

**T.C.
CELAL BAYAR ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**YÜKSEK LİSANS TEZİ
ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI**

KARİDES SÜRÜSÜ ALGORİTMASI İLE GÖRÜNTÜ İŞLEME

FATMA HARMAN

**Danışman
Yrd.Doç.Dr. İLKER KILIÇ**



MANİSA-2016

**Fatma
HARMAN**

KARIDES SÜRÜSÜ ALGORİTMASI İLE GÖRÜNTÜ İŞLEME

2016

Tez Sırtı Örneđi

TEZ ONAYI

FATMA HARMAN tarafından hazırlanan "**Karides Sürüsü Algoritması ile Görüntü İşleme**" adlı tez çalışması 1/08/2016 tarihinde aşağıdaki jüri üyeleri önünde Celal Bayar Üniversitesi Fen Bilimleri Enstitüsü **Elektrik Elektronik Mühendisliği Anabilim Dalı**'nda **YÜKSEK LİSANS** olarak başarı ile savunulmuştur.

Danışman **Yrd.Doç. Dr. İlker KILIÇ**
Manisa Celal Bayar Üniversitesi

.....

Jüri Üyesi **Yrd.Doç.Dr. Erkan Zeki ENGİN**
Ege Üniversitesi

Jüri Üyesi **Yrd.Doç.Dr. Mustafa NİL**
Manisa Celal Bayar Üniversitesi

.....

TAAHHÜTNAME

Bu tezin Celal Bayar Üniversitesi Mühendislik Fakültesi Elektrik Elektronik Mühendisliği Bölümü'nde, akademik ve etik kurallara uygun olarak yazıldığını ve kullanılan tüm literatür bilgilerinin referans gösterilerek tezde yer aldığını beyan ederim.

FATMA HARMAN



İÇİNDEKİLER

	Sayfa
İÇİNDEKİLER	I
ŞİMGELER VE KISALTMALAR DİZİNİ	II
ŞEKİLLER DİZİNİ.....	III
TABLO DİZİNİ	IV
TEŞEKKÜR.....	V
ÖZET.....	VI
ABSTRACT.....	VII
1. GİRİŞ	1
2. VEKTÖR KUANTALAMA.....	8
2.1. Linde Buzo Gray Algoritması	8
3. METASEZGİSEL ALGORİTMALAR	11
3.1. Karides Sürüsü Algoritması	11
3.1.1. Karides Sürüsü Algoritmasının Temel Yapısı.....	13
3.1.2. Diğer Karideslerin Neden Olduğu Hareket	14
3.1.3. Yiyecek Arama Hareketi	15
3.1.4. Random Yayılma	16
3.1.5. Karides Sürüsünün Hareketini Tamamlama Süreci	17
3.1.6. Karides Sürüsü Algoritmasının Görüntüye Uygulanması.....	17
3.2. Parçacık Sürüsü Algoritması	20
3.2.1. Parçacık Sürüsü Algoritmasının İşleyişi.....	20
3.3. Karınca Kolonisi Algoritması	23
3.3.1. Karınca Kolonisi Algoritmasının İşleyişi.....	24
3.4. Havuz Tabanlı Genetik Algoritma	28
3.4.1. Havuz Tabanlı Genetik Algoritmanın İşleyişi	30
4. SONUÇLAR VE ÖNERİLER	32
KAYNAKLAR	45
ÖZGEÇMİŞ	52

SİMGELER VE KISALTMALAR DİZİNİ

KSA	Karides Sürüsü Algoritması
PSO	Parçacık Sürüsü Algoritması
KKA	Karınca Kolonisi Algoritması
HTGA	Havuz Tabanlı Genetik Algoritması
ACO	Karınca Kolonisi Algoritması
PBGA	Havuz Tabanlı Genetik Algoritma
KHA	Karides Sürüsü Algoritması
PSO	Parçacık Sürüsü Algoritması
LZW	Lempel- Ziv-Welch
TIFF	Etiketlenmiş Resim Dosya Formatı
BMP	Bit Haritası Resim Dosyası
PCX	Resim Görüntü
PCM	DarbeKod Modülasyonu
DPCM	Ayrımsal Darbe Kod Modülasyonu
JPEG	Birleşik Fotoğraf Uzmanları Grubu
JPEGXR	Genişletilmiş Aralıklı Birleşik Fotoğraf Uzmanları Grubu
IFS	Ötelenmiş Fonksiyon Sistemi
PIFS	Bölüntülü Ötelenmiş Fonksiyon Sistemi
MPEG	Hareketli Fotoğraf Uzmanlar Grubu
DjVU	Dejavu Resim Sıkıştırma Algoritması
CINEFORM	Görüntü Çözücü Formu
BBC DIRAC	BBC Görüntü Çözücü
LBG	Linde-Buzo-Gray
MSE	Ortalama Kareysel Hata
PSNR	Doruk Sinyal Gürültü Oranı
MATLAB	Matris Laboratuvarı

ŞEKİLLER DİZİNİ

	Sayfa
Şekil 2.1. Örnek Vektör Kuantalama Alanları	9
Şekil 2.2. LBG Algoritmasının Akış Diyagramı.....	10
Şekil 3.1. Antarktik Karidesleri	12
Şekil 3.2. Önerilen Çözüm Yönteminin Akış Diyagramı.....	19
Şekil3.3. PSO Akış Diyagramı.....	22
Şekil 3.4. Karınca Kolonisi Algoritması Akış Diyagramı.....	27
Şekil 3.5. Havuz Tabanlı Genetik Algoritması Akış Diyagramı.....	30
Şekil 4.1. 8 Kod Kelimesine Sahip Populasyona KSA Uygulanması Sonucu...	32
Şekil 4.2. 16 Kod Kelimesine Sahip Populasyona KSA Uygulanması Sonucu..	33
Şekil 4.3. 32 Kod Kelimesine Sahip Populasyona KSA Uygulanması Sonucu..	33
Şekil 4.4. 64 Kod Kelimesine Sahip Populasyona KSA Uygulanması Sonucu..	34
Şekil 4.5. 8 Kod Kelimesine Sahip Populasyona Tüm Algoritmaların Uygulanması Sonucu.....	35
Şekil 4.6. 16 Kod Kelimesine Sahip Populasyona Tüm Algoritmaların Uygulanması Sonucu.....	36
Şekil 4.7. 32 Kod Kelimesine Sahip Populasyona Tüm Algoritmaların Uygulanması Sonucu.....	37
Şekil 4.8. 64 Kod Kelimesine Sahip Populasyona Tüm Algoritmaların Uygulanması Sonucu.....	38
Şekil 4.9. 8 Kod Kelimeli Populasyona HTGA Uygulanması Sonucu.....	39
Şekil 4.10. 16 Kod Kelimeli Populasyona HTGA Uygulanması Sonucu.....	40
Şekil 4.11. 32 Kod Kelimeli Populasyona HTGA Uygulanması Sonucu.....	40
Şekil 4.12. 64 Kod Kelimeli Populasyona HTGA Uygulanması Sonucu.....	41
Şekil 4.13. Orijinal Lena Resmi.....	42
Şekil 4.14. KSA Genetik Operatörsüz.....	42
Şekil 4.15. KSA Mutasyon Operatörüyle.....	43
Şekil 4.16. KSA Çaprazlama Operatörüyle.....	43
Şekil 4.17. KSA Genetik Operatörlerle Boyut.....	43
Şekil 4.18. PSA ile Lena Resmi.....	43
Şekil 4.19. KKA Genetik Operatörsüz.....	43
Şekil 4.20. KKA Mutasyon Operatörüyle.....	43
Şekil 4.21. KKA Çaprazlama Operatörüyle.....	43
Şekil 4.22. KKA Genetik Operatörlerle Boyut.....	43
Şekil 4.23. HTGA ile Lena Resmi.....	44

TABLO DİZİNİ

Sayfa

Tablo 4.2. Tüm Algoritmaların 8 Kod Kelimeli Populasyona Uygulanması.....	35
Tablo 4.2. Tüm Algoritmaların 16 Kod Kelimeli Populasyona Uygulanması.....	36
Tablo 4.3. Tüm Algoritmaların 32 Kod Kelimeli Populasyona Uygulanması.....	37
Tablo 4.4. Tüm Algoritmaların 64 Kod Kelimeli Populasyona Uygulanması.....	38
Tablo 4.5. Havuz Tabanlı Genetik Algoritmasının 8 Kod Kelimeli Populasyona Uygulanması.....	39
Tablo 4.6. Havuz Tabanlı Genetik Algoritmasının 16 Kod Kelimeli Populasyona Uygulanması.....	39
Tablo 4.7. Havuz Tabanlı Genetik Algoritmasının 32 Kod Kelimeli Populasyona Uygulanması.....	40
Tablo 4.8. Havuz Tabanlı Genetik Algoritmasının 64 Kod Kelimeli Populasyona Uygulanması.....	41

TEŐEKKÖR

Çalıőmamın her aőamasında bana destek olan, bilgi ve deneyimleri ile yol gösteren danıőman hocam Sayın Yrd.Doç.Dr. İLKER KILIÇ'a öęrenim hayatım boyunca beni maddi ve manevi olarak destekleyen ve hep yanımda olan aileme yürekten teőekkür ederim.

FATMA HARMAN
Manisa, 2016

ÖZET

Yüksek Lisans Tezi

Karides Sürüsü Algoritması ile Görüntü İşleme

Fatma HARMAN

Celal Bayar Üniversitesi

Fen Bilimleri Enstitüsü

Elektrik Elektronik Mühendisliği Anabilim Dalı

Danışman: Yrd.Doç.Dr İlker KILIÇ

Teknolojinin günden güne gelişmesiyle, sıkıştırılmamış görüntünün transferinin ve saklanması zorluk derecesi büyük ölçüde artmış ve buna bağlı olarak daha fazla sıkıştırma oranı ve görüntü kalitesi yakalamaya ilişkin birçok yeni çözüm yöntemi geliştirilmiştir. Bu anlamda, literatürde çeşitli problemlerin çözümünde kullanılan metasezgisel yaklaşımların bu alanda kullanımı yaygınlaşmıştır.

Bu çalışmada metasezgisel algoritmaların genel yapısı ve görüntü sıkıştırma üzerindeki etkisinin incelenmesi amaçlanmış ve bu amaç doğrultusunda Karides Sürüsü Algoritması (KSA), Genetik Operatörlü KSA, KKA, Genetik Operatörlü KKA, Parçacık Sürüsü Algoritması (PSA), Havuz Tabanlı Genetik Algoritma (HTGA) standart görüntü üzerinde uygulanmıştır.

KSA ve KKA standart görüntüye uygulanmış, sonraki aşamada ise yerel minimum noktalarından kurtulmak amacıyla algoritmalar genetik operatörlerle birleştirilmiştir. Böylece, ortalama karesel hata ve sıkıştırma oranı açısından daha iyi sonuçlar elde edilmiştir. Yapılan analizlere ek olarak ele alınan probleme sırasıyla PSA ve GA uygulanmıştır. Algoritmaların çözüm performansı karşılaştırıldığında, ortalama karesel hata açısından en iyi sonucun HTGA ile elde edildiği görülmüştür. Genetik algoritmayı sırasıyla, Genetik Operatörlü KSA, PSA, Genetik Operatörlü KKA, KSA ve KKA izlemektedir.

Anahtar Kelimeler: Görüntü Sıkıştırma, Metasezgisel Algoritmalar, Karides Sürüsü Algoritması, Genetik Algoritma, Karınca Kolonisi Algoritması, Parçacık Sürüsü Algoritması

2016, 63 sayfa

ABSTRACT

M.Sc.

Image Processing with Krill Herd Algorithm

Fatma HARMAN

**Celal Bayar University
Graduate School of Electrical and Electronical Engineering
Department of Electrical and Electronical Engineering**

Supervisor: Assoc. Prof. Dr. İlker KILIÇ

The holding and transferring of the compressed image is becoming increasingly more difficult and because of this a new solution technique with a more compressed rate and image quality has been developed. The use of metaheuristic algorithms as evident throughout various literature has become more popular as a way to solve these various problems.

This research will aim to analyse the effects of the metaheuristic algorithms on image compression and in accordance with this aim, Krill Herd Algorithm with Genetic Operators, Ant Colony Optimisation with Genetic Operator, Particle Swarm Optimisation and Pool Based Genetic Algorithm (PBGA) are all applied on this image.

Krill Herd Algorithm and Ant Colony Algorithm are applied to the standard image and will be applied in order to unite with the genetic operators to escape from the local minimum. As a result, the mean square error and compression rate will be better. In addition to the analysis, PSO and PBGA will be applied to this problem. When the solution performances of the algorithms are compared with the same compression rate the best result is seen with PBGA. PSO, ACO with GO, ACO with GO, KHA and ACO all follow PBGA respectively.

Keywords: Image Compression, Metaheuristic Algorithms, Krill Herd Algorithm, Pool Based Genetic Algorithm, Ant Colony Algorithm, Particle Swarm Optimisation

2016, 63 pages

1. GİRİŞ

Elektronik ortamda sinyal işleme, teknolojinin gelişmesiyle günden güne hızlı bir şekilde artmıştır. Bu artış, bilgisayar teknolojisine bağlı olarak meydana gelmiştir. Bilgisayarların giderek küçülmesi, bellek kapasitelerinin ve veri işleme hızlarının artması çoklu ortamdan en iyi şekilde faydalanmayı gerektirmiştir [1]. Kullanıcıların çoklu ortamdan en fazla faydayı sağlayabilmesi için saklama, iletme, işleme gibi etkenlerin bir araya gelmesi gerekmektedir. Bu etkenler yeterli olmamasına rağmen yakalama, sıkıştırma, görüntüleme kabiliyetleri ve farklı çoklu ortam bilgileri için gereklidir [2].

Gelişen çoklu ortam ve teknolojiyle birlikte bilgiyi hızlı bir şekilde elde etme ve bilgidен yararlanmaya olan gereksinim artmıştır. Bu gereksinim bilginin verimli bir şekilde saklanabilmesi, bilgiye hızlı bir şekilde ulaşıp iletilmesi durumunu beraberinde getirmektedir [3]. Bilgiyi sıkıştırma, belleğin kapasite değerinin azalmasını ve bant genişliğinin daralmasını sağladığı için veri transferlerinde önemli bir noktaya sahiptir. Görüntü sıkıştırma ise bilginin fark edilebilir bir kaybı olmadan görüntünün iletimi ve depolanması için gerekli olan bit sayısının azalmasıyla ilgili bir tekniktir. Kayıplı ve kayıpsız olmak üzere iki tip görüntü sıkıştırma tekniği vardır. Kayıplı teknikte bilginin kaybı vardır ve belirli bir hata oranında orijinal görüntü elde edilir. Kayıpsız sıkıştırmada ise bilginin kaybı yoktur ve orijinal görüntü düşük sıkıştırma oranında tekrar oluşturulur [4].

Kayıpsız görüntü sıkıştırma teknikleri olarak literatürde Huffman Kodlaması, Aritmetik Kodlaması, Sözlük Tabanlı Kodlama ve Run-Length Kodlaması bulunmaktadır. Huffman kodlaması 1952 yılında David. A. Huffman tarafından geliştirilmiş olup her sembol için ikilik bir kod üretir. Bu kodlar bit haritası şeklindedir ve veri içerisinde en az kullanılan için en uzun, en çok kullanılan kod için en kısa kod üretilir [5,6]. Az sayıda karakterli ve büyük boyutlu verilerde kullanışlıdır [6]. Huffman kodlamasında oluşturulan ağacın veriye eklenmesi gerekmektedir. Bu da verimi düşüreceğinden bu tip sorunların üstesinden gelebilmek için 1985 yılında Donald E. Knuth tarafından Dinamik Huffman Kodlaması geliştirilmiştir [7].

Aritmetik Kodlama Huffman kodlamadan sonra en çok kullanılan kayıpsız sıkıştırma tekniğidir. 1948'de Shannon bir makalesinde Aritmetik kodlamaya yakın bir kavramdan bahsetmiştir. Peter Elias bu fikrin özyinelemeli bir uyarlamasını gerçekleştirmiştir. Fakat bu çalışmaları yayınlamadığından dolayı bu çalışmalar Abramson'un 1963'teki kitabından anlaşılmıştır [8]. Modern aritmetik kodlamanın başlaması, sonlu duyarlılık problemiyle birlikte başlamıştır [Pascoe ve Rissanen, 1976]. Pratik aritmetik kodlama algoritmaları [Rissanen ve Langdon, 1979] ve veri sıkıştırmada kullanılması [Witten vd., 1987] [Moffat vd., 1995] [Howard ve Vitter, 1994] ile bir çok makale bulunmaktadır. Görüntü sıkıştırma da ilk defa 1989 yılında C. D. Hardin ve S. Zabel tarafından uygulanmıştır [9].

Veri parçasını kodlamada 0 ve 1 aralığındaki gerçel sayı aralığı ile temsil etmeye dayalı olan aritmetik kodlamada, temsil edilen veri kümesindeki her sembol, bu gerçel sayı aralığı daraltır. Kullanım sıklığı az olanlar bu aralığı daraltırken fazla olanlar daha az daraltmaktadır.

