,529670	0,330857	0,103828	0,434685	17,846787
Çalışma-7	6	3	0,081181	0,061077	0,196908	0,339166	0,180994	0,181105	0,362099	16,919657
Çalışma-8	5	2	0,015828	0,015819	0,139554	0,171201	0,402444	0,002326	0,404770	16,336937
Çalışma-9	2	2	0,071748	0,070597	0,215520	0,357865	0,193194	0,026165	0,219359	16,857638
Çalışma-10	9	2	0,021894	0,021831	0,147096	0,190821	0,219606	0,072131	0,291737	16,413927

Çizelge EK-4.14. NSGA2'nin 3096 numaralı test görüntüsü için aldığı çalıştırma sonuçları

	BÇS	BBS	LCE	GCE	BCE	Toplam Tutarlılık Hatası	Dağılma (S)	Yoğunluk (Dbw)	S_Dbw	Geçen Süre (s)
Çalışma-1	2	3	0,032267	0,031382	0,044001	0,107650	0,017324	0,252256	0,269581	7,951744
Çalışma-2	0									6,372367
Çalışma-3	3	16	0,039428	0,037970	0,655044	0,732441	0,029196	0,160165	0,189361	300,704154
Çalışma-4	5	2	0,037648	0,033785	0,039855	0,111288	0,046315	0,000000	0,046315	308,061556
Çalışma-5	2	14	0,035807	0,035040	0,479096	0,549943	0,027885	0,000000	0,027885	163,788719
Çalışma-6	7	23	0,048092	0,047385	0,778604	0,874080	0,006997	0,000000	0,006997	873,238495
Çalışma-7	0									5,735782
Çalışma-8	8	5	0,037478	0,034632	0,056723	0,128833	0,014857	0,650303	0,665160	305,014628
Çalışma-9	0									5,730103
Çalışma-10	4	10	0,030151	0,029596	0,306937	0,366683	0,034701	0,047138	0,081839	271,281240

Çizelge EK-4.15. NSGA2'nin 42049 numaralı test görüntüsü için aldığı çalıştırma sonuçları

	BÇS	BBS	LCE	GCE	BCE	Toplam Tutarlılık Hatası	Dağılma (S)	Yoğunluk (Dbw)	S_Dbw	Geçen Süre (s)
Çalışma-1	3	16	0,179718	0,162379	0,458188	0,800285	0,000134	0,000000	0,000134	87,442873
Çalışma-2	1	7	0,392460	0,190384	0,729428	1,312272	0,000098	0,000000	0,000098	13,619936
Çalışma-3	1	14	0,172474	0,156514	0,321513	0,650501	0,000088	0,000000	0,000088	62,640613
Çalışma-4	1	9	0,301857	0,144175	0,568062	1,014094	0,000102	0,000000	0,000102	14,602475
Çalışma-5	4	9	0,388119	0,181272	0,778288	1,347678	0,000099	0,000000	0,000099	14,610491
Çalışma-6	1	12	0,210794	0,166955	0,370936	0,748685	0,000098	0,000000	0,000098	50,570022
Çalışma-7	1	17	0,170513	0,156168	0,406036	0,732718	0,000125	0,000000	0,000125	64,286374
Çalışma-8	1	12	0,180821	0,163823	0,291493	0,636137	0,000129	0,000000	0,000129	76,098348
Çalışma-9	1	14	0,173279	0,158452	0,450212	0,781943	0,000132	0,000000	0,000132	65,672262
Çalışma-10	3	13	0,179783	0,161220	0,460288	0,801291	0,000118	0,000000	0,000118	142,917059

Çizelge EK-4.16. NSGA2'nin 62096 numaralı test görüntüsü için aldığı çalıştırma sonuçları

	BÇS	BBS	LCE	GCE	BCE	Toplam Tutarlılık Hatası	Dağılma (S)	Yoğunluk (Dbw)	S_Dbw	Geçen Süre (s)
Çalışma-1	1	15	0,066480	0,054846	0,484073	0,605399	0,000140	0,000000	0,000140	52,729425
Çalışma-2	2	14	0,066750	0,055089	0,411597	0,533436	0,000162	0,000000	0,000162	96,697289
Çalışma-3	2	14	0,066750	0,055089	0,413107	0,534946	0,000174	0,000000	0,000174	97,260456
Çalışma-4	1	13	0,066728	0,055035	0,431463	0,553225	0,000197	0,000000	0,000197	51,592906
Çalışma-5	1	15	0,066576	0,054917	0,442338	0,563831	0,000163	0,000000	0,000163	53,483025
Çalışma-6	1	15	0,066578	0,054944	0,469879	0,591402	0,000152	0,000000	0,000152	53,335168
Çalışma-7	1	14	0,066750	0,055088	0,415768	0,537606	0,000095	0,000000	0,000095	53,477310
Çalışma-8	1	13	0,067914	0,054074	0,384713	0,506700	0,000119	0,000000	0,000119	50,745152
Çalışma-9	1	14	0,066750	0,055089	0,412976	0,534815	0,000198	0,000000	0,000198	52,872313
Çalışma-10	1	14	0,069233	0,057593	0,471336	0,598163	0,000225	0,000000	0,000225	62,022219

Çizelge EK-4.17. NSGA2'nin 86016 numaralı test görüntüsü için aldığı çalıştırma sonuçları

	BÇS	BBS	LCE	GCE	BCE	Toplam Tutarlılık Hatası	Dağılma (S)	Yoğunluk (Dbw)	S_Dbw	Geçen Süre (s)
Çalışma-1	1	11	0,277371	0,198633	0,474068	0,950072	0,017760	0,000000	0,017760	50,015898
Çalışma-2	7	3	0,044152	0,036526	0,278389	0,359067	0,087681	0,000000	0,087681	363,129981
Çalışma-3	4	12	0,317319	0,221779	0,528459	1,067556	0,035297	0,036232	0,071529	154,055806
Çalışma-4	4	13	0,364377	0,270071	0,540090	1,174537	0,023531	0,011161	0,034692	438,676899
Çalışma-5	3	2	0,029167	0,029167	0,273544	0,331877	0,123098	0,000000	0,123098	33,468635
Çalışma-6	1	17	0,402133	0,220020	0,595470	1,217623	0,017062	0,000000	0,017062	335,847878
Çalışma-7	3	11	0,306101	0,191976	0,441781	0,939859	0,044557	0,000000	0,044557	127,751019
Çalışma-8	5	11	0,282991	0,203486	0,486264	0,972741	0,032809	0,095336	0,128144	217,384484
Çalışma-9	1	11	0,383709	0,304538	0,538309	1,226556	0,017529	0,000000	0,017529	74,873611
Çalışma-10	1	12	0,453407	0,299516	0,597347	1,350270	0,014008	0,000000	0,014008	572,606377

Çizelge EK-4.18. NSGA2'nin 113016 numaralı test görüntüsü için aldığı çalıştırma sonuçları

	BÇS	BBS	LCE	GCE	BCE	Toplam Tutarlılık Hatası	Dağılma (S)	Yoğunluk (Dbw)	S_Dbw	Geçen Süre (s)
Çalışma-1	1	34	0,114900	0,114168	0,567702	0,796771	0,015140	0,000000	0,015140	770,847366
Çalışma-2	1	3	0,152569	0,115728	0,291876	0,560174	0,012346	0,000000	0,012346	7,806258
Çalışma-3	2	19	0,124475	0,123568	0,474172	0,722215	0,009824	0,249231	0,259055	1379,662558
Çalışma-4	1	39	0,124215	0,123709	0,690564	0,938488	0,011172	0,000000	0,011172	1101,488731
Çalışma-5	2	19	0,127074	0,125885	0,451641	0,704601	0,012413	0,508454	0,520867	789,943794
Çalışma-6	3	29	0,129681	0,129107	0,576890	0,835677	0,011518	0,417695	0,429214	3472,692002
Çalışma-7	1	44	0,126738	0,125934	0,681516	0,934188	0,009691	0,000000	0,009691	2188,072449
Çalışma-8	1	35	0,112984	0,112429	0,611007	0,836420	0,012927	0,000000	0,012927	670,391248
Çalışma-9	1	44	0,124046	0,123589	0,646435	0,894071	0,012252	0,000000	0,012252	1423,798154
Çalışma-10	1	35	0,124434	0,124209	0,620357	0,869001	0,013990	0,000000	0,013990	1054,803426

Çizelge EK-4.19. NSGA2'nin 135069 numaralı test görüntüsü için aldığı çalıştırma sonuçları

	BÇS	BBS	LCE	GCE	BCE	Toplam Tutarlılık Hatası	Dağılma (S)	Yoğunluk (Dbw)	S_Dbw	Geçen Süre (s)
Çalışma-1	1	24	0,048704	0,041072	0,922063	1,011839	0,001253	0,000000	0,001253	196,202670
Çalışma-2	2	20	0,045959	0,034719	0,904417	0,985096	0,001528	0,086957	0,088485	291,390420
Çalışma-3	1	23	0,049299	0,037292	0,920664	1,007256	0,001031	0,000000	0,001031	237,792037
Çalışma-4	1	27	0,048338	0,041174	0,921675	1,011187	0,000760	0,000000	0,000760	354,895179
Çalışma-5	8	20	0,050535	0,038166	0,912960	1,001660	0,001510	0,000000	0,001510	393,869717
Çalışma-6	1	23	0,051099	0,038457	0,916239	1,005794	0,001093	0,000000	0,001093	208,620427
Çalışma-7	1	20	0,050333	0,038115	0,909554	0,998003	0,001117	0,000000	0,001117	222,229006
Çalışma-8	2	22	0,049516	0,041827	0,912719	1,004062	0,000992	0,000000	0,000992	446,091615
Çalışma-9	2	18	0,050848	0,038165	0,880366	0,969380	0,001322	0,729614	0,730936	314,049001
Çalışma-10	1	24	0,048504	0,040926	0,919727	1,009157	0,001322	0,000000	0,001322	200,356934