Bir diğer kayıpsız sıkıştırma tekniği olan Sözlük Tabanlı Kodlama, bir metin ya da görüntüdeki yinelenen kalıpların belirlenip bir sözlük oluşturması ve bu oluşturulan sözlükteki kalıpların sözlük sıra numarasının kodlanmasına dayalı bir tekniktir. Sözlük Tabanlı Kodlama literatüre ilk defa 1993 yılında Rodney J. Smith patentiyle kazandırılmıştır [10]. Ian H. Witten, Alistair Moffat ve Timothy C. Bell 1994 yılında geliştirilmiştir [11].

Statik, yarı statik ve dinamik olmak üzere üç kategoriye ayrılmıştır. En çok kullanılan dinamik sözlük modeli olup hem sözlük oluşturulup hem sıkıştırma gerçekleştirilir. Bu tekniklerin büyük bir çoğunluğu Jacob Ziv ve Abraham Lempel tarafından yayınlanan makalelerle geliştirilmiştir [12,13]. 1977'deki makaleyi temel alana LZ77, 1978'dekilere LZ78 denir. LZ78'in en iyi sıkıştırmayı sağlayan LZW algoritması Terry Welch tarafından 1984 yılında geliştirilmiştir [14].

Run-Length Kodlama ise bir diğerk kayıpsız sıkıştırma tekniğı olup, ilk olarak 1959 yılında J.Capon tarafından ortaya atılmıştır [15]. Run-Length kodlamasının temelinde bir değerin ardışık tekrar etmesi durumunda, ilk olarak o değeri bir defa kodlayıp ardından tekrar sayısını kodlama prensibi yatmaktadır. Her türlü veri için kullanılsada tekrar eden sembolün sıkıştırılması iyi bir sıkıştırma sağladığı için daha çok TIFF, BMP, PCX görüntü formatlarında uygulanır.

Kayıplı görüntü sıkıştırmada birçok ara işlem bulunur. Bu yöntemlerde ilk olarak önce renk dönüşümü gerçekleştirilir ve ardından fazlalıklar atılır [8]. Daha sonra dönüşüm yapılarak veri daha az bit ile ifade edilir. Niceleme işlemi ile uygun bit oranına indirgenip kayıpsız bir yöntem ile sıkıştırılma işlemi gerçekleştirilir. Günümüzde en yaygın kullanılan kayıplı sıkıştırma JPEG, JPEG2000 ve JPEGXR dir.

Kayıplı sıkıştırma tekniklerinden biri olan Skalar Kuantalama, genellikle analog sinyallerin sayısal biçime dönüştürülebilmesi için örnekleme ve niceleme adımlarından geçirilmesi gerektiğinde kullanılan bir yöntemdir. Bu işlemler genellikle Darbe Kod Modülasyonu yöntemi ile yapılır. Genellikle sıkıştırılmamış ses sinyalinin sıkıştırılmasında kullanılan bir methodtur. İlk olarak Britanyalı mühendis Alec Reeves 1937 yılında Fransa'da International Telephone and Telegraph'ta çalışırken farkında olmadan ses sinyali için darbe kod modülasyonu düşünmüş ve 1943'te tasarlayarak patentini almıştır [16]. 1940 ların sonu 1950 lerin başında darbe kod modülasyonu, delikli levha elektoduna sahip katot ışın tüpü kullanarak ossiloskopta dikey sapma ışığın daha aşağı ya da yukarı delikli levhadan geçmesine sebep olan giriş analog sinyal tarafından kontrol edilirken yatay pozisyonda aynı oranda ışın sürüklenmiştir. Delikli levha ışığı her bir zamanda bir bit üreterek toplamış ve geçirmiştir [17]. Bu konuda daha önce patent alan Bernard M.Oliver, Claude Shannon ve J.R.Pierce 1948 yılında tekniğın temelini anlatan çalışmayı oluşturmuşlardır [18].

PCM sinyal çıkışına kayıpsız sıkıştırma tekniğini direkt uygulamak, değer aralıkları ve korelasyonun yüksek olması nedeniyle etkisi fazla olmayacağından DPCM gibi bir yöntem uygulamak korelasyon azaltıp, daha başarılı sıkıştırma gerçekleştirebilir.

Öngörülü Kodlama ise telefon sisteminde çok popüler olup hala sayısal telefon iletişimde kullanılmaktadır. Basit fark kodlama yöntemleri her örneğin bir önceki örneğe farkını kodlarken, DPCM ise kodlanacak olan örnek değerinin önceki örneklerin yardımıyla tahmin ederek gerçek değer ile tahmin değeri arasındaki farkı kodlar [8]. Yani artık değer kodlaması olarak düşünülebilir. Tahmin yapmanın ana fikri fark değerlerini daha da azaltmaktır. Literatürde C. Chapin Cutler tarafından 1950’de icat edilip patenti alınmıştır [19].

Dönüşüm Kodlaması, kayıplı sıkıştırma tekniği olarak Ayrık Kosinüs Transfer Dönüşümü ve Dalgacık Transfer Dönüşümü kullanılmaktadır. Ayrık Kosinüs Transfer Dönüşümü sabit ve hareketli resimlerde en sık kullanılan dönüşümlerden biri olup, frekans alanında en uygun dağılımı sağladığından günümüzde sıkça kullanılmaktadır. Genellikle JPEG ve MPEG formatına uygulanır [20]. Ayrık Kosinüs Transfer Dönüşümünde ilk olarak görüntüyü 8 yatay ve dikey bloklara ayrılır ve böylece her blokta 64 piksel bulunur. Ard arda gelen 2 yatay ve dikey bloklarda makro bloğunu oluşturarak her bloğa Ayrık Kosinüs Dönüşümü uygulanır. Kodlama da özel bir formül yardımıyla resmin yatay ve dikey renk değerleri bulunup görüntüdeki en yüksek ve en düşük frekans noktaları saptanır [21,22]. Bu noktaların saptanmasının temelinde insan gözünün yüksek frekansa daha az hassasiyeti olduğu için bu değerleri azaltmak gerekirse 0’a düşürmek yatmaktadır. Blok değerlerinin 0’a yaklaştırılması piksel bloklarının farklı kalite değerinde olmasına sebebiyet vereceğinden kalite değerlerinin ortalaması alınarak birbirine yaklaştırılır ve bu 64 nokta bir dizi şeklinde sıralanarak kayıpsız sıkıştırma uygulanarak Ayrık Kosinüs Transfer Dönüşümü tamamlanır [21,23]. Literatüre ilk defa 1974 yılında N.Ahmed, T.Natarajan, K.R.Rao tarafından kazandırılmıştır [24]. Fourier Dönüşüm ile kıyaslandığında yaklaşık iki kat kadar kosinüs dönüşümün hızlı olduğu 1978 yılında M. Narasimha ve A.Peterson tarafından ispatlanmıştır [25]. Görüntü sıkıştırmaya ise

1988 yılında Yukihiro ARAI, Takeshi AGUI ve Masayuki Nakajima tarafından uygulanmıştır ve 1992 yılında E.Feig ve S.Winograd tarafından Kosinüs Transfer Dönüşümü için hızlı algoritmalar oluşturulmuştur [26,27].

Bir diğer dönüşüm tekniği ise Dalgacık Dönüşümüdür. Dalgacık Dönüşümü görüntü sıkıştırmaya en çok uygulanan bazen de ses ve video sıkıştırma da kullanılan bir veri sıkıştırma formudur. JPEG 2000, DjVU, Cineform ve BBC Dirac kayda değer uygulamalarından bazılarıdır. Hedef ise olabildiğince küçük uzayda görüntü datasını saklamaktır. Dalgacık Dönüşümü ise burada ya kayıplıdır ya da kayıpsızdır ve JPEG 2000’de kayıpsız dönüşüm için 5/3 kayıplı dönüşüm için ise 9/7 dalgacık kullanılabilir [28,29]. Dalgacık dönüşümüyle görüntü sıkıştırmada ilk olarak görüntü bulunduğu boyuttan başka bir boyuta aktarılır. Orada katsayılarla kodlanır ve bu kodlama esnasında görüntü üzerindeki elemanlar arası korelasyon azalarak fazla bilginin kodlanmasından kaçınılır. Böylece görüntü enerjisinin büyük bölümünün küçük bir bölgede toplanıp bütün katsayılar kullanılmadan kodlama gerçekleştirilmiş olur. Ayrık Kosinüs Transfer Dönüşümüne göre en büyük avantajı yüksek sıkıştırma oranıyla birlikte saklama alanı ve band genişliği üzerinde tasarruf sağlamasıdır. Literatürde Ayrık Dalgacık Dönüşümü 1977 yılında ECG sinyallerinin sıkıştırılmasında kullanılmıştır [30]. Görüntü sıkıştırmada ise 1995 yılında Dalgacık Dönüşümü kullanarak çok ölçekli görüntüyü algılamayla ilgili çalışma yapılmıştır [31].

Kayıplı sıkıştırma tekniklerinden Alt Band Kodlaması, sinyal işlemede bir sinyali birden fazla frekans bantlara ayıran kod dönüştürme formudur ve her bir frekans bandını bağımsız olarak kodlar [32]. Alt band kodlamanın temelinde maskelenen frekansların, atılan bilgiyle veri azaltmayı sağlama yatmaktadır. Sonuçlar orijinal sinyalden farklıdır. Fakat atılan sinyal dikkatlice seçilirse, fark farkedilmeyebilir. Literatürde R.E.Crochiere 1980 yılında Alt Band Kodlaması ve uygulama alanlarına ait çalışma yapmıştır [33]. Alt Band Kodlaması görüntüye ilk defa J.Woods ve S.O’Neil tarafından 1986 yılında uygulanmıştır [34]. Takahiro Saito, Hirojiuni Higuchi ve Takashi Komatsu tarafından geliştirmiş permütasyon kodlarıyla görüntü sıkıştırmada kullanılmıştır [35].

Kesirsel Kodlama ise bir diğer kayıplı sıkıştırma tekniğidir. Kesirsel görüntü sıkıştırma, görüntü içerisinde benzer olan parçaların bulunması işlemini temel alır. Doğayla ilgili görüntülerin sıkıştırılmasında diğer yöntemlere göre daha iyi bir yöntem olup, aynı zamanda diğer yöntemlerin sıkıştırılmasında kullanılır. Benzer parçaların aranması oldukça fazla karşılaştırma sağladığından, görüntü üzerinde ele alınan parçaların ve bunları eşleştirilecek parçaların sınıflandırılarak, benzerliklerin bu sınıflar arasında aranması tercih edilir. Literatürde Michael Barnsley öncülüğünde 1987 yılında kesirsel sıkıştırma geliştirilip çeşitli alanlarda patenti alınmıştır [36]. En bilinen pratik Kesirsel Sıkıştırma Algoritması Barnsley ve Alan Sloan tarafından icat edilmiştir. Barnsley'in öğrencisi Arnaud Jacquin 1992 yılında ilk otomatik algoritmayı yazılıma uygulamıştır [37]. Tüm kesirsel sıkıştırma ile ilgili methodlar ötelenmiş fonksiyon sistemi kullanarak kesirsel dönüşüm temellidir. Kesirsel görüntü, temsili matematiksel olarak ötelenmiş fonksiyon sistemi (IFS) olarak tanımlanır. Bölmeli ötelenmiş fonksiyon sistemi (PIFS) ise yapı olarak aynı kodlara sahip olup IFS'den tek farkı kodları tüm görüntü yerine görüntünün bir kısmına uygulanıp elde etmesidir.

Kayıplı sıkıştırma tekniklerinden Vektör Kuantalama ise sayısal haberleşme sisteminde iletilecek bilginin kendisini en iyi şekilde temsil edecek daha az veri ile ifade edilmesi yöntemidir. Bu yöntemin temel amacı ise sürekli girdi uzayı için çıktı uzayında girdi uzayını yansıtmaktır. Bu tekniğin temelinde görüntünün olasılık yoğunluk fonksiyonu oluşturularak, en düşük bit sayısı ile görüntünün daha yüksek kalitede elde edilmesi yatmaktadır. Görüntüde her bir çıkış kod vektör, çıkış kümesi ise kod kitabıdır. Bu yöntemde başarılı sonuçlar elde edebilmek için görüntü içerisindeki blokların içindeki bilgilerin kod kitabı içerisindeki en yakın vektöre yaklaşması için en yakın komşuluk sağlanmalı ve kod kitabı içerisindeki vektörlerin uzayın merkezinde olması gerekmektedir [3]. İlk olarak 1984 yılında Gray, R.M tarafından literatüre kazandırılmıştır [38]. Daha sonraki yıllarda ise geliştirilerek görüntü sıkıştırmaya uygulanmıştır [39,40]. Bu uygulamaların farklı domainlerde çalışmaları da incelenmiştir [41]. Bu uygulamaların dışında Vektör Kuantalama konuşma kodlamaya da uyarlanmıştır [42].

Bu çalışmada literatürde ilk defa Karides Sürüsü Algoritması, Vektör Kuantalama Algoritmasının bir çeşidi olan Linde Buzo Gray Algoritması temel alınarak standart görüntüye uygulanmış ve genetik operatörler eklenerek görüntü sıkıştırma üzerinde 8, 16, 32, 64 kod kelimeli popülasyonlardaki ortalama karesel hata ve psnr değerlerindeki değişimler 100 iterasyonda gözlemlenmiştir. Bu çalışmada Karides Sürüsü Algoritmasının genetik operatörlerle çözüm uzayını daha fazla girmesi ve lokal minimumdan takılmadan global minimuma hızlı bir şekilde ulaşması hedeflenmiştir. Bu analize ek olarak probleme Karınca Kolonisi Algoritması (Genetik Operatörlü / Operatörsüz), Parçacık Sürüsü Algoritması ve Havuz Tabanlı Genetik Algoritma uygulanmıştır. Literatürde birçok problemin optimizasyonunda kullanılan bu algoritmalar LBG tabanlı görüntü sıkıştırmaya uygulanmış ve Karides Sürüsü Algoritmasında olduğu gibi bu algoritmalar arasındaki ortalama karesel hata ve psnr değişimi 8, 16, 32, 64 kod kelimeli popülasyonlarda grafiklerle 100 iterasyonda gözlenip sonuçları incelenmiştir. Böylece Karides Sürüsü Algoritmasının (Genetik Operatörlü / Operatörsüz) çeşitli sıkıştırma oranlarında diğer ek analizlerle performansları kıyaslanmış ve üstünlükleri görülmüştür.

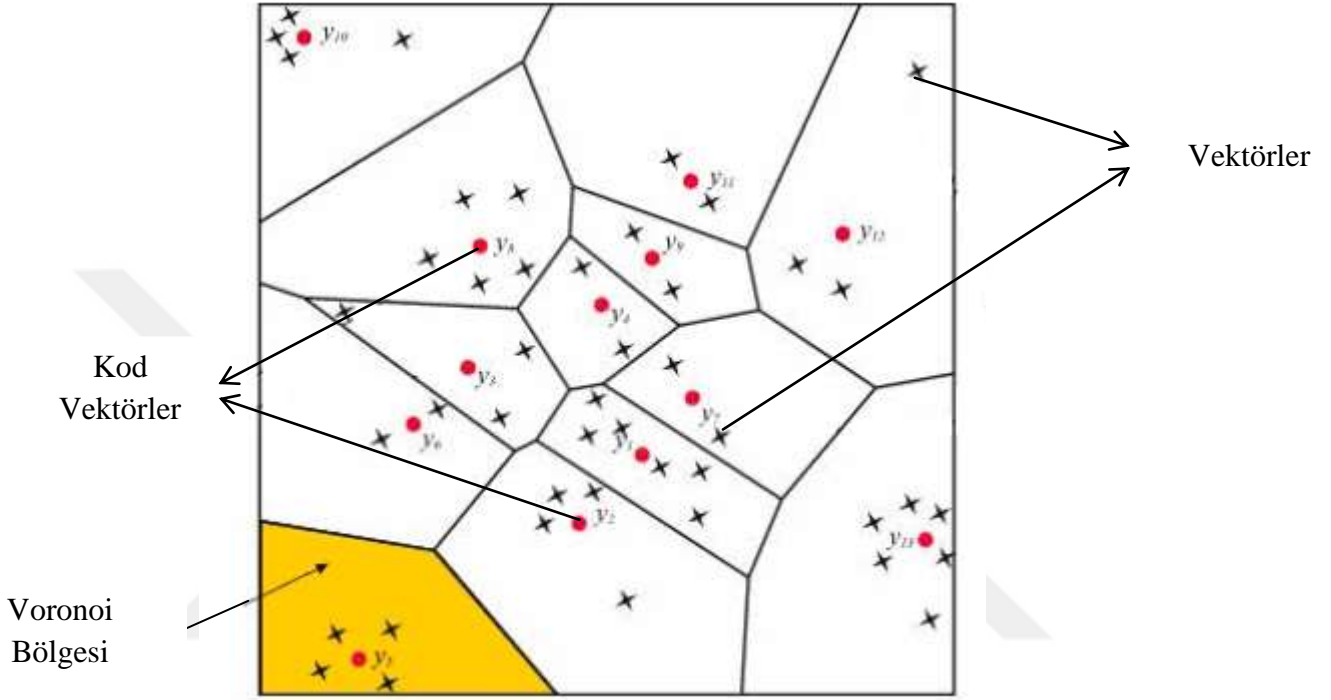
2. VEKTÖR KUANTALAMA

2.1. Linde Buzo Gray Algoritması

Vektör Kuantalama blok kodlama prensibine dayalı kayıplı veri sıkıştırma tekniğidir ve sabit uzunlukta bir algoritmadır. Daha önceki zamanlarda çok boyutlu birleşme için gerekli ihtiyaçtan dolayı vektör niceleyici zorlu bir problem olarak düşünülüyordu. Fakat 1980 yılında Linde, Buzo ve Gray eğitim dizisi temelli Vektör Kuantalama tasarım algoritması önerdiler. Eğitim dizisinin kullanımı çok boyutlu birleşme için ihtiyacı gidermiştir. Bu tekniği kullanarak tasarlanan vektör kuantalama literatürde LBG Vektör Kuantalama olarak geçmektedir ve bu teknik iyi kod kitabından türemiştir [43]. Veri kümeleme algoritmalarından K-means'e benzemektedir. Bu algoritma literatüre ilk olarak Lloyd Algoritmasının genişletilmiş olarak girmiş olup [44], daha sonra görüntü vektör kuantalamada daha hızlı ve genişletilmiş versiyonları geliştirilmiştir [45,46]. LBG algoritması en yakın komşuluk ve merkezilik şartı gibi iki en iyilik kriterlerini çözümlediği için aynı zamanda görüntü sıkıştırma kullanılmıştır [47].