Çizelge EK-4.20. NSGA2'nin 299091 numaralı test görüntüsü için aldığı çalıştırma sonuçları

	BÇS	BBS	LCE	GCE	BCE	Toplam Tutarlılık Hatası	Dağılma (S)	Yoğunluk (Dbw)	S_Dbw	Geçen Süre (s)
Çalışma-1	3	25	0,076781	0,076781	0,657161	0,810724	0,010983	0,016949	0,027932	5485,131879
Çalışma-2	3	18	0,076569	0,076569	0,609592	0,762729	0,012849	0,000000	0,012849	4734,349180
Çalışma-3	1	24	0,091411	0,091411	0,679580	0,862402	0,010514	0,000000	0,010514	4157,920853
Çalışma-4	1	27	0,077602	0,077602	0,731775	0,886979	0,007183	0,000000	0,007183	4202,686952
Çalışma-5	1	36	0,082675	0,082675	0,777061	0,942411	0,003585	0,000000	0,003585	4302,060824
Çalışma-6	3	28	0,073527	0,073527	0,696439	0,843493	0,006651	0,000000	0,006651	4885,905200
Çalışma-7	1	21	0,090433	0,090433	0,619515	0,800381	0,007658	0,000000	0,007658	3819,817994
Çalışma-8	1	26	0,076651	0,076651	0,660287	0,813588	0,015562	0,000000	0,015562	4363,722477
Çalışma-9	1	25	0,082254	0,082254	0,666809	0,831317	0,009491	0,000000	0,009491	4241,481845
Çalışma-10	1	29	0,079669	0,079669	0,736155	0,895493	0,008903	0,000000	0,008903	4373,795353

Çizelge EK-4.21. FCM'nin 3096 numaralı test görüntüsü için aldığı çalıştırma sonuçları

	LCE	GCE	BCE	Toplam Tutarlılık Hatası	Geçen Süre (s)
Çalışma-1	0,035146	0,031644	0,034811	0,101600	4,297040
Çalışma-2	0,035146	0,031644	0,034811	0,101600	4,263865
Çalışma-3	0,035146	0,031644	0,034811	0,101600	4,228908
Çalışma-4	0,035146	0,031644	0,034811	0,101600	4,201756
Çalışma-5	0,035146	0,031644	0,034811	0,101600	4,215818
Çalışma-6	0,035146	0,031644	0,034811	0,101600	4,221693
Çalışma-7	0,035146	0,031644	0,034811	0,101600	4,252115
Çalışma-8	0,035146	0,031644	0,034811	0,101600	4,203542
Çalışma-9	0,035146	0,031644	0,034811	0,101600	4,251715
Çalışma-10	0,035146	0,031644	0,034811	0,101600	4,566745

Çizelge EK-4.22. FCM'nin 42049 numaralı test görüntüsü için aldığı çalıştırma sonuçları

	LCE	GCE	BCE	Toplam Tutarlılık Hatası	Geçen Süre (s)
Çalışma-1	0,612882	0,531385	0,697818	1,842084	18,578802
Çalışma-2	0,612882	0,531385	0,697818	1,842084	18,530603
Çalışma-3	0,602686	0,518241	0,680669	1,801596	18,695250
Çalışma-4	0,602686	0,518241	0,680669	1,801596	18,613239
Çalışma-5	0,612882	0,531385	0,697818	1,842084	18,682518
Çalışma-6	0,602686	0,518241	0,680669	1,801596	18,837946
Çalışma-7	0,616841	0,525923	0,701108	1,843872	18,500513
Çalışma-8	0,377183	0,338840	0,632877	1,348900	18,689864
Çalışma-9	0,616841	0,525923	0,701108	1,843872	18,457013
Çalışma-10	0,612882	0,531385	0,697818	1,842084	18,765718

Çizelge EK-4.23. FCM'nin 62096 numaralı test görüntüsü için aldığı çalıştırma sonuçları

	LCE	GCE	BCE	Toplam Tutarlılık Hatası	Geçen Süre (s)
Çalışma-1	0,102107	0,094006	0,264986	0,461098	10,385511
Çalışma-2	0,102107	0,094006	0,264986	0,461098	10,672484
Çalışma-3	0,102107	0,094006	0,264986	0,461098	10,277046
Çalışma-4	0,102107	0,094006	0,264986	0,461098	10,314269
Çalışma-5	0,102107	0,094006	0,264986	0,461098	10,325453
Çalışma-6	0,102107	0,094006	0,264986	0,461098	10,293728
Çalışma-7	0,102107	0,094006	0,264986	0,461098	11,944729
Çalışma-8	0,102107	0,094006	0,264986	0,461098	12,212585
Çalışma-9	0,102107	0,094006	0,264986	0,461098	10,809510
Çalışma-10	0,102107	0,094006	0,264986	0,461098	10,363689

Çizelge EK-4.24. FCM'nin 86016 numaralı test görüntüsü için aldığı çalıştırma sonuçları

	LCE	GCE	BCE	Toplam Tutarlılık Hatası	Geçen Süre (s)
Çalışma-1	0,048687	0,048686	0,296424	0,393797	4,271056
Çalışma-2	0,048687	0,048686	0,296424	0,393797	4,395971
Çalışma-3	0,048687	0,048686	0,296424	0,393797	4,249038
Çalışma-4	0,048687	0,048686	0,296424	0,393797	4,244777
Çalışma-5	0,048687	0,048686	0,296424	0,393797	4,240383
Çalışma-6	0,048687	0,048686	0,296424	0,393797	4,220029
Çalışma-7	0,048687	0,048686	0,296424	0,393797	4,244118
Çalışma-8	0,048687	0,048686	0,296424	0,393797	4,280665
Çalışma-9	0,048687	0,048686	0,296424	0,393797	4,217866
Çalışma-10	0,048687	0,048686	0,296424	0,393797	4,211782

Çizelge EK-4.25. FCM'nin 113016 numaralı test görüntüsü için aldığı çalıştırma sonuçları

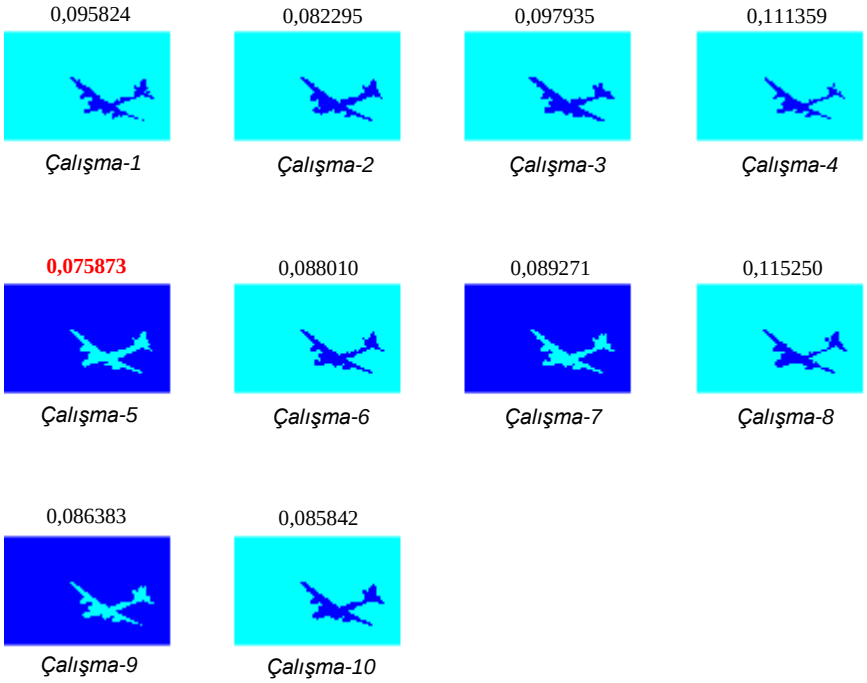
	LCE	GCE	BCE	Toplam Tutarlılık Hatası	Geçen Süre (s)
Çalışma-1	0,192610	0,183654	0,519677	0,895942	10,231946
Çalışma-2	0,192610	0,183654	0,519677	0,895942	10,623676
Çalışma-3	0,192610	0,183654	0,519677	0,895942	10,225470
Çalışma-4	0,192610	0,183654	0,519677	0,895942	10,330819
Çalışma-5	0,192610	0,183654	0,519677	0,895942	10,252631
Çalışma-6	0,192610	0,183654	0,519677	0,895942	10,627003
Çalışma-7	0,192610	0,183654	0,519677	0,895942	10,598107
Çalışma-8	0,192610	0,183654	0,519677	0,895942	10,564418
Çalışma-9	0,192610	0,183654	0,519677	0,895942	10,436398
Çalışma-10	0,192610	0,183654	0,519677	0,895942	10,421566

Çizelge EK-4.26. FCM'nin 135069 numaralı test görüntüsü için aldığı çalıştırma sonuçları

	LCE	GCE	BCE	Toplam Tutarlılık Hatası	Geçen Süre (s)
Çalışma-1	0,032415	0,020859	0,637633	0,690908	8,269594
Çalışma-2	0,032415	0,020859	0,637633	0,690908	8,237039
Çalışma-3	0,032415	0,020859	0,637633	0,690908	8,216117
Çalışma-4	0,032415	0,020859	0,637633	0,690908	8,182712
Çalışma-5	0,032415	0,020859	0,637633	0,690908	8,345273
Çalışma-6	0,032415	0,020859	0,637633	0,690908	8,207136
Çalışma-7	0,032415	0,020859	0,637633	0,690908	8,206287
Çalışma-8	0,032415	0,020859	0,637633	0,690908	8,356067
Çalışma-9	0,032415	0,020859	0,637633	0,690908	8,300011
Çalışma-10	0,032415	0,020859	0,637633	0,690908	8,204184

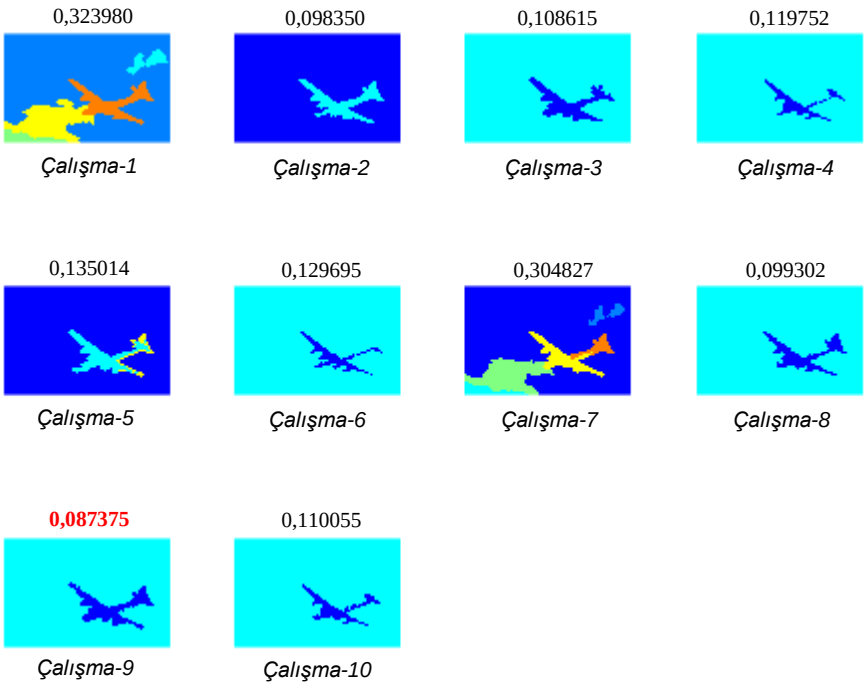
Çizelge EK-4.27. FCM'nin 299091 numaralı test görüntüsü için aldığı çalıştırma sonuçları

	LCE	GCE	BCE	Toplam Tutarlılık Hatası	Geçen Süre (s)
Çalışma-1	0,292942	0,140141	0,418282	0,851365	6,379537
Çalışma-2	0,292942	0,140141	0,418282	0,851365	6,355882
Çalışma-3	0,292942	0,140141	0,418282	0,851365	6,344418
Çalışma-4	0,292942	0,140141	0,418282	0,851365	6,348445
Çalışma-5	0,292942	0,140141	0,418282	0,851365	6,317442
Çalışma-6	0,292942	0,140141	0,418282	0,851365	6,317004
Çalışma-7	0,292942	0,140141	0,418282	0,851365	6,272489
Çalışma-8	0,292942	0,140141	0,418282	0,851365	6,406578
Çalışma-9	0,292942	0,140141	0,418282	0,851365	6,378309
Çalışma-10	0,292942	0,140141	0,418282	0,851365	6,377808

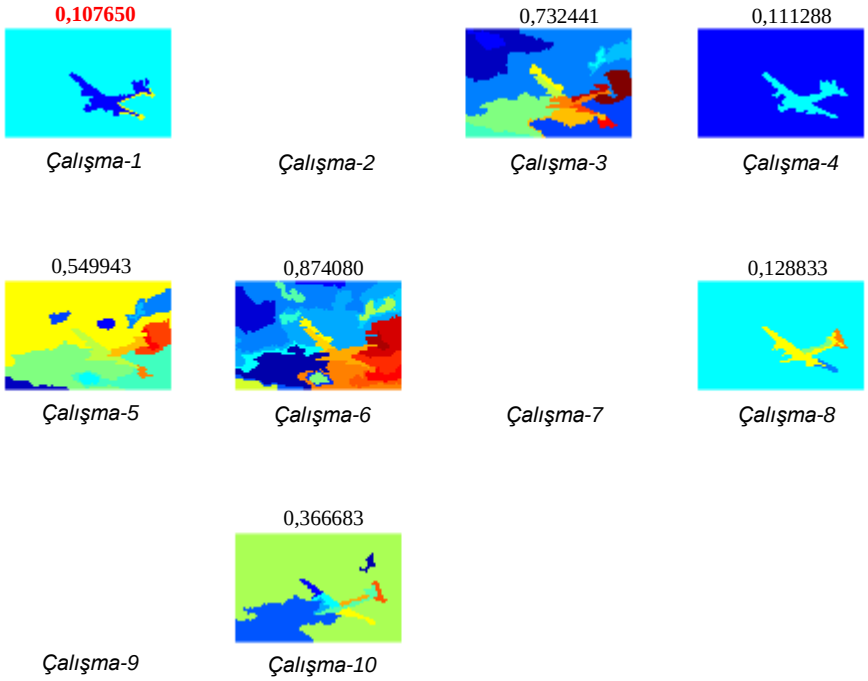


Şekil Ek-4.1. 3096 numaralı test görüntüsü için 500 FEs sonunda IBMO ile alınan 10 çalıştırma sonucu

Segentasyonların toplam tutarlılık hataları görüntülerin üzerinde verilmiştir.



Şekil Ek-4.2. 3096 numaralı test görüntüsü için 500 FEs sonunda NSPSO ile alınan 10 çalıştırma sonucu

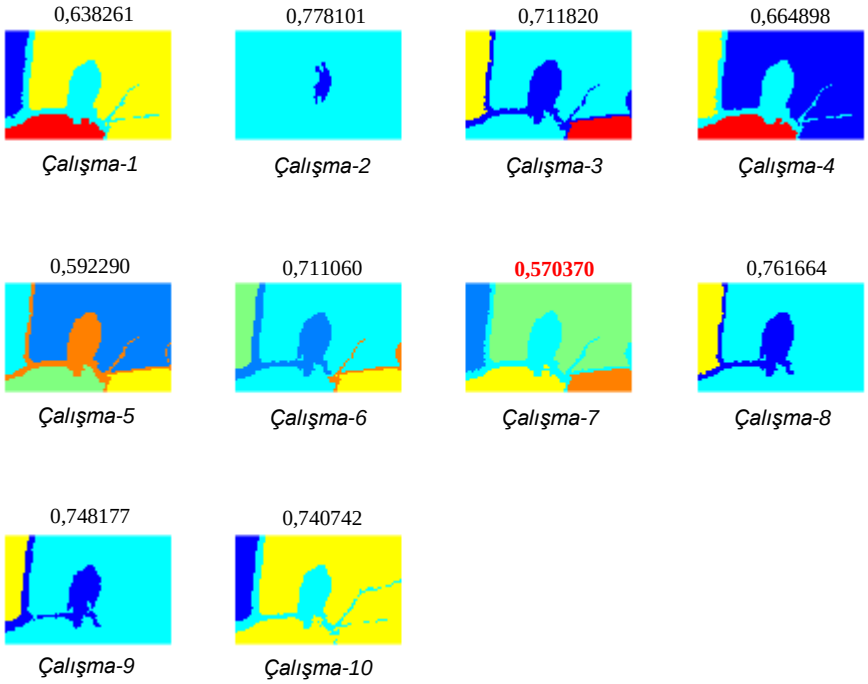


Şekil Ek-4.3. 3096 numaralı test görüntüsü için 500 *FEs* sonunda NSGA2 ile alınan 10 çalıştırma sonucu

Segentasyonların toplam tutarlılık hataları görüntülerin üzerinde verilmiştir.

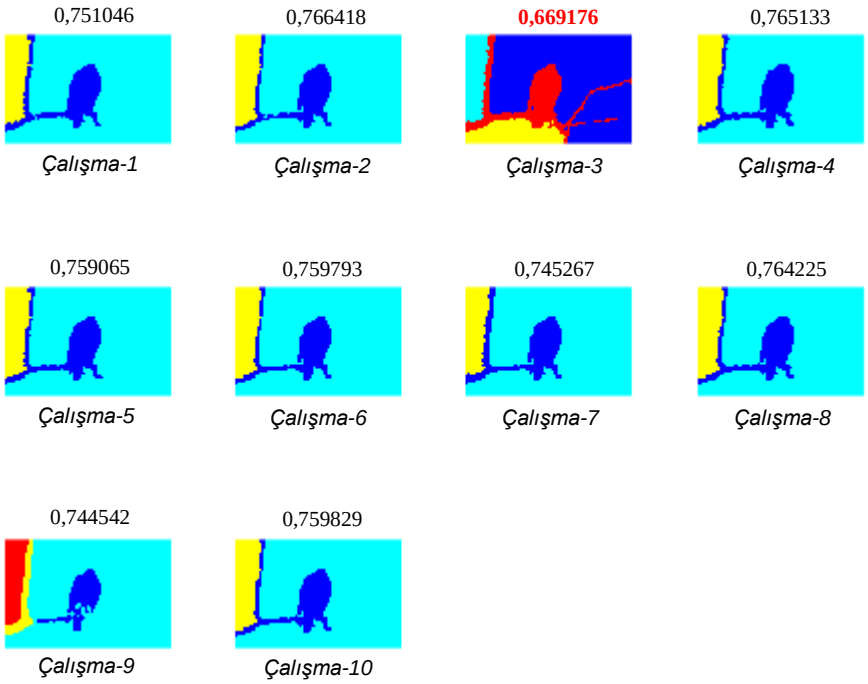


Şekil Ek-4.4. 3096 numaralı test görüntüsü için 500 *FEs* sonunda FCM ile alınan 10 çalıştırma sonucu