Linde Buzo Gray Algoritmasında 2^n vektör giriş verileri ortalama karesel hata kriteri ile daha az vektörle temsil edilir. İlk vektör veri havuzundaki vektörlerin aritmetik ortalaması olup bu ilk vektöre küçük değerler eklenerek vektör sayısı ikiye çıkarılır ve ilk vektör değerini bu yeni elde edilen iki vektör değerine bırakır. LBG'in her bir adımında bu iki vektöre en yakın vektörler ortalama karesel hataya bağlı olarak ortalamalarını yeni değerler üreterek günceller. Her adım bölünerek çoğalan yeni vektörlerin değerleri değişmeyinceye kadar devam eder ve bu veri havuzundaki vektör sayısı 2^n olacak şekilde artırılabilir. Vektörlerin sayısının azami sınırını belirleyecek olan veri havuzundaki vektörlerin sayısıdır. Algoritmanın adımlarında oluşturulan boş vektörler vektör listesini vektör listesinin boyutunu daha fazla artıramayacağımızı gösterir [48].

LBG algoritması bir grup noktanın tek bir noktaya uzaklıkları toplamının minimum olması için o noktanın grupların merkezinde olmasına dayandığından dolayı vektör kuantalamadaki amaç vektör uzayındaki vektörleri N adet vektörle temsil etmektir.



Şekil 2.1. Örnek Vektör Kuantalama Alanları [49]

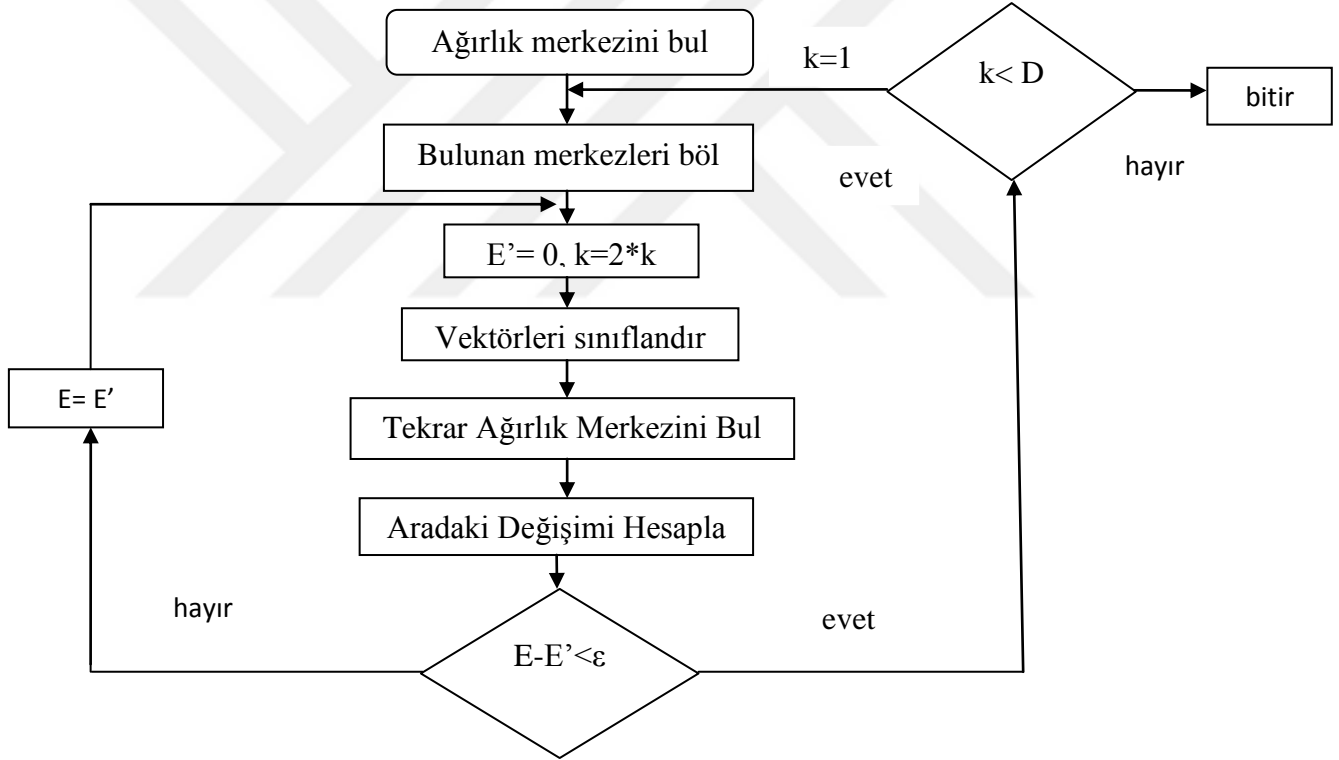
Voronoi Alanı, düzlemdeki noktalar kümesindeki her bir noktaya, kümedeki diğer noktalardan daha yakın bulunan noktaların geometrik yerine denir [50].

LBG algoritmasında ilk olarak tüm vektörleri temsil eden kod vektörler bulunup istenen kod vektör sayısına kadar vektörler gruplanır [49].

Algoritmanın akışı ise;

1. Bir boyutlu vektör kod kitabı oluşturup vektörlerin tamamının ağırlık merkezi hesaplanır.

2. Her bir kod kitabı $X_n^+ = X_n (1+\epsilon)$ ve $X_n^- = X_n (1-\epsilon)$ kuralına göre ikiye ayrılır. (n değeri 1 den istenen değere kadar ve ϵ ise 0,01-0,05 arasında olup bölme parametresidir) Her bir vektör öklid uzaklığına bağlı olarak en yakınındaki kod vektörün etrafında gruplanır.
3. Ağırlık merkezi yeniden hesaplanır ve atanan yeni vektörlere göre ağırlık merkezi güncellenir.
4. İstenen sayıda kod vektörü elde edilip ağırlık merkezi değişmeyinceye kadar adımlar tekrarlanır [51,52].



Şekil 2.2. LBG Algoritması Akış Diyagramı [51]

Bu çalışmada ise LBG tekniği kullanılarak Karides Sürüsü Algoritması görüntü sıkıştırma problemine ilk defa uygulanıp test edilmiştir.

3. METASEZGİSEL ALGORİTMALAR

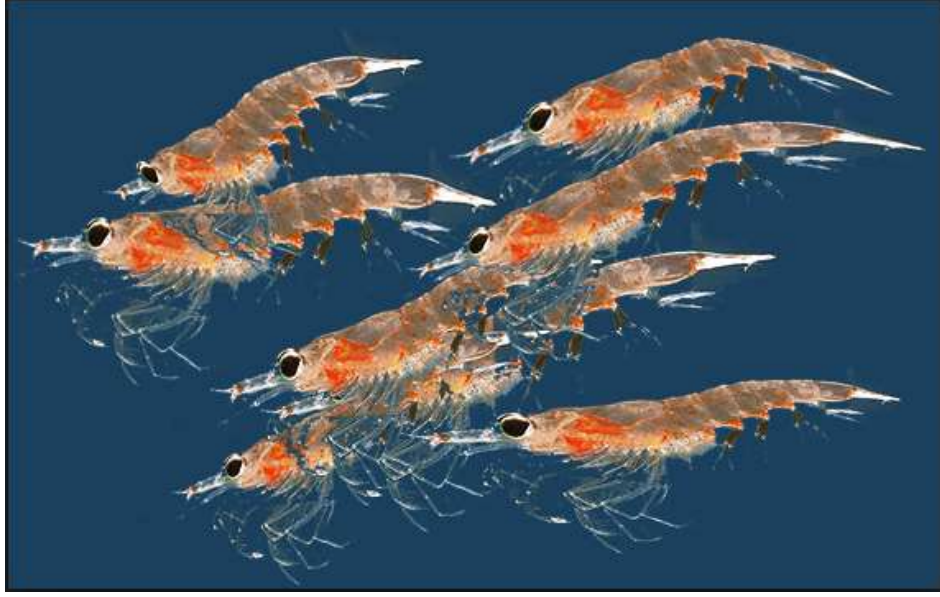
Metasezgisel Algoritmalar, herhangi bir amacı gerçekleştirmek veya hedefe varmak için çeşitli alternatif hareketlerden etkili olanlara karar vermek amacı ile tanımlanan kriterler veya metodlardır [53].

Metasezgiseller, arama sürecine rehberlik yapan stratejilerdir. Amaç, en iyi ya da en iyiye yakın çözümü bulmak için arama uzayını hızlıca araştırmaktır. Kısacası, metasezgiseller, farklı metodlarla arama uzayının araştırılması için yüksek seviye taktiklerdir [54].

Bu çalışmada da görüntü sıkıştırma problemine çeşitli metasezgisel algoritmaların performansları test edilmiştir.

3.1. Karides Sürüsü Algoritması

Karides sürüsü bireyleri beslenme, avcılardan korunma, çevresel şartlara hızlı bir şekilde uyum sağlayıp çoğalma konusunda önemli bir kabiliyete sahiptir [55,56]. Deniz hayvanları arasında en çok araştırılan deniz hayvanlarından biri Antartik karidesleridir. Bu karideslerin araştırılmasının sebebi ise kısa zaman dilimi içinde hızlı bir şekilde büyük sürüler oluşturabilmesidir [57,58]. Ancak karides sürüsü popülasyonunun dağılmasına neden olan birçok etken bulunmaktadır [59]. Karides sürüsünün doğadaki formunu anlamak için önerilen bazı kavramsal modeller vardır ve bu modellerden elde edilen sonuçlar karides bireylerinin temel birimlerden oluştuğunu gösterir [60,61].



Şekil 3.1. Antarktik karidesleri [62]

Karides bireyleri avcılarının her saldırmasının ardından karides sürü yoğunluğunu azaltma yönünde hareket eder. Bu saldırı sonucunda ise karides sürüsü karides yoğunluğunu artırmak ve yeme ulaşmak için tekrar oluşur. Bu amaçlar doğrultusunda karidesler olası en iyi çözümün olduğu noktaya yönelirler [56].

Karides Sürüsü Algoritması Gandomi ve Allavi tarafından 2012 yılında literature kazandırılmış olup birçok farklı problemin çözümünde kullanılmıştır [56]. Bu algoritma Gandom ve Allavi tarafından 2012 yılında ilk olarak lineer olmayan test fonksiyonları üzerine uygulanmıştır. 2013 yılında ise Varinder Singh ve Mrs. Monica Sood, karides sürüsü temelli bir kümeleme algoritması ortaya koymuşlardır [63]. Gai-Ge Wang ve Lihong Guo 2013 yılında global optimizasyon problemi için benzetilmiş tavlama temelli Karides Sürüsü Algoritması geliştirmişlerdir [64]. Gai-Ge Wang ve Lihong Guo global yakınsama hızını artırmak için Karides Sürüsü Algoritmasına kaos teorisi ekleyerek kaotik parçacıklı Karides Sürüsü Algoritmasını oluşturmuşlardır [65] ve aynı zamanda sınırlı çalışma dilimi içinde problem çözmek için algoritmaya Levy Flight eklemiş [66], global nümerik optimizasyon için yeni Karides Sürüsü Algoritması geliştirmişlerdir [67]. Gai-Ge Wang tarafından tekrar 2014 yılında yeni değişkenler ekleyerek Stud Karides Algoritması literature kazandırılmıştır [68]. Junpeng Li ve Yinggan Tang lineer azalan adımla birlikte

geliştirilen Karides Sürüsü Algoritması elde etmiş ve optimizasyon çalışmasındaki çalışma ve araştırma arasındaki dengeyi sağlamıştır [69].

Karides Sürüsü Algoritması birçok problemin çözümünde başarılı olduğu literatürdeki analizlerle gözlemlenmiştir. Fakat görüntü sıkıştırma problemine henüz uygulanmamıştır. Bu çalışmada bu algoritmayı LGB tekniğini kullanarak ilk defa görüntü sıkıştırmaya uygulayacağız.

3.1.1. Karides Sürüsü Algoritmasının Temel Yapısı

Karides Sürüsü Algoritmasında ilk durum, sürünün diğer canlılar tarafından avlandığını ve karides yoğunluğunun azalarak yem kaynağından uzaklaşmasıyla başlar. Doğal sistemde, her bireyin uygunluğu karidesin yoğunluğu ve yem bulunan noktalara uzaklığın birleşimiyle gerçekleşir. Böylece uygunluk, amaç fonksiyonunun değeri olmaktadır. Karides bireylerinin zamana bağlı durumları 3 temel eylemle gerçekleşmektedir.

- Diğer karideslerin neden olduğu hareket
- Yiyecek arama hareketi
- Random yayılma,

n boyutlu karar uzayında Lagrange modeli,

$$\frac{dx_i}{dt} = Ni + Fi + Di \quad (3.1)$$

Burada Ni diğer karideslerin neden olduğu hareket, Fi yiyeceği yemi arama hareketi, Di ise i . karidesin random yayılmasıdır [61].

3.1.2. Diğer Karideslerin Neden Olduğu Hareket

Her bir karides bireyi için bu hareket aşağıdaki şekildedir.

$$N_i^{\text{yeni}} = N^{\text{maks}} a_i + w_n N_i^{\text{eski}} \quad (3.2)$$

$$a_i = a_i^{\text{lokal}} + a_i^{\text{hedef}} \quad (3.3)$$

N^{maks} maksimum neden olunan hız, 0.01 (m/s) olarak alınmıştır. a_i neden olunan hareketin doğrultusu, w_n [0,1] aralığında neden olunan hareketin atalet ağırlığı, N_i^{eski} son neden olunan hareket, a_i^{lokal} komşu karideslerin sağladığı lokal etkiler, a_i^{hedef} en iyi karides tarafından sağlanan hedef yönün etkisidir. Karidesin tekilliğinden uzaklaşmak için küçük pozitif değer ε paydaya eklenmiştir ve 0,001 olarak alınmıştır. Karidesin hareketine komşu karidesin etkisi şu şekilde formülize edilir [70].

$$a_i^{\text{lokal}} = \sum_{j=1}^{NN} K_{i,j} X_{i,j} \quad (3.4)$$

$$X_{i,j} = \frac{x_j - x_i}{\|x_j - x_i\| + \varepsilon} \quad (3.5)$$

$$K_{i,j} = \frac{K_i - K_j}{K_{\text{worst}} - K_{\text{best}}} \quad (3.6)$$

Burada K_{worst} ve K_{best} karides bireylerinin en iyi ve en kötü uygunluk değeri olup K_i i. karides bireyinin amaç fonksiyon değerini gösterirken, K_j j. komşu bireyin amaç fonksiyonu değeridir. Her karides bireyinin fonksiyonu X iken, NN toplam komşu sayısını göstermektedir. Komşu seçimi ise hissedilen uzaklık temel alınmıştır.

$$d_{s,i} = \frac{1}{5N} \sum_{j=1}^N \|X_i - X_j\| \quad (3.7)$$

$d_{s,i}$ karides bireyinin hissedilen uzaklığı olup N toplam karides sayısını gösterir. Denklem göre eğer iki karides bireyi arasındaki uzaklık d_s den küçükse, karideslerin komşu olduğu çıkartılmaktadır. En iyi amaç fonksiyonuna sahip karidesin i . karides üzerindeki etkisi şu şekildedir.

$$a_i^{\text{hedef}} = C^{\text{best}} \cdot K_{i,\text{best}} \cdot X_{i,\text{best}} \quad (3.8)$$

C^{best} en iyi etki katsayısıdır.

$$C^{\text{best}} = 2 \left(\text{rand} + \frac{I}{I_{\text{maks}}} \right) \quad (3.9)$$

I iterasyon sayısı, I_{maks} maksimum iterasyon sayısı olup rand değeri ise 0 ile 1 aralığındadır.

3.1.3. Yiyecek Arama Hareketi

Bu temel adımda iki temel adım önemli olup bunlardan ilki yiyeceği yemin yeri bir diğeri ise daha önceki adımlarından elde ettiği tecrübedir.

$$F_i = V_f \cdot \beta_i + w_f \cdot F_i^{\text{eski}} \quad (3.10)$$

$$\beta_i = \beta_i^{\text{yem}} + \beta_i^{\text{best}} \quad (3.11)$$

V_f yem arama hızıdır ve Price'ın çalışmasına dayanarak 0.02 (m/s) olarak alınmıştır. w_f yem arama hareketinin [0,1] aralığında ataleti ağırlığı, F_i^{eski} son yem arama hareketi, β_i^{yem} yem çekiciliğini ve β_i^{best} i . karidesin şu zamana kadar ki en iyi amaç fonksiyonunun etkisidir. Her iterasyonda yiyecek merkezi şu şekilde güncellenir.

$$X^{\text{yem}} = \frac{\sum_{i=1}^N (1/K_i) X_i}{\sum_{i=1}^N 1/K_i} \quad (3.12)$$

i . karides bireyi için yemin çekiciliği ise

$$\beta_i^{\text{yem}} = C^{\text{yem}} K_{i,\text{yem}} X_{i,\text{yem}} \quad C^{\text{yem}} = 2 \left(1 - \frac{I}{I_{\text{max}}}\right) \quad (3.13)$$

i . karides bireyine ait en iyi amaç fonksiyonu modeli aşağıdaki gibi formülize edilmiştir.

$$\beta_i^{\text{best}} = K_{i,\text{best}} \cdot X_{i,\text{best}} \quad (3.14)$$

$K_{i,\text{best}}$ önceki gidilmiş en iyi pozisyon değeridir.

3.1.4. Random Yayılma

Fiziksel yayılma random gerçekleştiği için aşağıdaki şekilde formülize edilmiştir.

$$D_i = D_{\text{maks}} \left(1 - \frac{I}{I_{\text{maks}}}\right) \delta \quad (3.15)$$

D_{maks} [0.002, 0.01] (m/s) aralığında maksimum yayılım hızı, δ ise [-1,1] aralığında random yönlü vektördür. Bu çalışmada δ değeri 0,000001 olarak alınmıştır.

3.1.5. Karides Sürüsünün Hareketini Tamamlama Süreci

Her bir karides bireyinin t ve $t + \Delta t$ zaman aralığında konum değişikliği vektörü aşağıdaki şekilde modellenmiştir.

$$X_i(t + \Delta t) = X_i(t) + \Delta t \frac{dX_i}{dt} \quad \Delta t = C_t \sum_{j=1}^{NV} UB_j - LB_j \quad (3.16)$$

NV (number of variable) toplam değişken sayısını, LB_j ve UB_j j. değişkene ait alt ve üst limitleri, C_t değeri gösterir. Δt değeri ise sabit olup 0.001 olarak alınmıştır.

3.1.6. Karides Sürüsü Algoritmasının Görüntüye Uygulanması

Karides Sürüsü Algoritmasının akışı;

Adım 1: Görüntü 4x4'lük bloklara ayırarak 4096 adet vektörü oluşturulur. LGB tekniği kullanarak kod kitapları oluşturulur.

Adım 2: Kod kitaplarıyla oluşturulan karideslerin ortalama karesel hata değerleri bulunup belirlenen eşik değeri altındakiler atılır.

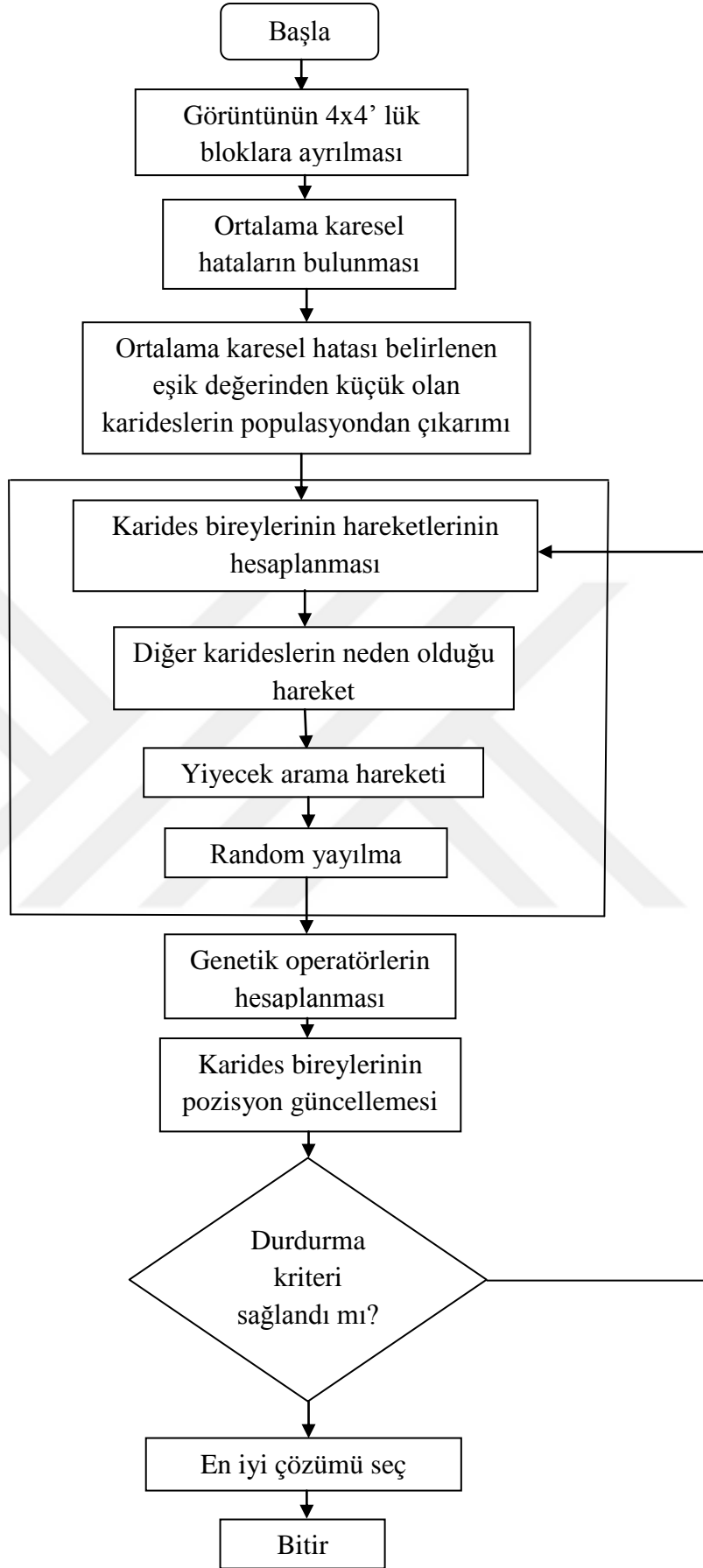
Adım 3: Karides bireylerinin zamana bağlı 3 temel adımları hesaplanır.

Adım 4: Genetik operatörler sırasıyla önce %10 çaprazlama daha sonra %4 mutasyon karides bireyelerine uygulanır.

Adım 5: Karides bireylerinin pozisyonları güncellenir ve durdurma şartı sağlanıncaya kadar karidesler hareketlerine devam ederler.

Adım 6: Her iterasyon sonucu istenen sonuca yakınsayıp yakınsamadığı kontrol edilir ve optimal sonuç bulunup algoritma bitirilir.





Şekil 3.2. Önerilen Çözüm Yönteminin Akış Diyagramı

3.2. Parçacık Sürüsü Algoritması

Kuş veya balık sürülerinin sosyal davranışlarından esinlenen popülasyon temelli sürü zekâsı algoritması olarak Eberhart ve Kennedy tarafından 1995'te geliştirilmiştir [71].

Eberhart ve Kennedy tarafından literatüre kazandırılan bu algoritma, ilk olarak kuş ya da balık sürüsündeki organizmaların hareketlerinin biçimlendirilmiş şekli olarak sosyal davranışı simüle edebilmek için ortaya çıkmıştır [73]. Algoritma basitleştirilerek optimizasyon problemlerine uyarlanmıştır. [74] kitabında da PSO ve sürü zekasının birçok olgusunu incelenmiştir. 2007 ve 2008 yıllarında R.Poli tarafından genişletilmiş çalışmalar yapılmıştır [75,76]. Son zamanlarda ise PSO tek hedefli sürekli uzay problemlerin çözümünde uygulanmıştır [77]. Kesirsel görüntü sıkıştırma olarak 2011 yılında K.Uma tarafından PSO ve çeşitli metasezgiseller karşılaştırmalı olarak çalışılmıştır [78].

Parçacık Sürüsü Algoritmasında kuş sürüleri yiyeceğin yerini bilmemelerine rağmen her adımda yiyeceğe ne kadar uzaklıkta olduklarını öğrenmektedirler. Bu öğrenme işlemi ise yiyeceğe en yakın kuşu takip etme işlemi ile gerçekleşir. Parçacık Sürüsü Algoritmasında her bir kuş, parçacığı; kuş topluluğu ise sürüyü temsil etmektedir. Parçacığın her hareketinde yiyeceğe ne kadar uzaklıkta olduğu hesaplanır. Parçacıklar şimdiye kadarki en iyi uygunluk değerini, bu değerdeki konumunu aynı zamanda her hareketinde çözüm uzayındaki konum ve hedefe ulaşma hızı hafıza da tutulmalıdır. Çünkü parçacıkların her adımda çözüm uzayında hızının ve konumunun nasıl değişeceği komşu ve parçacığın en iyi konumlarının birleşmesiyle elde edilecektir [72].

3.2.1. Parçacık Sürüsü Algoritmasının İşleyişi

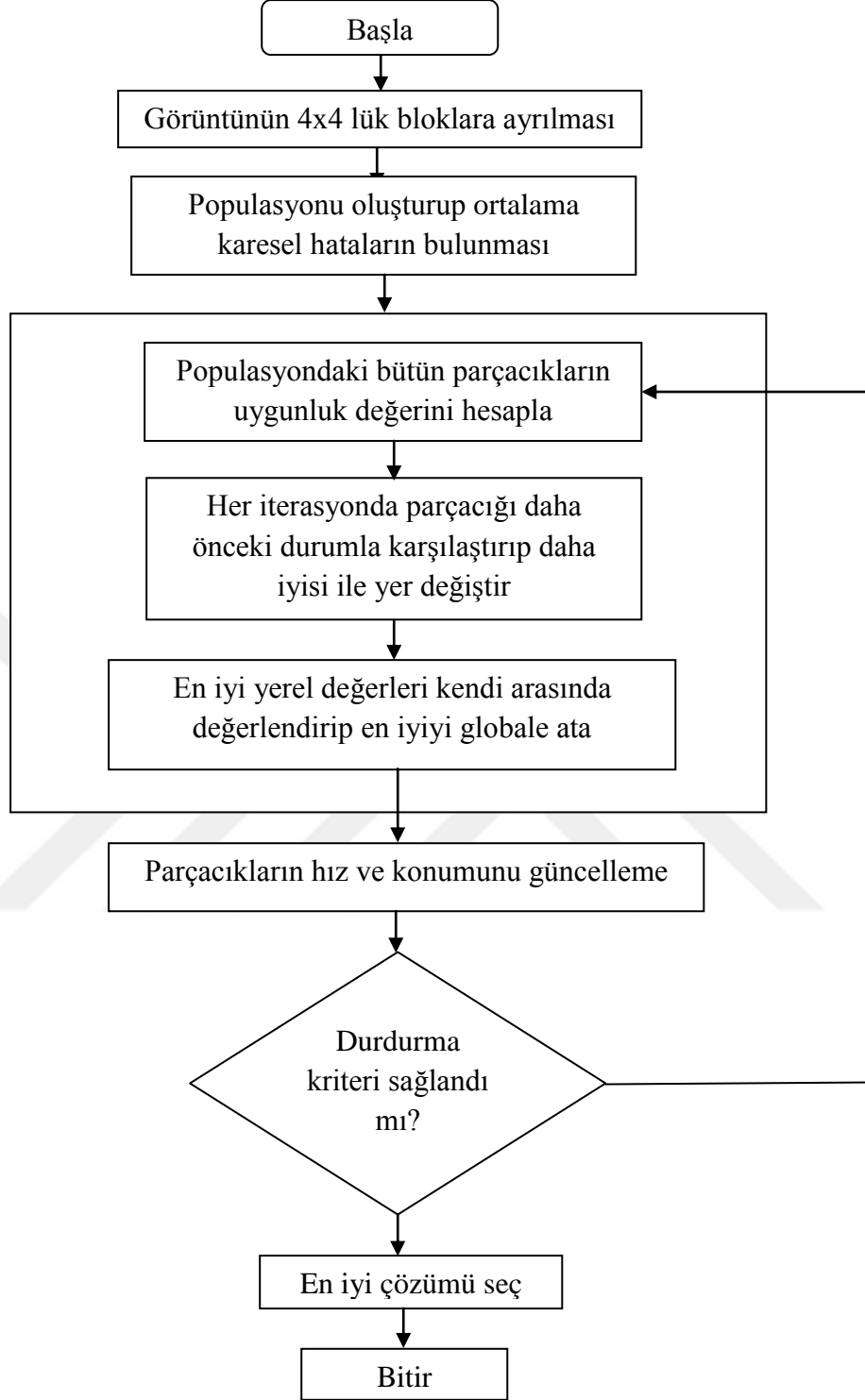
Random hareket eden parçacıklar, çözüm uzayında her adımda en iyi değeriyle güncellenir ve parçacığın o adıma kadar ki en iyi uygunluk değeri $pbest$, sürünün o adıma kadar ki en iyi uygunluk değeri $gbest$ 'tir. Bu iki değere bağlı olarak parçacığın hız ve konumu belirlenir.

$$V_i^{k+1} = V_i^k + c_1 \cdot rand_1^k \cdot (pbest_i^k - x_i^k) + c_2 \cdot rand_2^k \cdot (gbest^k - x_i^k) \quad (3.17)$$

$$x_i^{k+1} = x_i^k + V_i^{k+1} \quad (3.18)$$

rand 0 ile 1 arasındaki rastgele bir deęer olup, k iterasyon sayısını, c_1 ve c_2 öğrenme faktörüdür. Öğrenme faktörü deęerleri önemli parametreler olup uygun deęerler seçilmemesi durumunda çözüm uzayında hedefe ulaşamamaya sebebiyet verebilir. V_i deęeri ise parçacığın i. konumdaki hız deęeri olup başlangıçta 0 kabul edilir.





Şekil 3.3. PSO Akış Diyagramı

4x4'lük bloklara ayrılan görüntüde LBG tabanlı olarak parçacıklar oluşturulur ve bu parçacıklara algoritmanın temel adımları uygulanır. İstenen iterasyon sayısına ulaşıncaya kadar algoritma çalıştırılarak optimum sonuçla bitirilir.

3.3. Karınca Kolonisi Algoritması

Karınca Kolonisi Algoritması, bilgisayar bilimi ve yöneylem arařtırmalarında kullanılan, olasılığa baęlı grafiklerdeki iyi yörüngelerin bulunması için hesaplama probleminin azalmasını saęlayan bir tekniktir. Karıncalardan esinlenen bu algoritma ilk olarak 1992 yılında Marco Dorigo tarafından literatüre kazandırılmıştır [79,80]. Yemin kaynaęı ile koloni arasındaki yörüngeyi arařtıran karıncaların davranışını temel alan bu algoritma, ilk olarak sonlu elemanlı grafikteki en iyi yörüngeyi bulmayı hedeflemiştir. Temel fikir nümerik matematiksel problemleri çözmek için çeşitlendirilmiş, birçok problem ve problem çözümünde birçok karınca davranış çeşitleri ortaya çıkmıştır.

Karınca Kolonisi Algoritması gelişmeye açık son zamanlarda ortaya çıkan optimizasyon methodlarından biridir ve doğadaki karıncaların davranışlarından özellikle feromon salgısıyla karıncalara arasındaki dolaylı iletişimden esinlenilmiş bir algoritmadır [81]. Yapay zeka camiası arasında karınca kolonisi, yapay zeka kategorisi altında incelenmiştir [82]. Karınca Kolonisi Algoritması aynı zamanda çeşitli metasezgisel algoritmalarla birleştirilerek çeşitli optimizasyon problemlerinin çözümünde kullanılmıştır. 2009 yılında Benzetimli Tavlama Yöntemi ve Karınca Kolonisi Algoritması arasında hibrit bir çalışma yapılarak gezgin satıcı problemine uygulanmıştır [83]. 2011 yılında ise tabu arama algoritması ile Karınca Kolonisi Algoritması birleştirilerek süpermarket dağıtım problem üzerinde çalışılmıştır [84]. Bu algoritma aynı zamanda yerel arama algoritması ile birleştirilerek 2006 yılında ikinci dereceden atama problemi üzerinde uygulanmıştır [85]. Karınca Kolonisi Algoritması bu gibi birçok problem iyileştirmede kullanıldığı gibi kenar özellikli kesirsel görüntü sıkıştırma üzerine de uygulanmıştır [86].