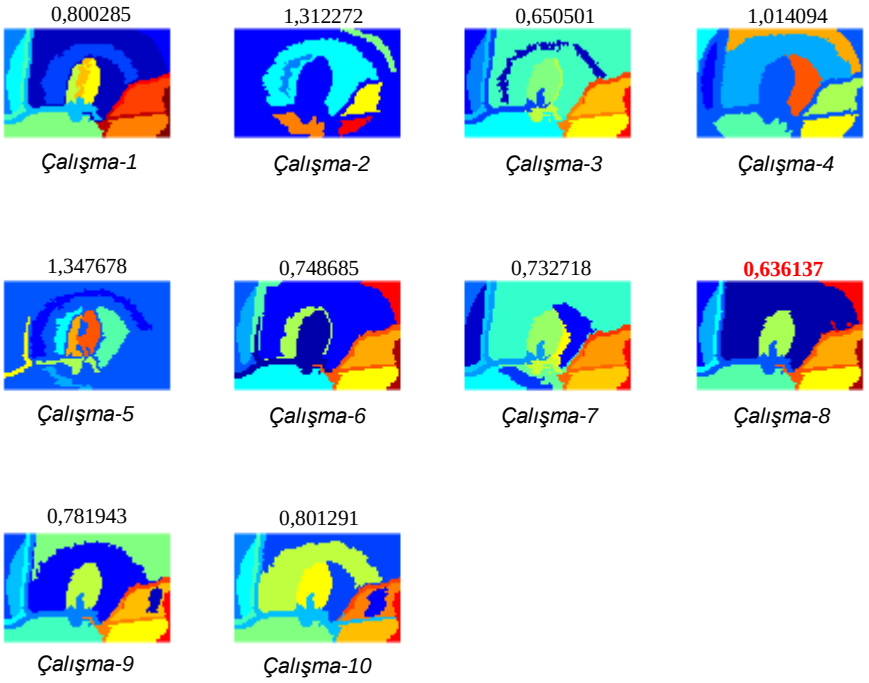


Şekil Ek-4.5. 42049 numaralı test görüntüsü için 500 FEs sonunda IBMO ile alınan 10 çalıştırma sonucu

Segentasyonların toplam tutarlılık hataları görüntülerin üzerinde verilmiştir.

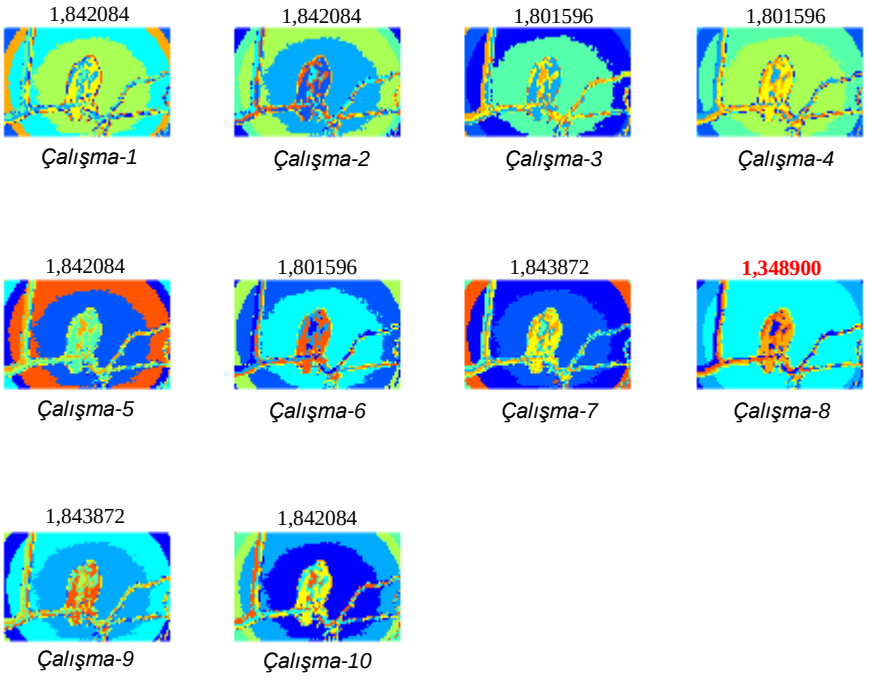


Şekil Ek-4.6. 42049 numaralı test görüntüsü için 500 FEs sonunda NSPSO ile alınan 10 çalıştırma sonucu

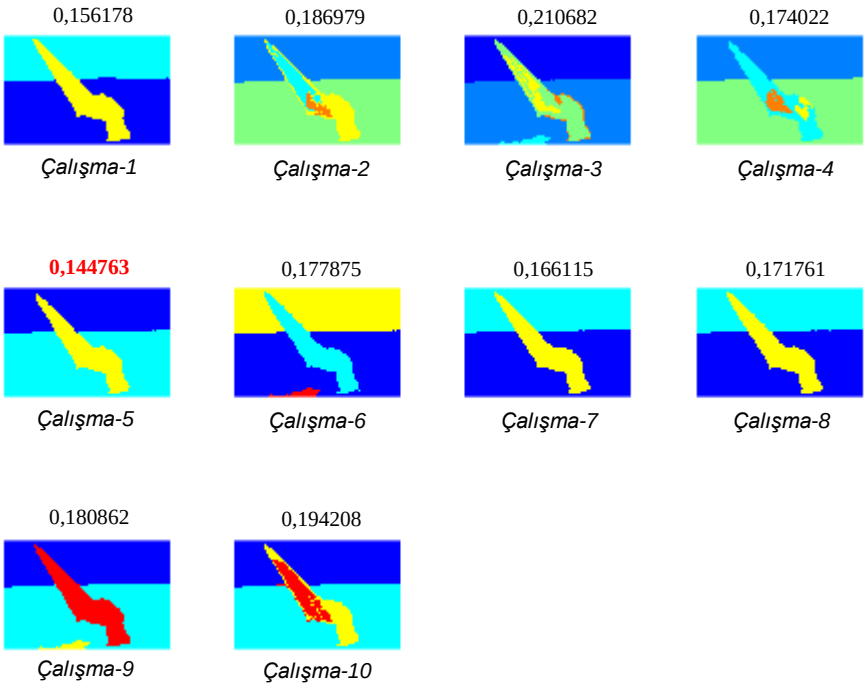


Şekil Ek-4.7. 42049 numaralı test görüntüsü için 500 *FEs* sonunda NSGA2 ile alınan 10 çalıştırma sonucu

Segentasyonların toplam tutarlılık hataları görüntülerin üzerinde verilmiştir.

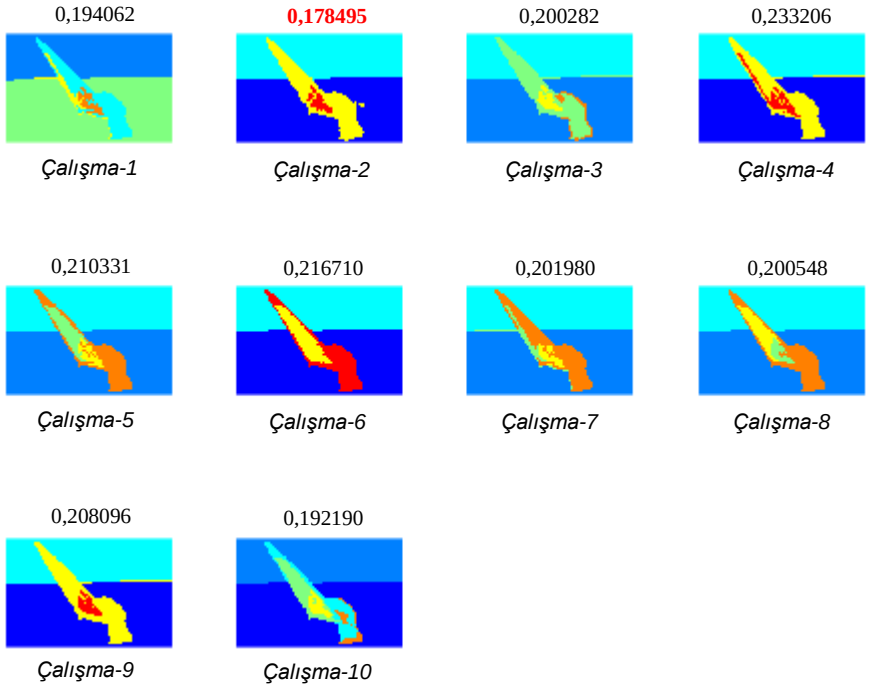


Şekil Ek-4.8. 42049 numaralı test görüntüsü için 500 *FEs* sonunda FCM ile alınan 10 çalıştırma sonucu

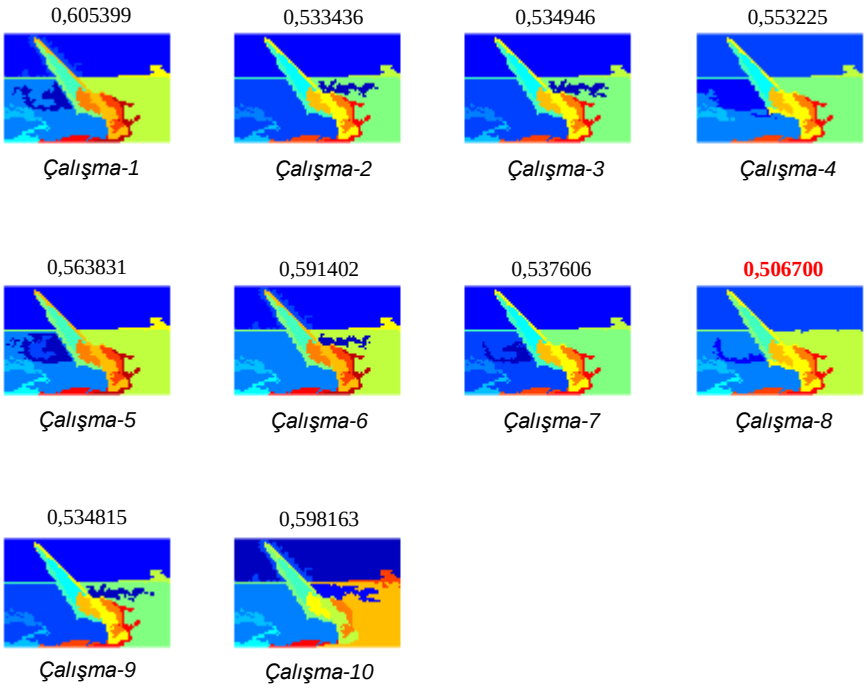


Şekil Ek-4.9. 62096 numaralı test görüntüsü için 500 FEs sonunda IBMO ile alınan 10 çalıştırma sonucu

Segentasyonların toplam tutarlılık hataları görüntülerin üzerinde verilmiştir.