Karınca Kolonisi Algoritmasının literatürde birçok varyasyonları bulunmaktadır. Bunlardan en yaygın olanları Elistist Karınca Sistemi, Max-Min Karınca Sistemi, Aşama Temelli Karınca Sistemi, Sürekli Ortogonal Karınca Sistemi, Özyinelemeli Karınca Kolonisi Algoritması'dır.

3.3.1. Karınca Kolonisi Algoritmasının İşleyişi

Karınca sürüsünün yuvası ile yemin bulunduğu nokta arasındaki en kısa mesafeyi bulma özelliğine bağlı olarak ortaya çıkan bu algorithmada, karıncalar alternatif yollar olması durumunda yollara eşit olarak dağılarak zamanla en kısa yolu, yol üzerindeki feromon aracılığı ile bulurlar. Görme duyuları olmayan karıncalar kısa yol üzerindeki feromon miktarı daha çok olduğundan o yolu tercih ederler.

Kısa yol üzerinde geçiş hızından dolayı birim zamanda geçen sayısı uzun yola göre daha fazladır. Bu yüzden yol ile feromon miktarı arasında ters orantı vardır. N tane noktadan $N-1$ yol ile yola başlanır ve her adımda noktaların ziyaretinden sonra yol sayısı azaltılıp feromon seviyesi güncellenir. Feromon seviyesi her adımda yol uzunluğu ve geçiş sayısına göre güncellenip yol seçimi buna göre gerçekleştirilir [87].

Yol tercihi belli olasılıkla iki şekilde gerçekleşir. Bunlardan ilki genellikle 0,9 olarak seçilen q_0 parametresi ile feromonun en yoğun olduğu yolun seçilmesi, ikincisi ise i . noktadaki karıncanın gideceği yol,

$$j = \max_{u \in J_k(i)} \{ [T(i, u)]^a \times [\eta(i, u)]^\beta \} \quad \text{eğer } q \leq q_0 \quad (3.19)$$

$T(i, j)$ i ve j noktaları arasındaki feromon, $\eta(i, j)$ seçilebilirlik parametresi i ve j nokta arasındaki tersi ($1 / \delta(i, j)$), a ve β ayarlanabilir parametreler, i . noktadaki karıncanın gideceği yol j yukarıdaki formülle modellenmiştir.

İkinci yol seçimi ise feromon seviyesine orantılı olarak seçmektir. Yol seçim olasılığı $1 - q_0$ 'dır ve $J_k(i)$ i . noktadaki karıncanın gideceği yolu gösterir ve gidilecek yolların olasılığı

$$p_k(i, j) = \frac{[T(i, j)]^a \times [\eta(i, j)]^\beta}{\sum_{u \in J_k(i)} [T(i, u)]^a \times [\eta(i, u)]^\beta} \quad (3.20)$$

Eğer $j \in J_k(i)$ ise yukarıdaki şekilde formülize edilir, diğer durumlarda 0'dır. Bu olasılık değerine göre hareket edilip feromon seviyesi yüksek olan yol seçilir.

Feromon güncellemeleri de iki şekilde gerçekleşir. Bunlardan ilki yol üzerindeki feromonların belli bir oranda buharlaşması, diğeri ise feromon seviyesini o yolu kullanan karıncaların yol uzunluğu ile ters orantılı olarak artırmaktır. Feromon seviyesi lokal ve global feromon seviyesinin toplamından oluşmaktadır.

Lokal feromon güncellemesi,

$T_{ij}(t)$, t iterasyonuna kadarki feromon miktarı, $\Delta T_{ij}^k(t+1)$ t iterasyonundaki feromon seviyesi ve ρ 0 ile 1 aralığında buharlaşma parametresi olmak üzere

$$T_{ij}(t+1) = (1-\rho)T_{ij}(t) + \sum_{k=1}^m \Delta T_{ij}^k(t+1) \quad (3.21)$$

i noktasındaki karınca (i,j) yolunu kullanmışsa

$$\Delta T_{ij}^k(t+1) = 1/L^k(t+1) \quad (3.22)$$

diğer durumlarda 0'dır. $L^k(t+1)$ karıncanın toplam tur uzunluğudur. Feromon seviyesine göre her adımda yol seçimi değişmekte ve bu şekilde kısa yollar bulmak hedeflenmektedir.

Global feromon güncellemesi,

Karıncaların geçerli adımdaki en iyi karıncanın feromon seviyesini takip etme şeklinde gerçekleşir ve feromon seviyesi ona göre artırılır. Belirli oranda en iyi sonuçlar diğeri adımlara aktarılır.

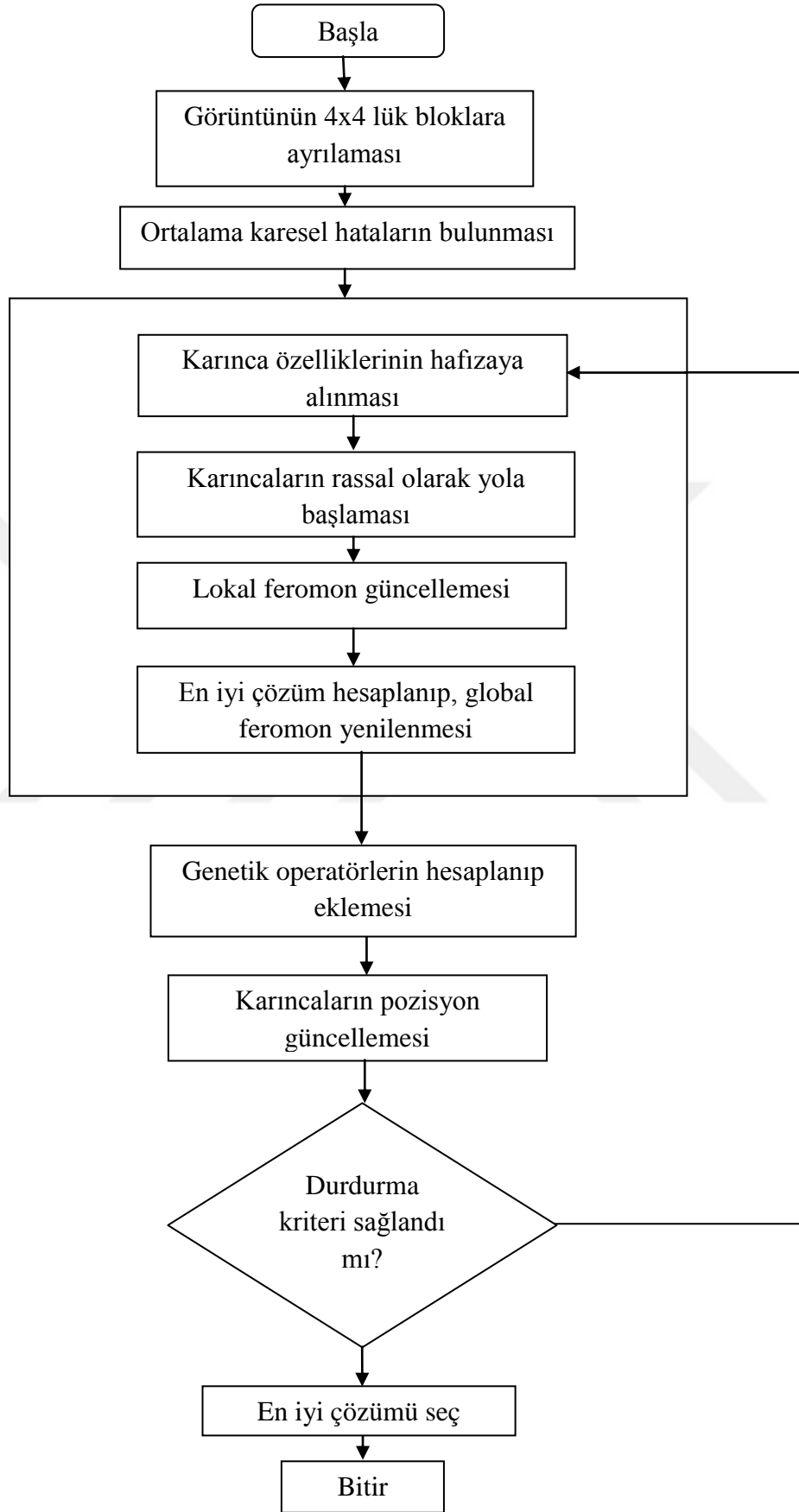
$$T_{ij}(t+1) = (1-\rho)T_{ij}(t) + \Delta T_{ij}^k(t+1) \quad (3.23)$$

Eğer (i, j) en iyi tura aitse

$$\Delta T_{ij}^{k_j}(t+1) = 1/L_{best}(t+1)'dir \quad (3.24)$$

Diğer durumlarda ise 0 dir. $L_{best}(t+1)$ geçerli adımdaki en iyi yol uzunluğudur. ρ , a , β temel parametreler olup, ρ buharlaşma, a feromon miktarının önemini, β ise yol uzunluğunun diğer adımlara etkisini gösteren parametredir. q_0 en iyi çözümün diğer adımlara etkisini aktaran parametredir.

Her bir karıncanın feromon güncellemesi ise lokal ve global feromon güncellemesinin birleşimiyle gerçekleşir.



Şekil 3.4. Karınca Kolonisi Algoritması Akış Diyagramı

4x4'lük bloklara ayrılan görüntüdeki 4096 adet kod kelimelerinden random olarak populasyon oluşturulur. Karıncalar random olarak harekete başlatılır. Her adımda belirli buharlaşma oranında lokal feromon seviyesi güncellenir ve populasyonun kod vektörleri bu feromon seviyesine göre güncellenir. Her adımda en iyi çözüm hesaplanıp global feromon seviyesi ve local feromon seviyesiyle birleştirilir. Populasyonun kod kelimeleri bu değere göre tekrar değiştirilir ve genetik operatörler sırasıyla %10 çaprazlama ve %4 mutasyon eklenir. Karıncalar populasyondan ayrılmaması için en iyi konum her adımda hafızada tutulur. Durdurma kriteri sağlanana kadar karıncalar hareketlerine devam ederler. Her adımda konumlar güncellenir ve en iyi çözüm seçilip algoritma tamamlanır.

3.4. Havuz Tabanlı Genetik Algoritma

Genetik algoritma yapay zekânın hızlı gelişen alanlarından olup, bir fonksiyonun global minimum ya da maksimum değerlerine en yakın sonuçları bulan bir metasezgisel algoritmadır. Evrimsel süreç kullanılarak, rastgele seçilen ebeveyn bireylerden yeni nesillerin türemesi ve bu yeni nesillerin en iyilerinin baş olarak kalıp en kötünün elenmesi mantığına dayanır. Yeni nesil oluşturma sürecinin tekrarlanması, belli bir eşik değeri miktarınca yada bir koşula bağlı olarak gerçekleşir.

Genetik algoritmalar probleme ilişkin tek bir çözüm yerine çözüm kümesi üzerinde çalışmaya yoğunlaşırlar. Genetik algoritmalar olasılık üzerine kurulu metodlar olup paralel çalışıp aynı anda problem uzayının birden çok noktasında arama çalışması yaparlar.

Genetik algoritma literatüre 1957 yılında Avustralyalı genetikçi Alex Fraser tarafından ölçülebilir özellikli çoklu loci kontrollü organizmaların yapay seçiliminin simülasyonu üzerinde çeşitli çalışmalar yapılarak kazandırılmıştır [88]. Bu çalışmalarla birlikte 1960 yılların başlarında biyologlar tarafından evrimin simülasyon methodları daha da yaygın olmuştur ve bu methodlar 1970'de Fraser ve Burnell ve 1973 'te ise Crosby tarafından kitaplarda tanıtılmıştır [89,90]. Fraser'ın simülasyonu şu anki genetik algoritmanın temel tüm elementlerini içermektedir.

Hans-Joachim Bremermann, Richard Friedberg, George Friedman ve Michael Conrad genetik algoritmanın öncülerindedir. 1998 yılında David B.Foegl tarafından bu çalışmalar tekrar kaleme alınmıştır [91]. Görüntü sıkıştırma problemine ise 1999 yılında Cheng Yimin, Wang Yixiao, Sun Qibin ve Sun Longxiang tarafından uygulanmıştır [92].

Havuz tabanlı genetik algoritma da ise literatürden farklı olarak yeni nesillerin türemesi en iyi ve diğerleri arasında gerçekleşmektedir. Böylece bireyler her adımda iyi gen havuzuna sahip bireyden bir gene sahip olup daha kötü gen değerlerine sahip olmayacaklardır. Bu da her adımda bireyin literatüre göre daha hızlı bir şekilde iyileşmesini sağlayacaktır.

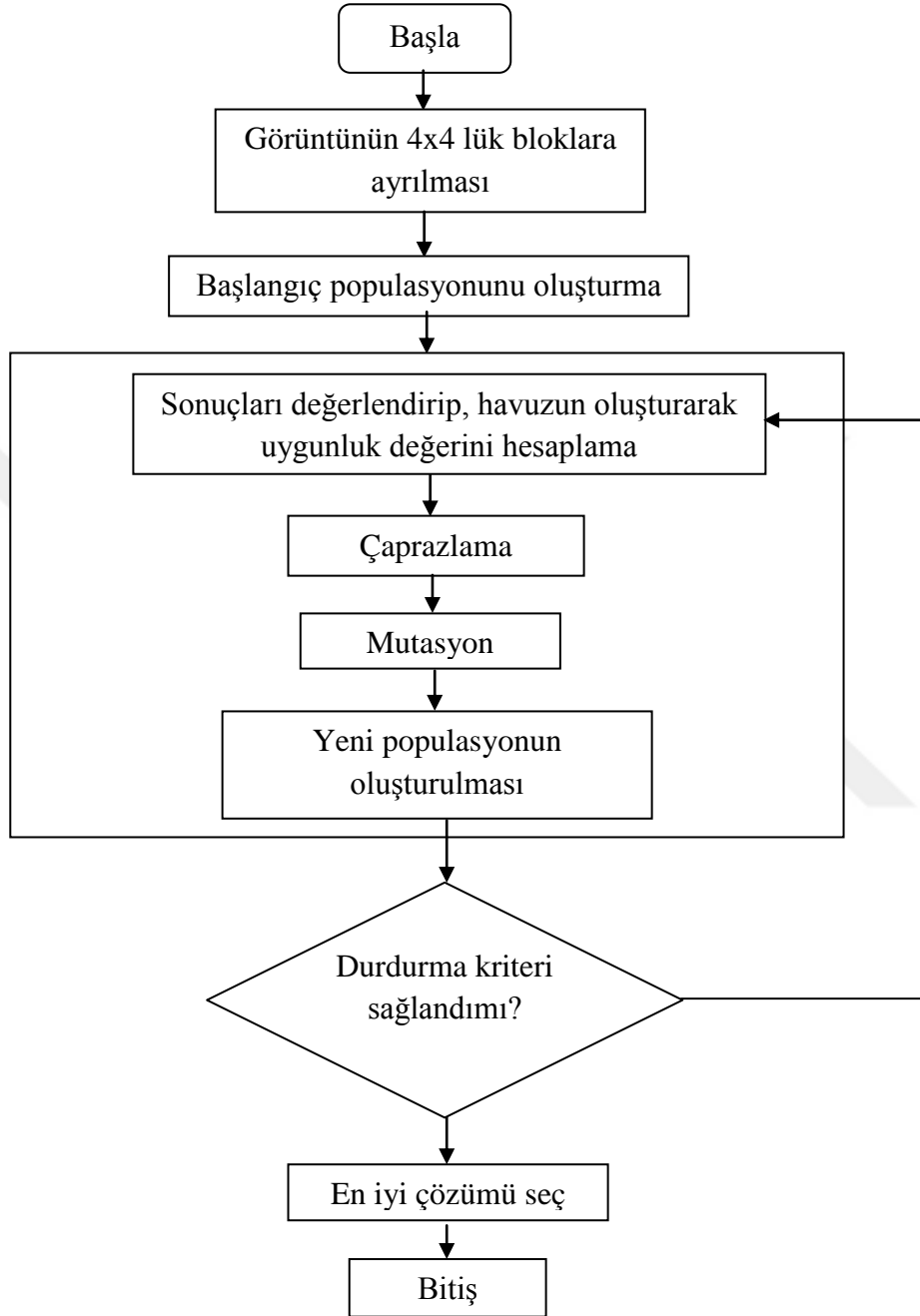
Havuz tabanlı genetik algoritma da ise bileşenler standart genetik algoritmadaki gibi gen, birey, popülasyon, uygunluk, çaprazlama ve mutasyondur.

Gen, algoritmadaki olası çözümün bir kaç bitlik bloklardır. Birey, olası bir çözümün yanında kromozom olarak da bilinir ve bu olası çözümlerin topluluğuna popülasyon denir. Uygunluk ise çözümün kalitesini belirler. Mutasyon ve çaprazlama operatörleri ise sırasıyla genetik çeşitliliği sağlamak ve benzer biçimdeki olası çözümlerin farklı bölümlerini birleştirerek daha iyi çözüme ulaşmak için gerçekleştirilir.

Birey seçim yöntemi standart genetikte çaprazlama için hangi yöntemle göre seçileceğini belirleyen parametre olup en çokları Rulet Seçimi Yöntemi, Sıralama Seçim Yöntemi, Turnuva Seçim Yöntemleridir.

Havuz tabanlı genetik algoritmada ise iki ayrı havuz bulunmaktadır. Bu havuzların birinde popülasyonun en az hata değerine sahip kod kitapları diğerinde ise popülasyondaki diğer kod kitapları bulunmaktadır. Çaprazlama bu iki havuz arasında gerçekleşmektedir ve çaprazlama oranı %10, mutasyon oranı olarak da %4 seçilmiştir.

3.4.1. Havuz Tabanlı Genetik Algoritmanın İşleyişi



Şekil3.5. Havuz Tabanlı Genetik Algoritması Akış Diyagramı

4x4'lük bloklara ayrılan standart görüntüden 4096 adet kod kelimeleri oluşturulur ve istenen popülasyon sayısına kod kelimeleri oluşturulur. Her bir bireyin LBG tabanlı olarak kod vektörleri güncellenip ortalama karesel hata değerleri bulunur. En az hata değerine sahip bireyler bir havuza, diğer bireyler diğer havuzda

toplanır. Bu bireylere standart genetik algoritmada olduđu gibi sırasıyla %10 aprazlama ve %4 mutasyon operatörü eklenir. Her adımda bireyler iki havuzda durdurma kriteri sađlanıncaya kadar işlemler devam eder. Kriter sađlandıktan sonra optimal sonuç seçilip algoritma bitirilir.



4. SONUÇLAR VE ÖNERİLER

Karides Sürüsü Algoritması, her bir bireyi 8, 16, 32, 64 kod kelimesine sahip 15 adet karideslere uygulanmış olup 100 iterasyon sonucunda genetik operatörlü ve operatörsüz sonuçları kıyaslandığında aşağıdaki grafikler elde edilmiştir. Tüm analizler MATLAB R2013b’de gerçekleştirilmiştir. Kıyaslama ölçütü olarak MSE ve PSNR kullanılmıştır. MSE elde edilen görüntü ile orjinal görüntü arasındaki hata değerlerini bulur, PSNR ise elde edilen ile orjinal görüntü arasındaki db cinsinden benzerliği ölçer. Aşağıdaki şekilde formülize edilmişlerdir.

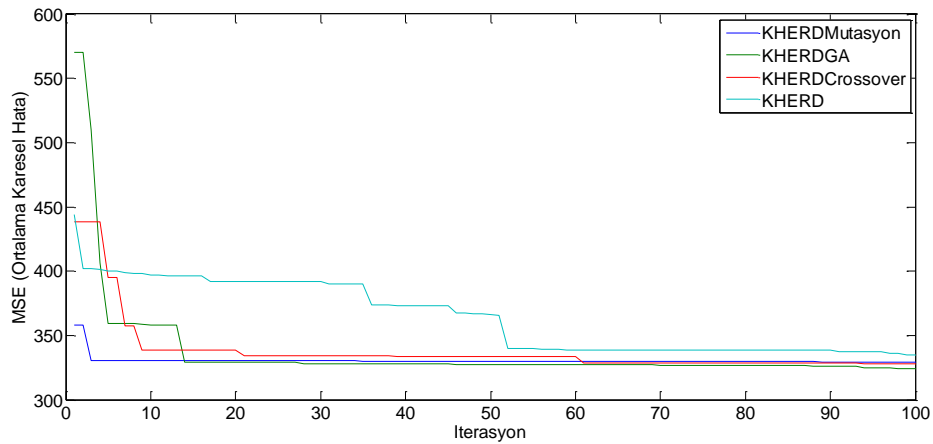
$$MSE = \sum_{i=1}^N (X_i - x_i)^2 / N \quad (4.1)$$

(X_i =Orjinal piksel, x_i = Tekrar elde edilen piksel, N= Toplam piksel sayısı)

$$PSNR = 10 \log_{10} [(\text{Max.Gray Level})^2 / MSE] \quad (4.2)$$

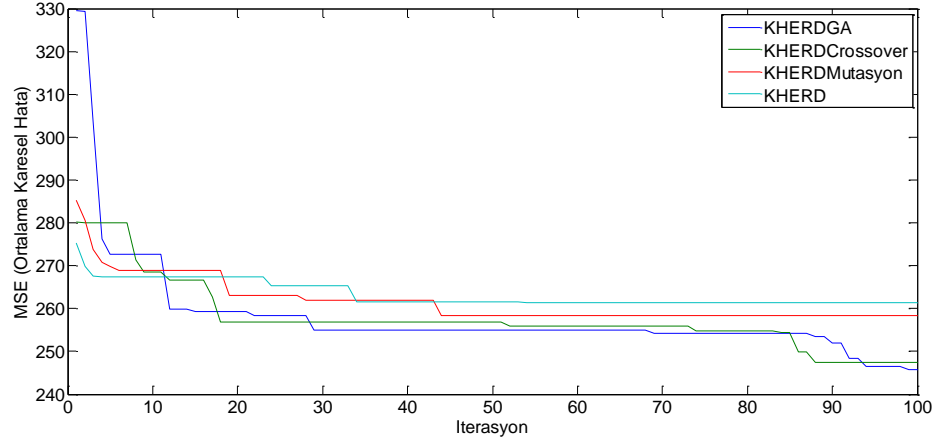
$$\text{Sıkıştırma Oranı} = \frac{\text{Codeword Sayısı} \times 16 \times 8 + 4096 \times \log_2 \text{Codeword Sayısı}}{256 \times 256 \times 8} \quad (\text{bpp}) \quad (4.3)$$

Karides Sürüsü Algoritması 8 kod kelimesine sahip 15 adet karidese uygulandığında MSE değişimi,



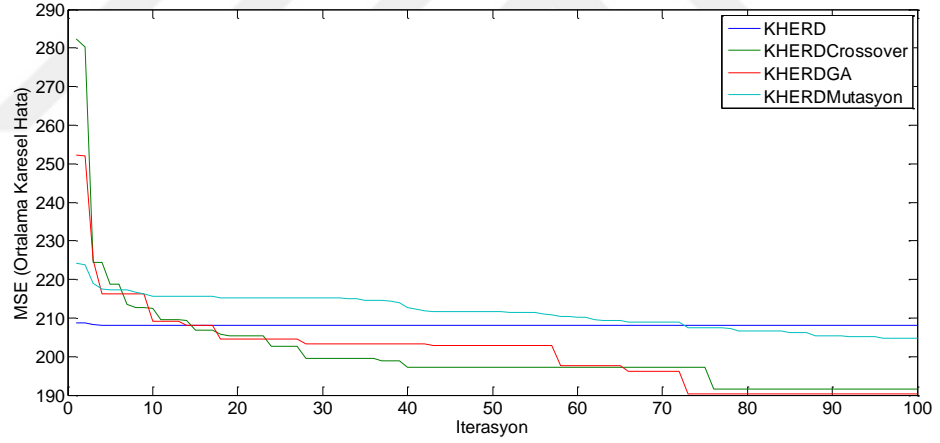
Şekil 4.1. 8 Kod Kelimesine Sahip Populasyona KSA Uygulanması Sonucu

16 kod kelimesine sahip ise,



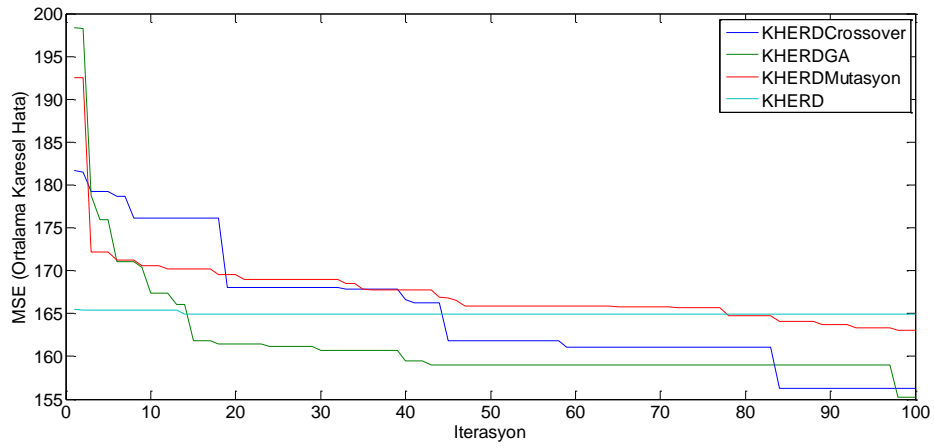
Şekil 4.2. 16 Kod Kelimesine Sahip Populasyona KSA Uygulanması Sonucu

32 kod kelimesine sahip ise,



Şekil 4.3. 32 Kod Kelimesine Sahip Populasyona KSA Uygulanması Sonucu

64 kod kelimesine sahipse,



Şekil 4.4. 64 Kod Kelimesine Sahip Populasyona KSA Uygulanması Sonucu

grafikleri elde edilir.

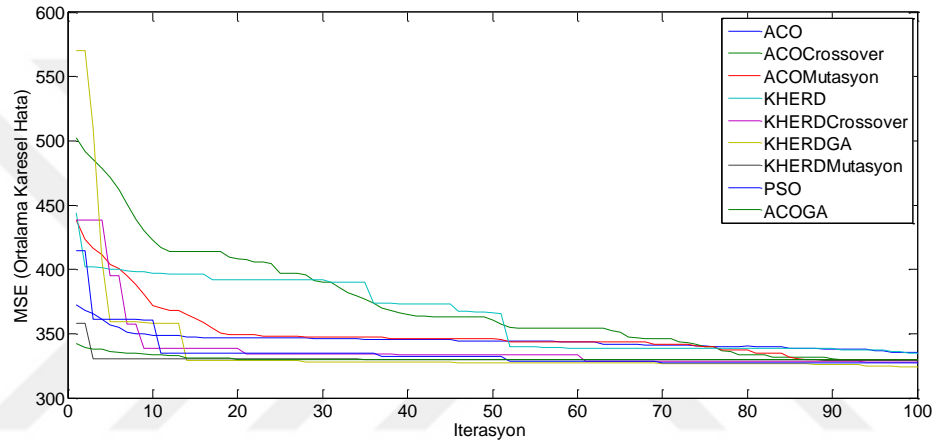
Yukarıdaki grafiklerden de gözlemlenebileceği gibi Genetik Operatörsüz Karides Sürüsü Algoritmasının Genetik Operatörlü Karides Sürüsü Algoritmasının gerisinde kaldığı 8, 16, 32, 64 kod kelimesine sahip bireylere uygulanan tüm çalışmalarda görülmüştür. 8, 16, 32 ve 64 kod kelimesine sahip bireylerde algoritma sonucunda Genetik Operatörlü KSA en iyi sonuca sahip olduğu, bu algoritmayı ise sırasıyla KSA Çaprazlama, KSA Mutasyon ve Genetik Operatörsüz KSA takip ettiği gözlemlenmiştir.

Bu çalışmada incelenen Karides Sürüsü Algoritması (Genetik Operatörlü/Operatörsüz), Karınca Kolonisi Algoritması (Genetik Operatörlü/Operatörsüz), Parçacık Sürüsü Algoritması, Havuz Tabanlı Genetik Algoritmaları 8, 16, 32, 64 kod kelimesine sahip 15 bireye sahip populasyona 100 iterasyonda uygulandığında aşağıdaki grafik ve tablolar elde edilmiştir.

8 kod kelimesine sahip bireylere uygulandığında MSE değişimi,

Tablo 4.1. Tüm Algoritmaların 8 Kod Kelimeli Populasyona Uygulanması

Lena	KHerd	Kherd Mutasyon	Kherd Crossover	Kherd GA	PSO	ACO	ACO Mutasyon	ACO Crossover	ACO GA
Codeword Sayısı	128 vektör	128 vektör	128 vektör	128 vektör	128 vektör	128 vektör	128 vektör	128 vektör	128 vektör
Sıkıştırma Oranı	0,0254 bpp	0,0254 bpp	0,0254 bpp	0,0254 bpp	0,0254 bpp	0,0254 bpp	0,0254 bpp	0,0254 bpp	0,0254 bpp
MSE	334,6887	329,205	327,9911	324,3981	327,2816	335,6641	329,9553	329,9266	329,9034
PSNR	22,8843 db	22,9561 db	22,9721 db	23,0200 db	22,9815 db	22,6204 db	22,9462 db	22,9466 Db	22,9469 db



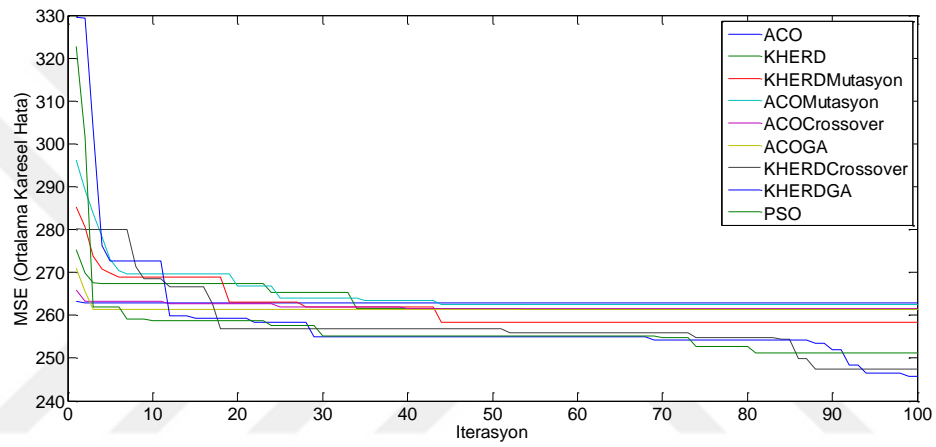
Şekil 4.5. 8 Kod Kelimesine Sahip Populasyona Tüm Algoritmaların Uygulanması Sonucu

KSA Genetik Operatörlü/Operatörsüz çalışmasına ek olarak, PSA, KKA, KKA Genetik Operatörlü/ Operatörsüz analizleri yapılmıştır. Aynı sıkıştırma oranında Genetik Operatörlü KSA ‘nın en iyi sonuca sahip olduğu, bu algoritmayı sırasıyla PSA, KSA Çaprazlama, KSA Mutasyon, KKA Genetik Operatörlü, KKA Çaprazlama, KKA Mutasyon, KSA Genetik Operatörsüz, KKA takip ettiği görülmüştür.

16 kod kelimesine sahip bireylere uygulandığında,

Tablo 4.2. Tüm Algoritmaların 16 Kod Kelimeli Populasyona Uygulanması

Lena	KHerd	Kherd Mutasyon	Kherd Crossover	Kherd GA	PSO	ACO	ACO Mutasyon	ACO Crossover	ACO GA
Codeword Sayısı	256 vektör	256 vektör	256 vektör	256 vektör	256 vektör	256 vektör	256 vektör	256 vektör	256 vektör
Sıkıştırma Oranı	0,0352 bpp	0,0352 bpp	0,0352 bpp	0,0352 bpp	0,0352 bpp	0,0352 bpp	0,0352 bpp	0,0352 bpp	0,0352 bpp
MSE	261,2931	256,3999	247,3918	245,8341	251,2566	262,8041	262,5633	261,5796	261,4473
PSNR	23,9595 db	24,0416 db	24,1969 db	24,2244 db	24,1296 db	23,9344 db	23,9385 db	23,9548 db	23,9570 db



Şekil 4.6. 16 Kod Kelimeli Populasyona Tüm Algoritmaların Uygulanması

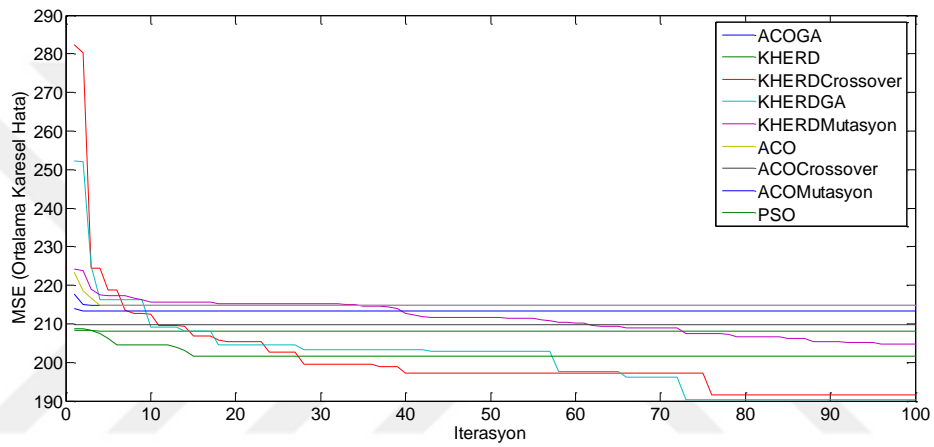
Sonucu

16 kod kelimesine sahip bireylere yapılan analizler sonucu aynı sıkıştırma oranında Genetik Operatörlü KSA'nın en iyi sonuca sahip olduğu görülmüştür ve bu çalışmayı KSA Çaprazlama, PSA, KSA Mutasyon, Genetik Operatörsüz KSA, Genetik Operatörlü KKA, KKA çaprazlama, KKA Mutasyon ve KKA takip etmiştir.