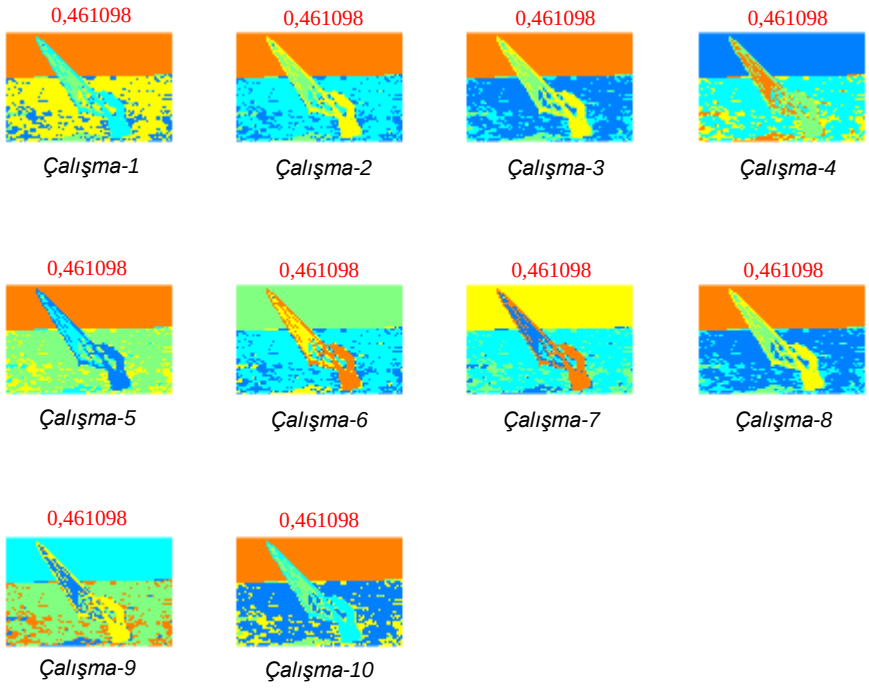


Şekil Ek-4.9. 62096 numaralı test görüntüsü için 500 FEs sonunda NSPSO ile alınan 10 çalıştırma sonucu

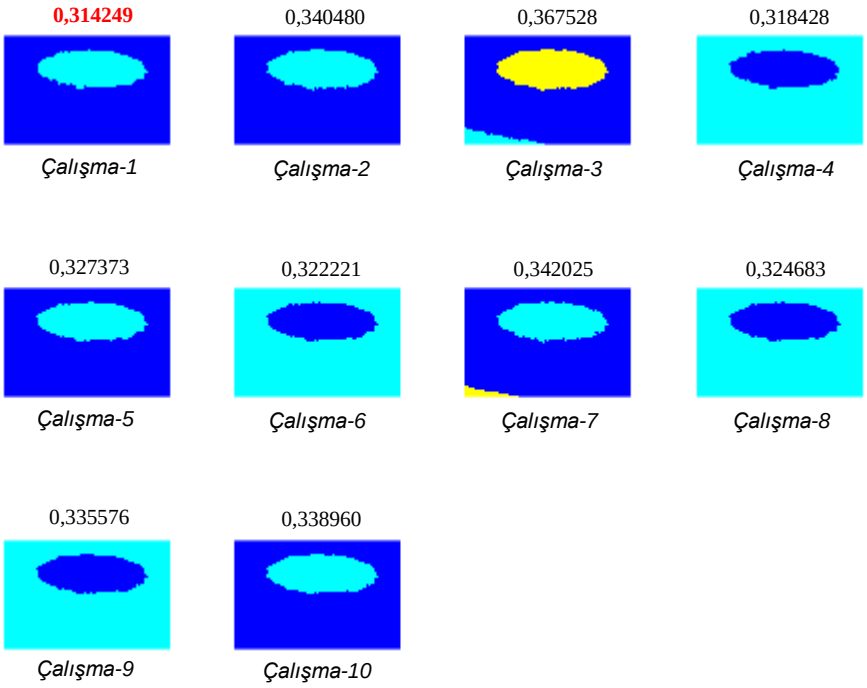


Şekil Ek-4.10. 62096 numaralı test görüntüsü için 500 *FEs* sonunda NSGA2 ile alınan 10 çalıştırma sonucu

Segentasyonların toplam tutarlılık hataları görüntülerin üzerinde verilmiştir.

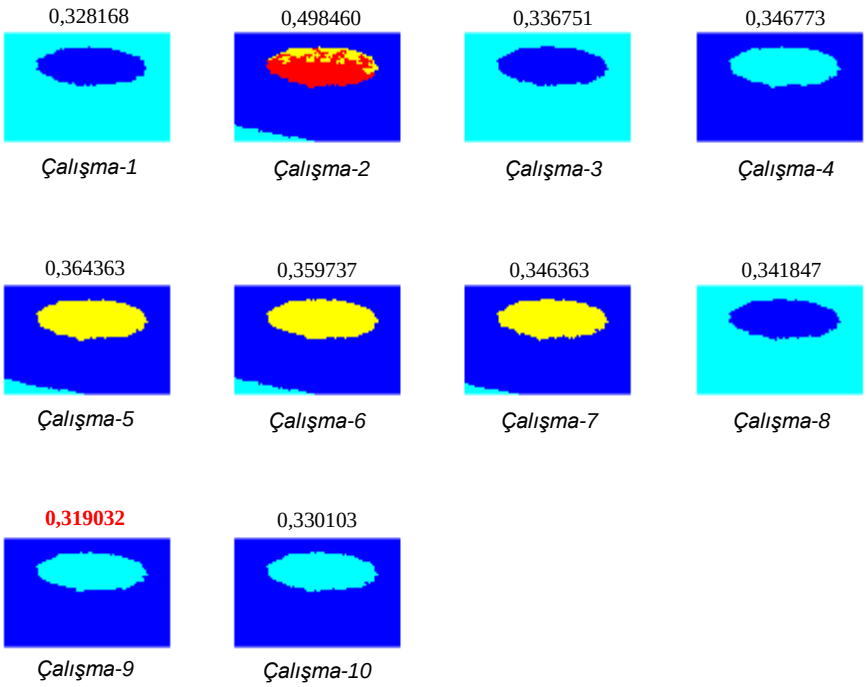


Şekil Ek-4.11. 62096 numaralı test görüntüsü için 500 *FEs* sonunda FCM ile alınan 10 çalıştırma sonucu

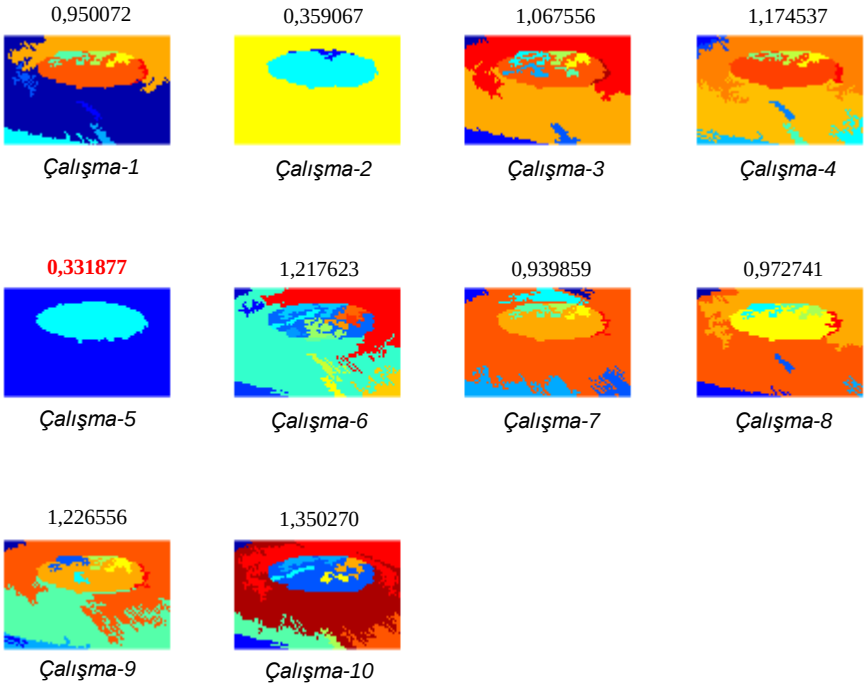


Şekil Ek-4.12. 86016 numaralı test görüntüsü için 500 FEs sonunda IBMO ile alınan 10 çalıştırma sonucu

Segentasyonların toplam tutarlılık hataları görüntülerin üzerinde verilmiştir.

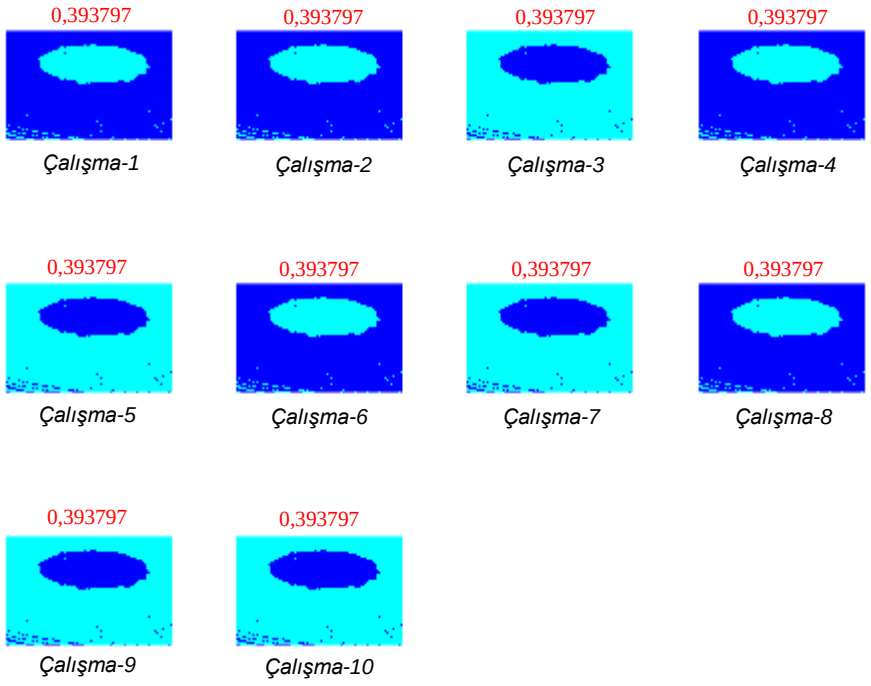


Şekil Ek-4.13. 86016 numaralı test görüntüsü için 500 FEs sonunda NSPSO ile alınan 10 çalıştırma sonucu

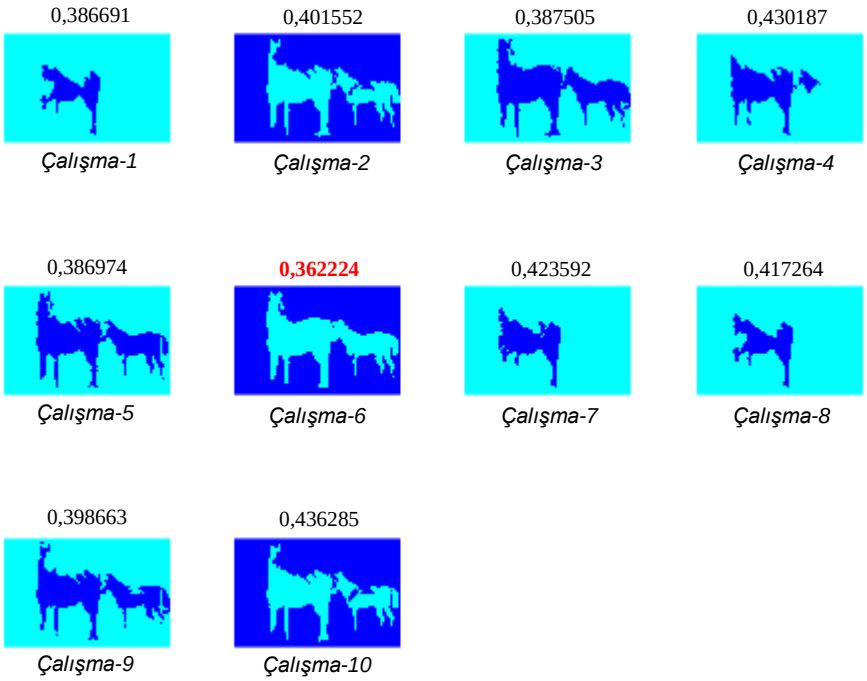


Şekil Ek-4.14. 86016 numaralı test görüntüsü için 500 *FEs* sonunda NSGA2 ile alınan 10 çalıştırma sonucu

Segentasyonların toplam tutarlılık hataları görüntülerin üzerinde verilmiştir.

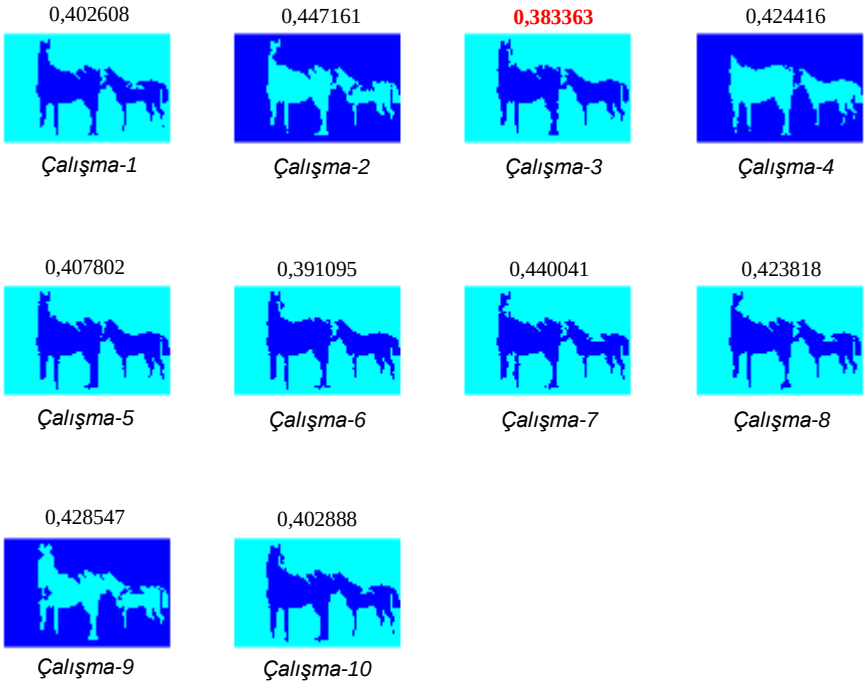


Şekil Ek-4.15. 86016 numaralı test görüntüsü için 500 *FEs* sonunda FCM ile alınan 10 çalıştırma sonucu

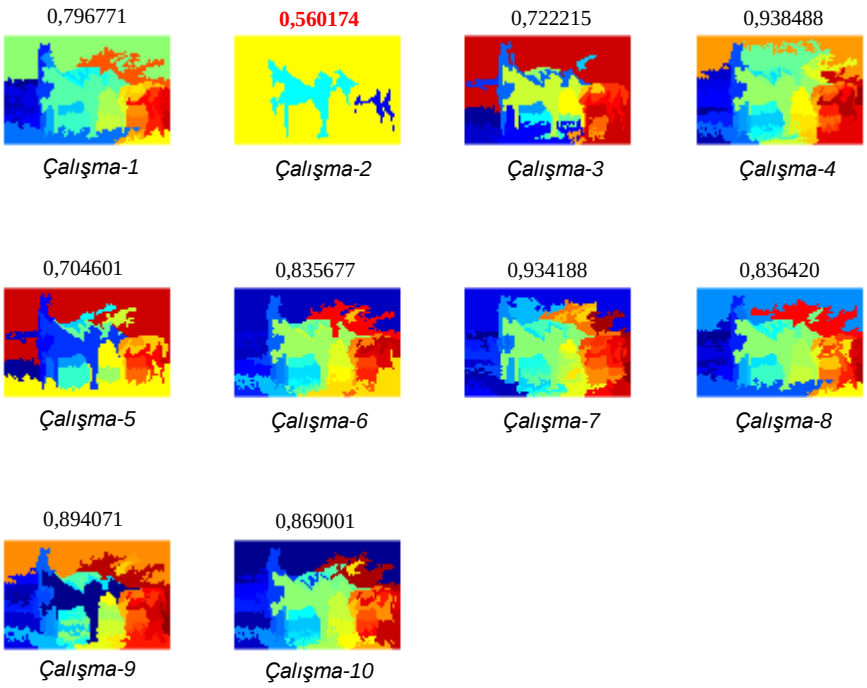


Şekil Ek-4.16. 113016 numaralı test görüntüsü için 500 FEs sonunda IBMO ile alınan 10 çalıştırma sonucu

Segentasyonların toplam tutarlılık hataları görüntülerin üzerinde verilmiştir.

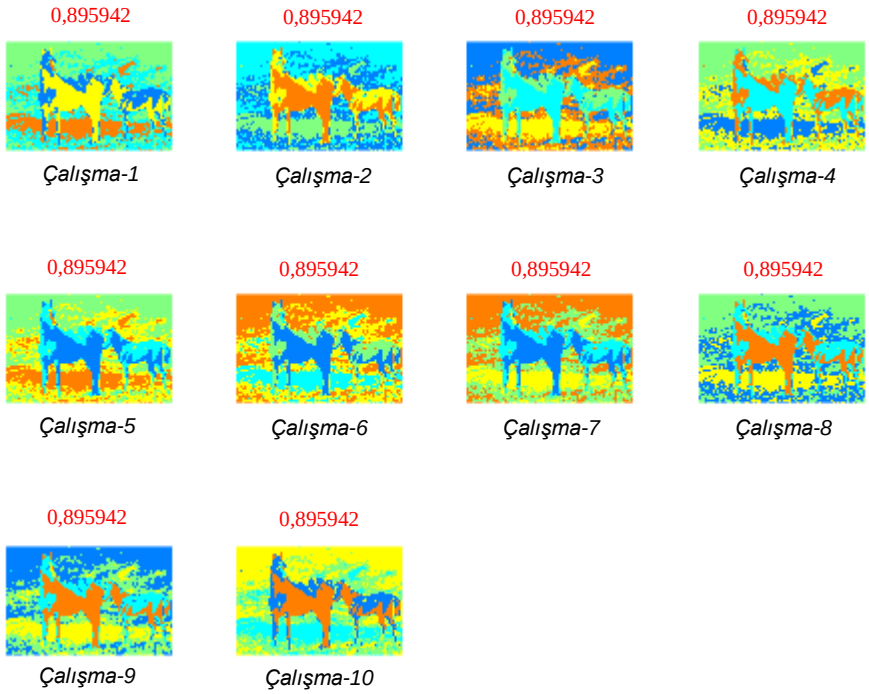


Şekil Ek-4.17. 113016 numaralı test görüntüsü için 500 FEs sonunda NSPSO ile alınan 10 çalıştırma sonucu