32 kod kelimesine sahip bireylere uygulandığında,

Tablo 4.3. Tüm Algoritmaların 32 Kod Kelimeli Populasyona Uygulanması

Lena	KHerd	Kherd Mutasyon	Kherd Crossover	Kherd GA	PSO	ACO	ACO Mutasyon	ACO Crossover	ACO GA
Codeword Sayısı	512 vektör	512 vektör	512 vektör	512 vektör	512 vektör	512 vektör	512 vektör	512 vektör	512 vektör
Sıkıştırma Oranı	0,0469 bbp	0,0469 bbp	0,0469 bbp	0,0469 bbp	0,0469 bbp	0,0469 bbp	0,0469 bbp	0,0469 bbp	0,0469 bbp
MSE	208,0744	204,763	191,5529	190,2629	201,5325	214,7737	212,3235	209,7271	214,8345
PSNR	24,9486 db	25,0183 db	25,3079 db	25,3373 db	25,0873 db	24,0811 db	24,8608 db	24,9143 db	24,8098 db



Şekil 4.7. 32 Kod Kelimeli Populasyona Tüm Algoritmaların Uygulanması

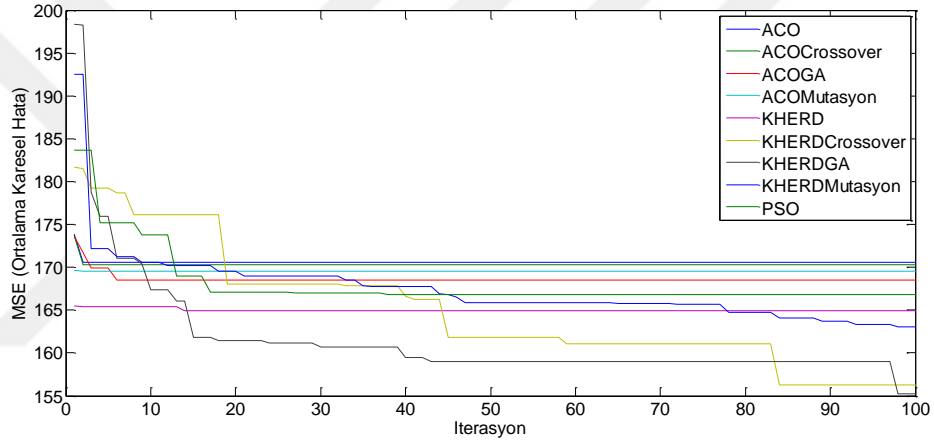
Sonucu

32 kod kelimesine sahip bireylere algoritmalar uygulandığında, aynı sıkıştırma oranında diğer analizlerde olduğu gibi Genetik Operatörlü KSA'nın en az hata oranına sahip olarak en iyi sonuca sahip olduğu ve bu algoritmayı sırasıyla KSA Çaprazlama, PSA, KSA Mutasyon, Genetik Operatörsüz KSA, KKA Çaprazlama, KKA Mutasyon, KKA ve Genetik Operatörlü KKA izlediği gözlemlenmiştir.

64 kod kelimesine sahipse,

Tablo 4.4. Tüm Algoritmaların 64 Kod Kelimeli Populasyona Uygulanması

Lena	KHerd	Kherd Mutasyon	Kherd Crossover	Kherd GA	PSO	ACO	ACO Mutasyon	ACO Crossover	ACO GA
Codeword Sayısı	1024 vektör	1024 vektör	1024 vektör	1024 vektör	1024 vektör	1024 vektör	1024 vektör	1024 Vektör	1024 vektör
Sıkıştırma Oranı	0,0625 bpp	0,0625 bpp	0,0625 bpp	0,0625 bpp	0,0625 bpp	0,0625 bpp	0,0625 bpp	0,0625 bpp	0,0625 bpp
MSE	164,9345	163,069	156,2451	155,2329	166,814	170,5776	169,5252	170,2811	168,5252
PSNR	25,9576 db	26,007 db	26,1927 db	26,221 db	25,9085 db	25,8116 db	25,8384 db	25,8191 db	25,8642 db



Şekil 4.8. 64 Kod Kelimeli Populasyona Tüm Algoritmaların Uygulanması

Sonucu

sonuçları elde edilir.

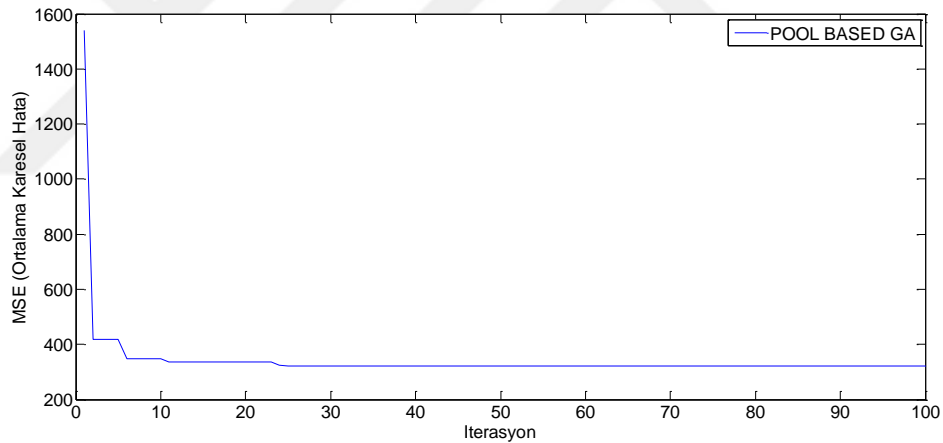
Son olarak 64 kod kelimeye sahip çalışmada diğer tüm analizlerde olduğu gibi Genetik Operatörlü KSA'nın en iyi sonuca sahip olduğu, diğer algoritmaların sırasıyla KSA Çaprazlama, KSA Mutasyon, Genetik Operatörsüz KSA, PSA, Genetik Operatörlü KKA, KKA Mutasyon, KKA Çaprazlama ve KKA takip ettiği görülmüştür.

Bu analizlerin dışında literatürde ilk defa Havuz Tabanlı Genetik Algoritması LBG tabanlı olarak 8, 16, 32, 64 kod kelimeye sahip 15 bireye 100 iterasyonda uygulanmıştır. Bu analiz sonucunda elde edilen grafik ve tablolar aşağıda gösterilmiştir.

8 kod kelimesine sahip bireylere uygulandığında değişim,

Tablo 4.5. Havuz Tabanlı Genetik Algoritmasının 8 Kod Kelimeli Populasyona Uygulanması

Havuz Tabanlı Genetik Algoritma	Codeword Sayısı	Sıkıştırma Oranı	MSE	PSNR
Lena 256x256 vektör	128 vektör	0,0254 bpp	320,5201	23,0722 db

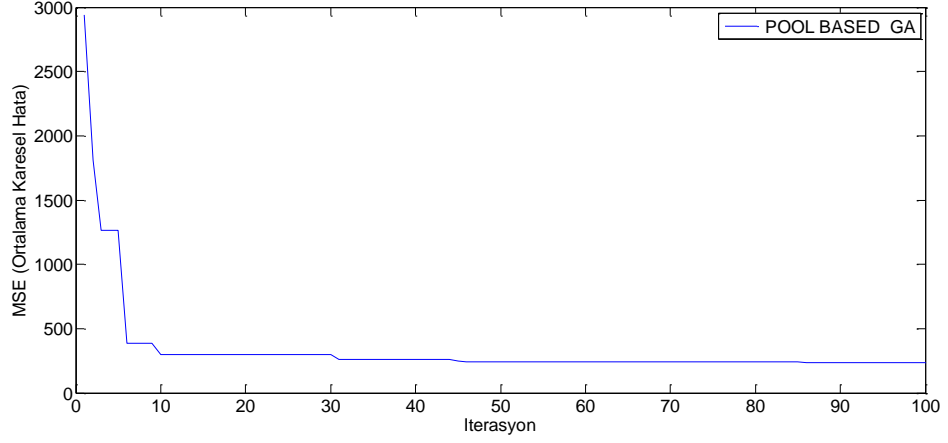


Şekil 4.9. 8 Kod Kelimeli Populasyona HTGA Uygulanması Sonucu

16 kod kelimesine sahip bireylere uygulandığında,

Tablo 4.6. Havuz Tabanlı Genetik Algoritmasının 16 Kod Kelimeli Populasyona Uygulanması

Havuz Tabanlı Genetik Algoritma	Codeword Sayısı	Sıkıştırma Oranı	MSE	PSNR
Lena 256x256 vektör	256 vektör	0,0352 bpp	237,7894	24,3689 db

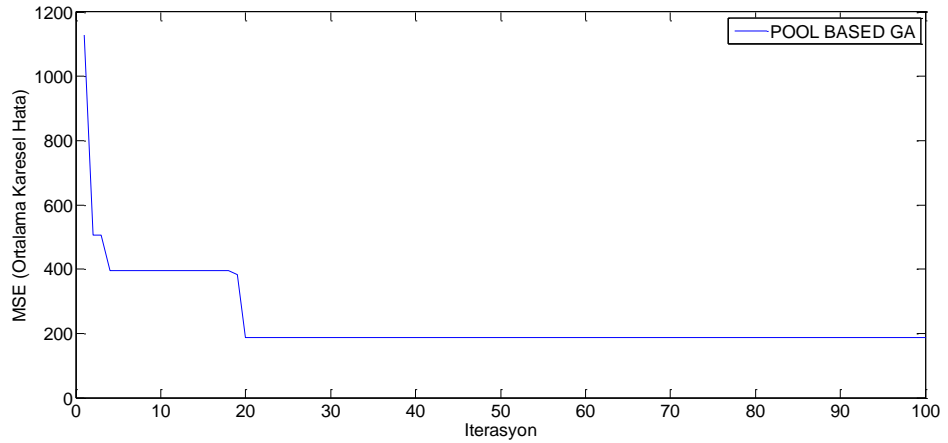


Şekil 4.10. 16 Kod Kelimeli Populasyona HTGA Uygulanması Sonucu

32 kod kelimesine sahip bireylere uygulandığında,

Tablo 4.7. Havuz Tabanlı Genetik Algoritmasının 32 Kod Kelimeli Populasyona Uygulanması

Havuz Tabanlı Genetik Algoritma	Codeword Sayısı	Sıkıştırma Oranı	MSE	PSNR
Lena 256x256 vektör	512 vektör	0,0468 bbp	185,9518	25,4368 db

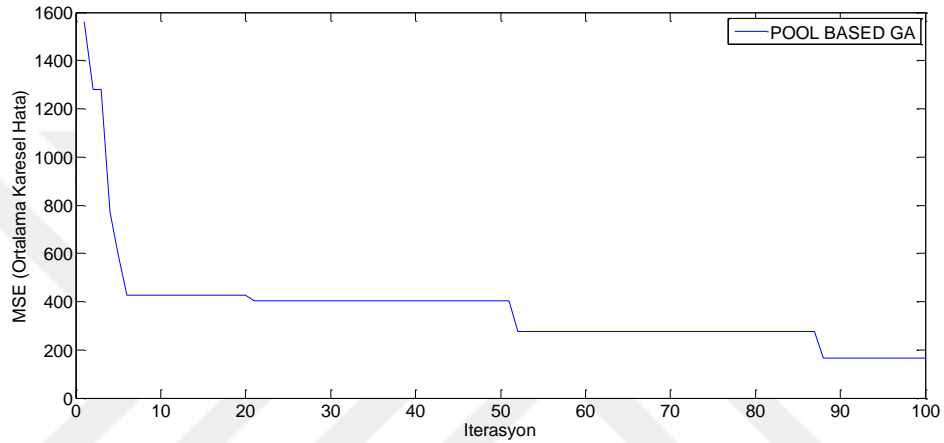


Şekil 4.11. 32 Kod Kelimeli Populasyona HTGA Uygulanması Sonucu

64 kod kelimesine sahip bireylere uygulandığında,

Tablo 4.8. Havuz Tabanlı Genetik Algoritmasının 64 Kod Kelimeli Populasyona Uygulanması

Havuz Tabanlı Genetik Algoritma	Codeword Sayısı	Sıkıştırma Oranı	MSE	PSNR
Lena 256x256 vektör	1024 vektör	0,0625 bpp	166,5081	25,9164 db



Şekil 4.12. 64 Kod Kelimeli Populasyona HTGA Uygulanması Sonucu

Havuz Tabanlı Genetik Algoritmanın yapılan tüm analizlerde hızlı bir şekilde global minimuma yaklaştığı gözlemlenmiştir.

Tüm analizler sonucunda tüm algoritmalar 8 kod kelime popülasyonuna uygulandığında Genetik Operatörlü KSA'nın, KSA Çaprazlama, KSA Mutasyon, KSA, PSA, Genetik Operatörlü KKA, KKA Çaprazlama, KKA Mutasyon, KKA 'dan sırasıyla 0,479 db, 0,639 db, 0,1357 db, 0,0385 db, 0,0731 db, 0,0734 db, 0,0738 db, 0,3996 db daha iyi ve daha az hata oranına sahip olduğu Tablo 4.1.'de gözlemlenmiştir.

16 kod kelimeli populusyona uygulandıđında ise Genetik Operatörlü KSA'nın, KSA Çaprazlama, KSA Mutasyon, KSA, PSA, Genetik Operatörlü KKA, KKA Çaprazlama, KKA Mutasyon, KKA 'dan sırasıyla 0,0275 db, 0,1828 db , 0,2649 db, 0,0948 db, 0,2674 db, 0,2696 db, 0,2859 db, 0,29 db daha iyi ve daha az hata oranına sahip olduđu Tablo 4.2.'de gözlemlenmiştir.

32 kod kelimeli populusyona uygulandıđında ise Genetik Operatörlü KSA'nın, KSA Çaprazlama, KSA Mutasyon, KSA, PSA, Genetik Operatörlü KKA, KKA Çaprazlama, KKA Mutasyon, KKA 'dan sırasıyla 0,0294 db, 0,319 db, 0,3887 db, 0,25 db, 0,5275 db, 0,423 db, 0,4765 db, 1,2562 db daha iyi ve daha az hata oranına sahip olduđu Tablo 4.3.'de gözlemlenmiştir.

64 kod kelimeli populusyona uygulandıđında ise Genetik Operatörlü KSA'nın, KSA Çaprazlama, KSA Mutasyon, KSA, PSA, Genetik Operatörlü KKA, KKA Çaprazlama, KKA Mutasyon, KKA 'dan sırasıyla 0,0283 db, 0,214 db, 0,2634 db, 0,3125 db, 0,3568 db, 0,4019 db, 0,3826 db, 0,4094 db daha iyi ve daha az hata oranına sahip olduđu Tablo 4.4.'de gözlemlenmiştir.

Bu algoritmaları 8 kod kelimeli populusyon ile Lena görüntüsüne uygulandıđında aşğıdaki görüntüler elde edilmiştir.



Şekil 4.13 Orijinal Resim Lena



Şekil 4.14 KSA Genetik Operatörsüz



Şekil 4.15 KSA Mutasyon Operatörüyle Operatörüyle



Şekil 4.16 KSA Çaprazlama Operatörüyle



Şekil 4.17 KSA Genetik Operatörlerle Boyut



Şekil 4.18 PSA ile Lena Resmi



Şekil 4.19 KKA Genetik Operatörsüz Operatörüyle



Şekil 4.20 KKA Mutasyon Operatörüyle



Şekil 4.21 KKA Çaprazlama Operatörüyle Operatörlerle Boyut



Şekil 4.22 KKA Genetik Operatörüyle



Şekil 4.23 HTGA ile Lena Resmi

Yapılan tüm analizler sonucu Genetik Operatörlü KSA'nın diğer tüm analizlerde daha iyi performansa sahip olduğu ve genetik operatörlerle KSA'nın çözüm uzayını gererek diğer algoritmalara göre daha kaliteli ve az hata oranında sıkıştırma yaptığı görülmüştür.

Bu tez çalışmasında görüntü sıkıştırma problemine uygulanan metasezgisel algoritmalar daha da genişletip ve geliştirilerek literatüre kazandırılabilir. Örneğin, çalışma esnasında test edilerek elde edilen seçili parametreleri optimize edecek yeni matematiksel modeller ortaya sürülerek bizim müdahalemize gerek kalmayabilir ya da çalışma adımlarını hızlandırmak için çeşitli fonksiyonlar algoritmaya eklenebilir.

Görüntü sıkıştırma problemi önemini gün geçtikçe önemini artırırken, daha yeni sıkıştırma teknikleri ortaya sürülebilir. Böylece hem zaman tasarrufu hem de kalite değerinin artışı gözlenebilir.