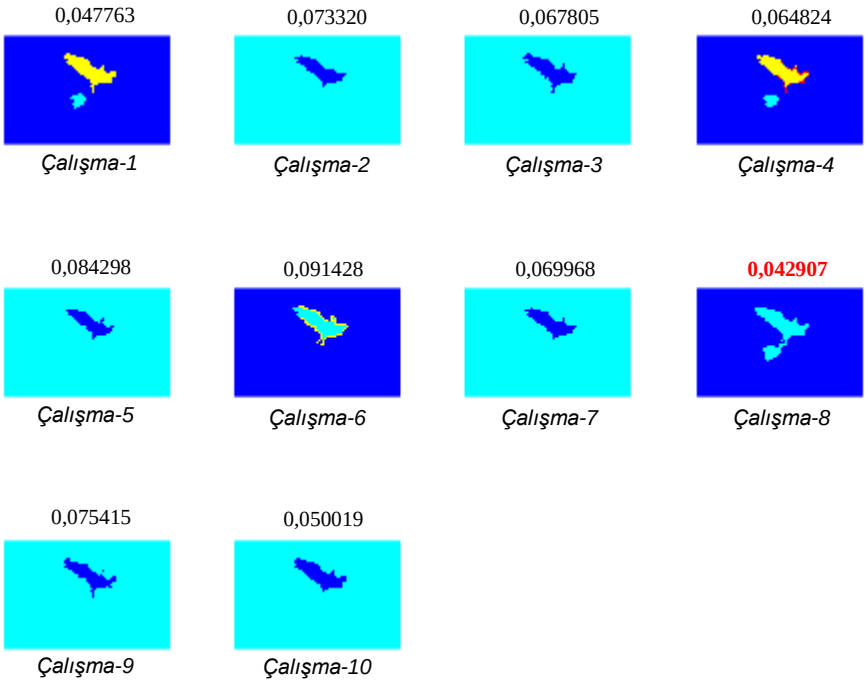


Şekil Ek-4.18. 113016 numaralı test görüntüsü için 500 FEs sonunda NSGA2 ile alınan 10 çalıştırma sonucu

Segentasyonların toplam tutarlılık hataları görüntülerin üzerinde verilmiştir.



Şekil Ek-4.19. 113016 numaralı test görüntüsü için 500 FEs sonunda FCM ile alınan 10 çalıştırma sonucu

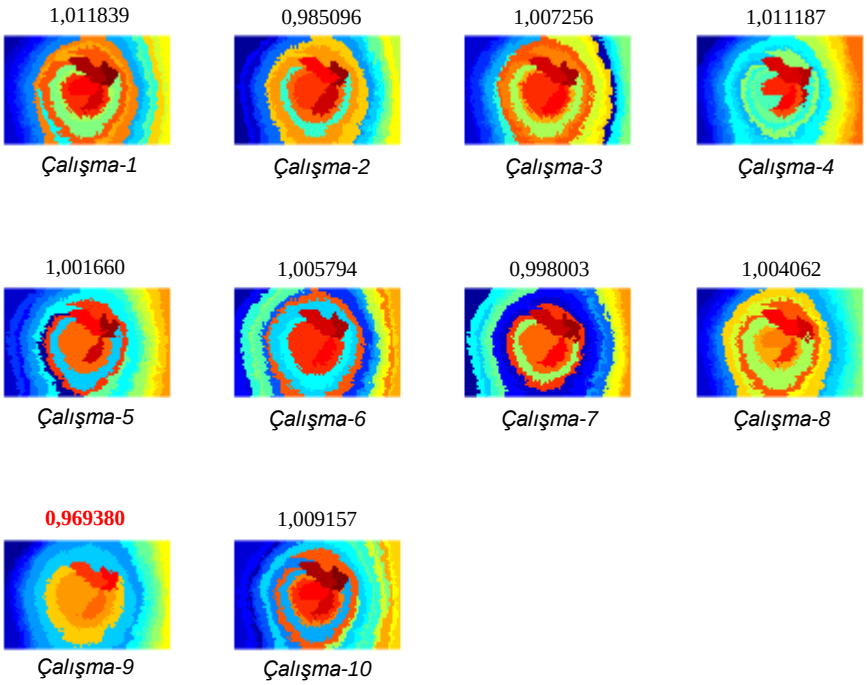


Şekil Ek-4.20. 135069 numaralı test görüntüsü için 500 FEs sonunda IBMO ile alınan 10 çalıştırma sonucu

Segentasyonların toplam tutarlılık hataları görüntülerin üzerinde verilmiştir.

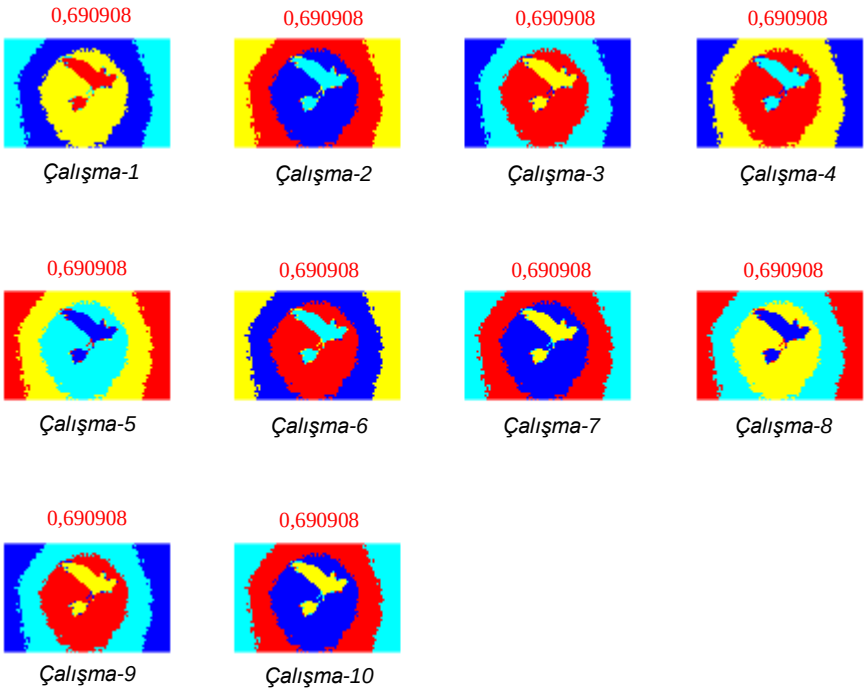


Şekil Ek-4.21. 135069 numaralı test görüntüsü için 500 FEs sonunda NSPSO ile alınan 10 çalıştırma sonucu

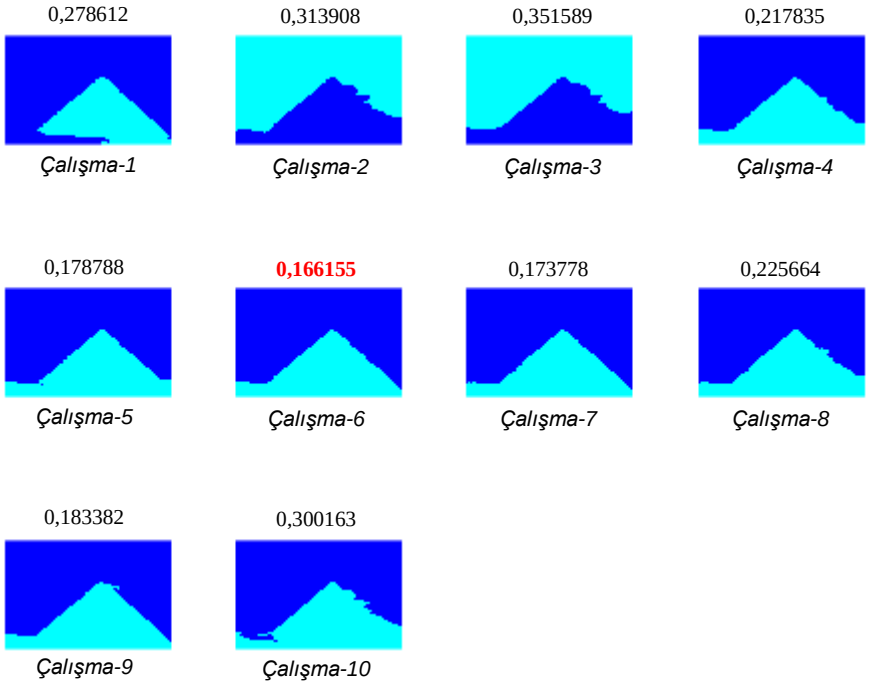


Şekil Ek-4.22. 135069 numaralı test görüntüsü için 500 FEs sonunda NSGA2 ile alınan 10 çalıştırma sonucu

Segentasyonların toplam tutarlılık hataları görüntülerin üzerinde verilmiştir.

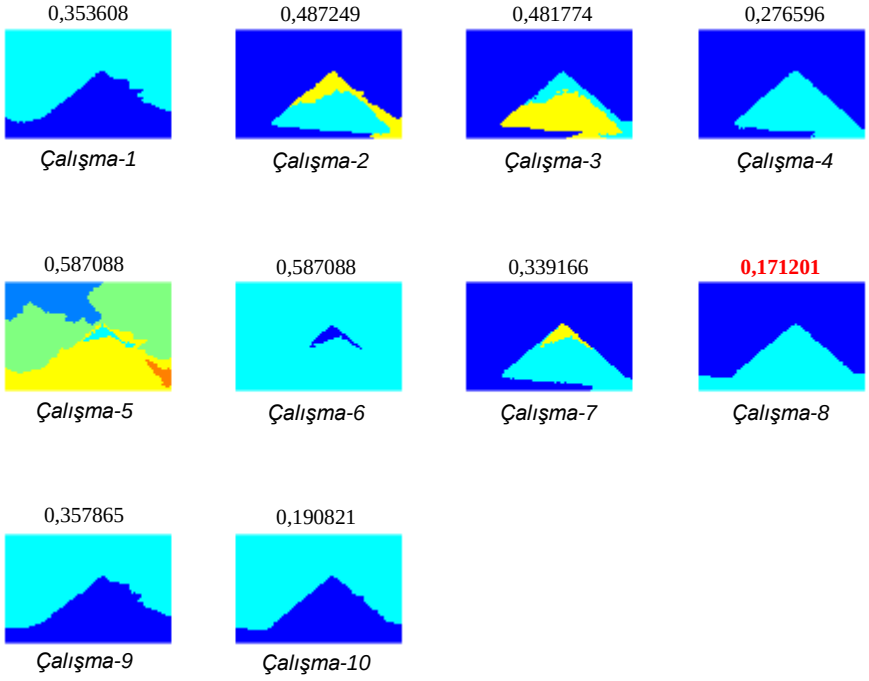


Şekil Ek-4.23. 135069 numaralı test görüntüsü için 500 FEs sonunda FCM ile alınan 10 çalıştırma sonucu

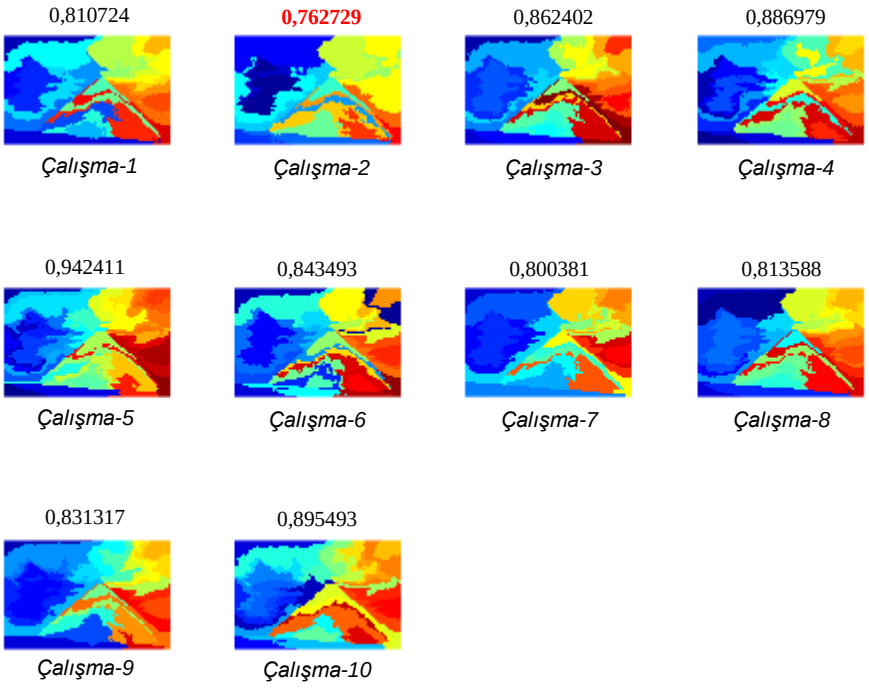


Şekil Ek-4.24. 299091 numaralı test görüntüsü için 500 FEs sonunda IBMO ile alınan 10 çalıştırma sonucu

Segentasyonların toplam tutarlılık hataları görüntülerin üzerinde verilmiştir.

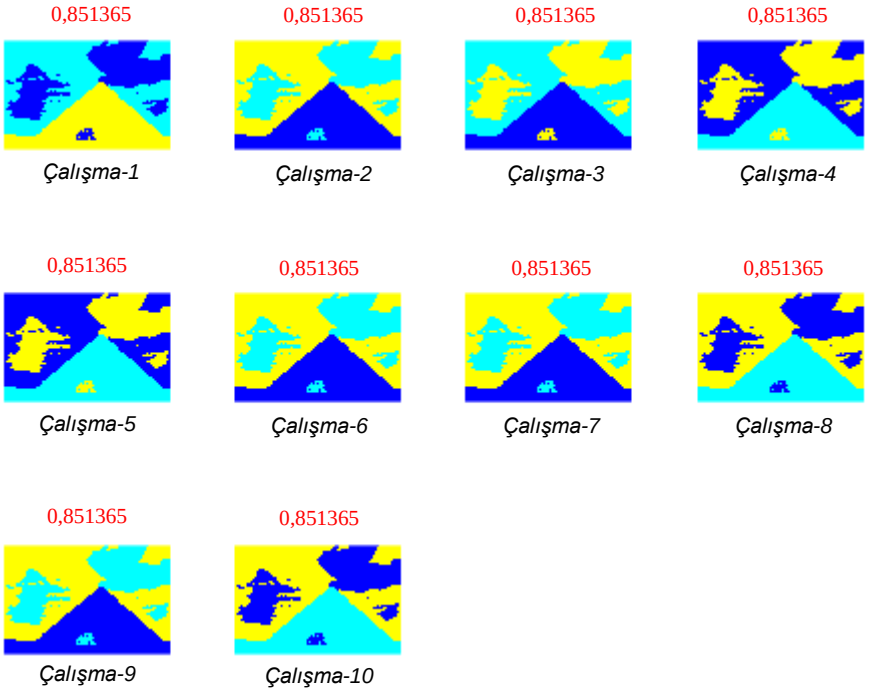


Şekil Ek-4.25. 299091 numaralı test görüntüsü için 500 FEs sonunda NSPSO ile alınan 10 çalıştırma sonucu



Şekil Ek-4.26. 299091 numaralı test görüntüsü için 500 FEs sonunda NSGA2 ile alınan 10 çalıştırma sonucu

Segentasyonların toplam tutarlılık hataları görüntülerin üzerinde verilmiştir.



Şekil Ek-4.26. 299091 numaralı test görüntüsü için 500 FEs sonunda FCM ile alınan 10 çalıştırma sonucu

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı : Tahir SAĞ
Uyruğu : T.C.
Doğum Yeri ve Tarihi : KONYA / 1983
Telefon : (332) 223 3332
e-mail : tahirsag@selcuk.edu.tr, tahirsag@gmail.com

EĞİTİM

Derece		Adı, İlçe, İl	Bitirme Yılı
Üniversite	:	Selçuk Üniversitesi Mühendislik-Mimarlık Fakültesi Bilgisayar Mühendisliği Bölümü	**2005**
Yüksek Lisans	:	Selçuk Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı	**2008**
Doktora	:	Selçuk Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı	**2015**

İŞ DENEYİMLERİ

Yıl	Kurum	Görevi
2005-2011	Selçuk Üniversitesi Teknik Eğitim Fakültesi Elektronik ve Bilgisayar Eğitimi Bölümü	Araştırma Görevlisi
2011-2015	Selçuk Üniversitesi Teknoloji Fakültesi Bilgisayar Mühendisliği Bölümü	Öğretim Görevlisi

YAYINLAR

1. Tahir SAĞ ve Mehmet ÇUNKAŞ, 2015, Color Image Segmentation based on Multiobjective Artificial Bee Colony Optimization, *Applied Soft Computing*, 34, 389-401. doi:10.1016/j.asoc.2015.05.016.

   (Doktora tezinden yapılmıştır)

2. Tahir SAĞ ve Mehmet ÇUNKAŞ, 2014, A New ABC-based Multi-Objective Optimization Algorithm with an Improvement Approach (IBMO: Improved Bee Colony Algorithm for Multi-Objective Optimization), *Turkish Journal of Electrical Engineering &Computer Sciences.* doi: 10.3906/elk-1402-310.

   (Doktora tezinden yapılmıştır)

3. Tahir SAĞ ve Mehmet ÇUNKAŞ, 2012, Development Of Image Segmentation Techniques Using Swarm Intelligence, *ICCIT 2012: 1st Taibah University International Conference on Computing and Information Technology,* Al-Madinah Al-Munawwarah, Saudi Arabia.

   (Doktora tezinden yapılmıştır)

4. Tahir SAĞ ve Mehmet ÇUNKAŞ, 2012, Application of Multi-objective Particle Swarm Optimization Algorithm in Image Segmentation*, 3rd World Conference on Information Technology*, Barcelona - Spain*.*

   (Doktora tezinden yapılmıştır)

5. Mustafa ARSLAN, Mehmet ÇUNKAŞ, Tahir SAĞ, 2012, Determination of induction motor parameters with differential evolution algorithm, *Neural Computing & Applications*, 21 (8), 1995-2004. doi: 10.1007/s00521-011-0612-8.

6. Tahir SAĞ, Mehmet ÇUNKAŞ, 2010, Comparison Of Heuristic Methods Applied To Optimization Problems, 2010, *International Conference on Computer Science and Software Engineering ICCSSE 2010*, Paris – France July 28-30.

   (Yüksek lisans tezinden yapılmıştır)

7. Tahir SAĞ, Mehmet ÇUNKAŞ, 2009, A tool for multiobjective evolutionary algorithms, *Advances in Engineering Software*. doi:10.1016/j.advengsoft.2009.01.001.

   (Yüksek lisans tezinden yapılmıştır)

8. Mehmet ÇUNKAŞ, Tahir SAĞ, 2009, Efficiency determination of induction motors using multi-objective evolutionary algorithms*, Advances in Engineering Software.* doi:10.1016/j.advengsoft.2009.07.008.

   (Yüksek lisans tezinden yapılmıştır)

9. Tahir SAĞ, Mehmet ÇUNKAŞ, 2009, Çok Amaçlı Genetik Algoritmalar İçin Bir Çevrimdışı Performans Değerlendirmesi*, 5. Uluslararası İleri Teknolojiler Sempozyumu (IATS'09),* 13-15 Mayıs 2009, Karabük, Türkiye.

   (Yüksek lisans tezinden yapılmıştır)

10. Mehmet ÇUNKAŞ, Tahir SAĞ, Mustafa ARSLAN, 2009, Determination of Induction Motor Parameters by Differential Evolution Algorithm and genetic algorithms, *The 2nd International Conference on Advanced Computer Theory and Engineering (ICACTE 2009),* Cairo, Egypt, 1, 777-784.

11. Mehmet ÇUNKAŞ, Tahir SAĞ, 2008, In-situ efficiency determination of an induction motor using the multiobjective genetic algorithms, *Ims'2008 6th International Symposium on Intelligent and Manufacturing Systems*, Sakarya.

    (Yüksek lisans tezinden yapılmıştır)

12. Tahir SAĞ, Mehmet ÇUNKAŞ, 2007, Multiobjective Genetic Estimation to Induction Motor Parameters, *International Aegean Conference on Electric Machines, Power Electronics(ACEMP) and Electro motion Joint Conference*, Bodrum.

    (Yüksek lisans tezinden yapılmıştır)