KAYNAKLAR

- [1] <http://www.yildiz.edu.tr/~bayram/sgi/saygi.htm>
- [2] Aksan, E., Doğan, S. Dalgacık Tabanlı Görüntü Sıkıştırma Tekniği. Havacılık ve Uzay Teknolojileri Dergisi. Temmuz 2004, 1(4), 47-53
- [3] Adıgüzel, V. Okatan, A. Atlı, Av. Vektör Kuantalama Yöntemi ve Farklı İmgelerde Sonuçları. *emo. org.tr*
- [4] Prakash, S.R. Shetty, V.S. Review On Optimization Techniques Used For Image Compression. International Journal of Research in Engineering and Technology. 2015 February, 4(2), 562-567
- [5] Huffman, D.A., A Method for the Construction of Minimum-Redundancy Codes. Proceedings of the I.R.E . 1952 September, 40(9), 1098-1101
- [6] https://tr.wikipedia.org/wiki/Huffman_kodu
- [7] Donald E. Knuth, Dynamic Huffman Coding. Journal of Algorithm. 1985, 6(2), 163-180
- [8] Mesut, Altan. Veri Sıkıştırmada Yeni Yöntemler. Trakya Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Anabilim Dalı, Edirne, 2006, 151. (Doktora Tezi).
- [9] Hardin, C. D., Zabele, S., Arithmetic Coding for Lossless and Loss-Inducing Image Compression. Sixth Multidimensional Signal Processing Workshop. 1989 September Session FA2, 214
- [10] Rodney, J. Smith. Streaming Using Compression System Dynamic Connection Groups .US Patent No 5,748,955, priority date 20 December 1993
- [11] Witten, H. I., Moffat, A., Bell, T.C. Managing Gigabytes. Ed: Jennifer Mann, Morgan Kaufmann, San Francisco, U.S.A, 1999, 550.
- [12] Ziv. J., Lempel, A., A Universal Algorithm for Sequential Data Compression. IEEE Transactions on Information Theory. 1977 May, 23(3), 337-343
- [13] Ziv. J., Lempel, A., Compression of Individual Sequences via Variable-Rate Coding. IEEE Transactions on Information Theory. 1978 September, 24(5), 530-536
- [14] Welch, T. A., A Technique for High-Performance Data Compression. Computer. 1984, 17(6), 8-19
- [15] Capon, J., A Probabilistic Model for Run-Length Coding of Pictures. IRE Transactions on Information Theory. 1959, 5(4), 157-163
- [16] https://en.wikipedia.org/wiki/Pulse-code_modulation

- [17] Sears, R. W., Electron Beam Deflection Tube for Pulse Code Modulation. The Bell System Technical Journal. 1948, 27(1), 44-57
- [18] Oliver, B. M., Pierce, J. R., Shannon, C. E., The Philosophy of PCM. Proceedings of the IRE. 1948, 36(11), 1324-1331
- [19] Cutler, C.C. Differential Quantization of Communication Signals US Patent No.US 2605361 A. New York, US Patent and Trademark Office, 1952.
- [20] <https://omertaban.com/2014/06/28/dct-ile-goruntu-kodlama-ve-sikistirma/>
- [21] <https://sellinalp.wordpress.com/2013/11/17/kayipli-hareketli-goruntu-sikistirma-yontemleri-h261-h263/>
- [22] http://www.chip.com.tr/makale/nedir-bu-jpeg-ayrik-kosinus-donusumu_11812_7.htm
- [23] <https://serkank.wordpress.com/2007/04/01/ayrik-kosinus-donusumu/>
- [24] Ahmed, N., Natarajan, T., Rao, K. R., Discrete Cosine Transform. IEEE Transactions on Computers. 1974, C-23(1), 90-93
- [25] Narasimha, M., Peterson, A., On the Computation of the Discrete Cosine Transform. IEEE Transactions on Communications. 1978, 26 (6), 934-936
- [26] Arai, Y., Agui, T., Nakajima, M., A Fast DCT-SQ Scheme for Images. IEICE Transactions. 1988, E71(11), 1095-1097
- [27] Feig, E., Winograd, S., Fast Algorithms for the Discrete Cosine Transform. IEEE Transactions on Signal Processing . 1992, 40(1), 2174-2193
- [28] https://en.wikipedia.org/wiki/Wavelet_transform
- [29] <https://jpeg.org/jpeg2000/index.html>
- [30] Ramakrishnan, A. G., Saha, S., ECG Coding by Wavelet-Based Linear Prediction. IEEE Transactions on Biomedical Engineering. 1977, 44 (12), 1253-1261
- [31] Lee, J. S., Sun, Y. N., Chen, C. H., Multi-Scale Corner Detection by Using Wavelet Transform. IEEE Transactions on Image Processing. 1995, 4(1), 100-104
- [32] https://en.wikipedia.org/wiki/Sub-band_coding
- [33] Crochiere, R.E., Sub-Band Coding. The Bell System Technical Journal. 1981, 60(7), 1633-1653
- [34] Woods, J., O'neil, S., Sub-Band Coding of Images. Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '86. 1986 April, Tokyo, Japan, 1005-1008

- [35] Saito, T., Higuchi, H., Komatsu, T., Sub-Band Coding with Improved Permutation Codes for High Fidelity Image Compression. Acoustics, Speech, and Signal Processing, 1991.ICASSP-91., 1991 International Conference on. 1991 April, Toronto, Canada, 2641-2644
- [36] Barnsley, M. F., Sloan, A. D., Methods and Apparatus for Image Compression by Iterated Function System US Patent No.US 4941193 A. New York, US Patent and Trademark Office, 1987
- [37] Jacquin, A.E., Image Coding Based on a Fractal Theory of Iterated Contractive Image Transformations. IEEE Transactions on Image Processing. 1992, 1(1), 18-30
- [38] Gray, R. M., Vector Quantization. IEEE ASSP Magazine. 1984, 1(2), 4-29
- [39] Goldberg, M., Boucher, P., Shlien, S., Image Compression Using Adaptive Vector Quantization. IEEE Transactions on Communications. 1986, 34(2), 180-187
- [40] Gersho, A., Ramamurthi, M., Image Coding Using Vector Quantization. Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '82, 1982 May, Paris, France, 428-431
- [41] Nasrabadi, N. M., King, R. A., Image Coding Using Vector Quantization: A Review. IEEE Transactions on Communications. 1988, 36(8), 957-971
- [42] Makhoul, J., Roucos, S., Gish, H., Vector Quantization in Speech Coding. Proceedings of the IEEE. 1985, 73(11), 1551-1588
- [43] <http://www.data-compression.com/vq.shtml#lbg>
- [44] Linde, Y., Buzo, A., Gray, R., An Algorithm for Vector Quantizer Design. IEEE Transactions on Communications. 1980, 28(1), 84-95
- [45] Lin, Y. C., Tai, S. C., A Fast Linde- Buzo-Gray Algorithm in Image Vector Quantization. IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing. 1998, 45(3), 432-435
- [46] Chang, R. F., Chen, W.T., Wang, J.S., A Fast Finite-State Algorithm for Vector Quantizer Design. IEEE Transactions on Signal Processing. 1992, 40(1), 221-225
- [47] Bardekar, A. A., Tijare, P. A., A Review on LBG Algorithm for Image Compression. International Journal of Computer Science and Information Technologies. 2011, 2(6), 2584-2589
- [48] Aydın, S., Kılıç, İ., Temeltaş, H. , Linde Buzo Gray Sınıflandırılmalı Yapay Sinir Ağları ile Mobil Robotlarda Yörünge Hesabı. KSÜ Fen ve Mühendislik Dergisi. 2007, 10(2), 51-57
- [49] Özkan, K., Seke, E., Görüntü İstatistiklerinden Faydalanarak LBG Algoritmasının Güncellenmesi. Eskişehir Osmangazi Üniversitesi Müh.Mim.Fak.Dergisi. 2008, 21(1), 141-152

- [50] <http://portal.netcad.com.tr/display/HELP/Voronoi>
- [51] Çelebi, M., Buldu, A., Ses Komut Tanıma ile Gezgin Araç Kontrolü. 2th International Symposium on Innovative Technologies in Engineering and Science. 2014 June, Karabuk, Turkey, 395-404
- [52] Rabiner, L., Juang, B. H., Fundamentals of Speech Recognition., AT&T/Prentice-Hall International, New Jersey, USA, 1993, 496.
- [53] http://web.firat.edu.tr/iaydin/bmu579/bmu579_Bolum_1.pdf
- [54] <https://metasezgisel.files.wordpress.com/>
- [55] Singh, G. P, Singh, A. Comparative Study of Krill Herd, Firefly and Cuckoo Search Algorithms for Unimodal and Multimodal Optimization. I. J. Intelligent Systems and Applications. 2014, 03, 35-49
- [56] Gandomi, A.H., Alavi A.H., Krill herd: A New Bio-inspired Optimization Algorithm, Communications in Nonlinear Science and Numerical Simulation. 2012, 17(12), 4831-4845
- [57] Hardy, A. C., Gunther, E.R., The Plankton of the South Georgia Whaling Grounds and Adjacent Waters., Cambridge Eng. University Press, London, England, 1998, 456.
- [58] Watkins, J. L., Variations In the Size of Antarctic Krill, Euphausia Superba Dana in Small Swarms. Marine Ecology-Progress Series. 1986, 31, 67-73
- [59] Marr, J.W.S. The Natural History and Geography of The Antarctic Krill (Euphausia Superba Dana). Discovery Reports. 1962, 32, 33-464
- [60] Nicol S., Living krill, Zooplankton and Experimental Investigations. Proceedings of the International Workshop on Understanding Living Krill for Improved Management and Stock Assessment Marine and Freshwater Behaviour and Physiology. 2003, 191-205
- [61] Harman, F., Kılıç, İ., Karides Sürüsü Algoritmasının Görüntü Sıkıştırma Kullanılması, EEB2016, 11-13 Mayıs 2016, Tokat, 77-81
- [62] <http://healthhamster.com/antarctic-krill-oil/>
- [63] Singh, V., Sood, M., Krill Herd Clustering Algorithm using DBSCAN Technique. International Journal of Computer Science & Engineering Technology. 2013 March, 4(3), 197-201
- [64] Wang, G. G., Guo, L., Gandomi. A. H, Alavi, A. H, Duan, H. , Simulated Annealing-Based Krill Herd Algorithm for Global Optimization. Abstract and Applied Analysis. 2013, 2013, 11 pages.

- [65] Gai Wang, G. G., Guo, L., Gandomi. A. H, Hao, G. S., Wang, H., Chaotic Krill Herd Algorithm. *Information Sciences*. 2014, 274, 17-34
- [66] Wang, G. G., Guo, L., Gandomi. A. H, Cao, L., Alavi, A. H., Duan, H., Li, J., Lévy-Flight Krill Herd Algorithm. *Mathematical Problems in Engineering*. 2013, 2013, 14 pages
- [67] Guo, G., Wang, G. G., Gandomi, A. H., Alavi, A. H., Duan, H., A New Improved Krill Herd Algorithm for Global Numerical Optimization. *Neurocomputing*. 2014, 138, 392-402
- [68] Wang, G. G., Gandomi, A. H., Alavi, A. H., Stud Krill Herd Algorithm. *Neurocomputing*. 2014, 128, 363-370
- [69] Li, J., Tang, Y., Hua, C., Guan, X., An Improved Krill Herd Algorithm: Krill Herd with Linear Decreasing Step. *Applied Mathematics and Computation*, 2014, 234, 356-367
- [70] Gölcük, İ., Baykasoğlu, A., Madenoğlu, S., Kril Sürüsü Algoritması İle Atölye Çizelgeleme DEU Mühendislik Fakültesi Mühendislik Bilimleri Dergisi. 2014 Eylül, 16(48), 61-75
- [71] Kennedy, J., Eberhart, R., Particle Swarm Optimization. *Neural Network*. 1995, 4, 1942-1948
- [72] Akyol, S., Alataş, B., Güncel Sürü Zekâsı Optimizasyon Algoritmaları. *Nevşehir Üniversitesi Fen Bilimleri Enstitü Dergisi*. 2012, 1, 36-50
- [73] Kennedy, J., The Particle Swarm: Social Adaptation of Knowledge. *Proceedings of IEEE International Conference on Evolutionary Computation*. 1997, Indianapolis, USA, 303-308
- [74] Kennedy, J., Eberhart, R. C., Shi, Y., *Swarm Intelligence*, Morgan Kauffman, San Francisco, USA, 2001, 512
- [75] Poli, R., An Analysis of Publications on Particle Swarm Optimisation Applications. *University of Essex, Technical Reports*. 2007, 57 pages.
- [76] Poli, R., Analysis of the Publications on the Applications of Particle Swarm Optimisation. *Journal of Artificial Evolution and Applications*. 2008, 2008, 10 pages
- [77] Bonyadi, M. R., Michalewicz, Z., Particle Swarm Optimization for Single Objective Continuous Space Problems: A Review. *MIT Press Journals*. 2016, DOI: 10.1162/ EVCO_r_00180
- [78] Uma, K., Palanisamy, P. G., Poornachandran, P. G., Comparison of Image Compression Using GA, ACO and PSO Techniques. *Recent Trends in Information Technology (ICRTIT)*, 2011 International Conference on. , 3-5 June 2011, Chennai, Tamil Nadu, 815-820

- [79]Colorni, A., Dorigo, M., Maniezzo, V. Distributed Optimization by Ant Colonies, Actes de la Première Conference Européenne Sur la vie Artificielle, Paris, France, Elsevier Publishing, 1991, Paris, France, 134-142
- [80] Dorigo, M., Optimization, Learning and Natural Algorithms. Politecnico di Milano, Dipartimento di Elettronica, Milan, 1992, 140 (Doktora Tezi).
- [81] Dorigo, M., Maniezzo, V., Colorni, A., Ant System: Optimization by a Colony of Cooperating Agents. IEEE Transactions on Systems, Man, and Cybernetics – Part B. 1996, 26, 29–41.
- [82] Bonabeau, B., Dorigo, M., Theraulaz, G., Swarm Intelligence: From Natural to Artificial Systems. Oxford University, New York, USA, 1999, 307.
- [83] Zhihong, X., Bo, S., Yanyan, G., Using Simulated Annealing and Ant Colony Hybrid Algorithm to Solve Traveling Salesman Problem, Intelligent Networks and Intelligent Systems, 2009. ICINIS '09. Second International Conference, 1-3 November, 2009, Tianjin, China, 507-510
- [84] Zhang- Liang, W., Yue- Guang, L., An Ant Colony Algorithm with Tabu Search and its Application. Intelligent Computation Technology and Automation (ICICTA), 2011 International Conference (Vol. 2), 28-29 March, 2011, Shenzhen, Guangdong , 412-416.
- [85] Mouhob, M., Wang, Z., Ant Colony with Stochastic Local Search for the Quadratic Assignment Problem, 18th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'06), November, 2006, Arlington, VA, 127-131.
- [86] Jaferzadeh, K., Shakeri, M., Gholami, S., Ant Colony Algorithm for Fractal Image Compression Based on Edge Property, Soft Computing, Computing with Words and Perceptions in System Analysis, Decision and Control, 2009(ICSCCW 2009 Fifth International Conference), 2-4 September, 2009, Famagusta, Cyprus, 1-4
- [87] Keskindürk, T., Söyler, H., Global Karınca Kolonisi Optimizasyonu. Gazi Üniv. Müh. Mim. Fak. Der. 2006, 21(4), 689-698.
- [88] Fraser, A., Simulation of Genetic Systems by Automatic Digital Computers. I. Introduction. Csiro Publishing Australian Journal of Biological Sciences. 1969, 10, 484-491
- [89] Fraser, A., Burnell, D., Computer Models in Genetics, New York: McGraw-Hill, New York, USA, 1970, 206 pages
- [90] Crosby, J. L., Computer Simulation in Genetics, London: John Wiley&Sons, Durham University, UK , 1973, 492 pages
- [91] Fogel, D. B., Evolutionary Computation: The Fossil Record, Wiley- IEEE Press, New York, USA, 1998, 656 pages

[92] Yimin, C., Yixiao, W., Qibin, S., Longxiang, S., Digital Image Compression Using a Genetic Algorithm. Real Time Imaging. December 1999, 5(6), 379-383



ÖZGEÇMİŞ

Adı Soyadı : Fatma HARMAN

Doğum Yeri ve Yılı : Alaşehir, 1989

Medeni Hali : Bekar

Yabancı Dili : İngilizce

E-posta : fatma.harman@cbu.edu.tr

Eğitim Durumu

Lise : Alaşehir Yabancı Dil Ağırlıklı Lisesi, 2007

Lisans Bölümü, 2011 :Balıkesir Üniversitesi, Elektrik Elektronik Mühendisliği

Mesleki Deneyim

Celal Bayar Üniversitesi Mühendislik Fakültesi, Araştırma Görevlisi 2014-(halen)

Yayımları

1. Harman, F., Kılıç, İ. Karides Sürüsü Algoritmasının Görüntü Sıkıştırma Kullanılması, EEB2016, 11-13 Mayıs 2016, Tokat (Bildiri Kitabı, 77-81 s.)