

## LEVY UÇUŞLU MEYVE SİNEĞİ ALGORİTMASI İLE GÖRÜNTÜ SIKIŞTIRMA

İlker KILIÇ\*

Pamukkale Üniversitesi, Mühendislik Fakültesi, Elektrik-Elektronik Mühendisliği Bölümü, Denizli, 20160, Türkiye  
Geliş Tarihi/Received Date: 07.08.2021 Kabul Tarihi/Accepted Date: 09.01.2022 DOI: 10.54365/adyumbd.980173

### ÖZET

Sayısal görüntülerin sıkıştırılıp arşivlenmesi günümüz teknolojisinde çok önemli bir ihtiyaç haline gelmiştir. Son yıllarda doğadan esinlenerek geliştirilen PSO(Parçacık sürü optimizasyonu), MSO(Meyve sineği optimizasyonu), ABO(Ateşböceği optimizasyonu), GA(Genetik Algoritma) gibi sezgisel metodlar da vektör tabanlı görüntü sıkıştırma için kullanılmaya başlamıştır. Bu çalışmada MSO, meyve sineklerinin sorunsuz bir şekilde global optimum noktaya ulaşabilmesi için Levy Uçuşu tekniği ile birleştirilmiştir. MSO algoritmasının en büyük sorunlarından biri de lokal minimum noktaya takılıp global minimuma ulaşamamasıdır. Çoğu zaman küçük nadiren de büyük yarıçap değeri veren Levy Fonksiyonu yardımı ile meyve sineği lokal minimum noktaya hiç takılmayıp global minimum noktayı garantilemektedir. Bu yeni geliştirilen LMSO(Levy uçuşlu meyve sineği optimizasyonu) tekniği standart görüntüler üzerinde test edilmiş ve aynı sıkıştırma oranlarında MSE, PSNR ölçütleri kullanıldığında diğer sezgisel algoritmalarından üstün olduğu gösterilmiştir.

**Anahtar Kelimeler:** Levy Uçuşu, Görüntü Sıkıştırma, Meyve Sineği Optimizasyonu, Vektör Nicemleme

## IMAGE COMPRESSION BY LEVY FLIGHT FRUIT FLY ALGORITHM

### ABSTRACT

It has become an important requirement to compress and archive digital images in nowadays technology. The bioinspired algorithms such as Particle Swarm Optimization (PSO), Fruit Fly Optimization (FFO), Firefly Optimization(FO), Bat Optimization(BO) algorithms have started to be used for vector-based image compression. In this paper, the Fruitfly algorithm is combined with the Lévy flight distribution in order to reach the global optimum point without encountering any problem. One of the FFO problems is to be captured the fruit flies by a local minimum and cannot reach the global optimum point. With the help of Levy function, which results in small values rarely and big values generally, the fruit flies guarantee the global minimum point without getting captured by any local minimum. This new proposed Levy Flight Fruit Fly Optimization (LFFO) technique is applied on the standard images and it is seen that at the same and compression rates, the proposed new LFFO technique is better than the other metaheuristic optimization methods

**Keywords:** Levy Flight, Image Compression, Fruit Fly Optimization, Vector Quantization

### 1. Giriş

Vektör Nicemleme(VN), blok tabanlı kayıplı başarılı bir görüntü sıkıştırma metodu olup vektör listesi üretim algoritması (LBG) ile birleştirilmiştir [1]. LBG, bölgesel minimum hata bulma prensibine göre resmi temsil eden en iyi vektör listesini oluşturur[2]. Bu metod Lin tarafından daha gürbüz ve hızlı

\* e-posta: [ilkerk@pau.edu.tr](mailto:ilkerk@pau.edu.tr) ORCID ID: <https://orcid.org/0000-0003-3978-4829> (Sorumlu Yazar),

hale getirilmiştir[3]. Toplam hatayı artıran kötü vektörlerin tespit edilip iyileştirilmesi [4] ve hesaplama süresinin azaltılması [5] ile LBG tekniği son şekline ulaştırılmıştır [6]. Bulanık K-Ortalamlar (BKO) [7-9] ve K-ortalamlar (KO) [10] metodları da görüntü sıkıştırma için kullanılan resmi temsil edebilen vektör listesi üretebilen tekniklerdendir.

Literatürde resmi temsil eden en iyi vektör listesi oluşturmak ve sıkıştırma yapmak için ilk olarak Genetik Algoritma (GA) [11-14] ve Genetik Benzetimli Tavlama[15] metodları kullanılmıştır. Bu makalelerde rasgele üretilip çaprazlama ve mutasyon işlemleri ile oluşturulan vektör listelerinden rastgele üretilip çaprazlama ve mutasyon işlemleri ile oluşturulan vektör listelerinden en iyisi saklanarak global sonuca ulaşılmıştır. Diğer bir çalışma da Parçacık Sürü Optimizasyonu (PSO) ile ve Bulanık Mantığı birleştiren (bir) algoritma olup vektör listesindeki resim bloklarının orijinal resimdeki bloklara benzerliği bulanık karar verme mekanizması ile artırılmıştır[16]. Orijinal resimde vektör listesini oluşturmak için kullanılan bir diğer metod ise Bal Arısı Çiftleşmesi (BAÇ) metodudur [17]. Buradaki hedef, oluşturulan vektör listesinin güvenilirliğini ve kalitesini artırmaktır. Bir diğer metod ise Ateşböceği optimizasyonu(ABO) metodu olup yüksek sıkıştırılarda vektör listesinin hata fonksiyonu minimize edilmiştir[18-19]. Sayısal görüntü için oluşturulan vektör listesi Karınca Kolonisi Optimizasyonu (KKO) metodu ile diğer algoritmalara göre çok daha kısa sürede elde edilmiştir [20]. Parametre sayısı ve uygulama kolaylığı nedeniyle MSO metodu da analitik çözümü olmayan mühendislik problemlere başarılı yaklaşık çözümler üretebilmiştir [21-32], [37-40]. En son çıkan tekniklerden biri olan Yarasa Optimizasyonu (YO) birçok tekniğin birleştirilmesi ile literatürde yerini almıştır [33]. Başka bir Algoritma olan Guguk kuşu algoritması VN tekniğine başarılı bir şekilde entegre edilmiştir [36].

## 2. Vektör Listesi Üretimi

Bu bölümde önerilen yeni sezgisel algoritmanın alt parçalarından olan LBG algoritması ve Levy uçuşu metotları anlatılacaktır.

### 2.1 Vektör Nicemleme (VN) ve LBG algoritması

VN vektör tabanlı kayıplı bir görüntü sıkıştırma tekniğidir. LBG algoritması VN tekniğinin en önemli parçasıdır. Orijinal görüntü  $N \times N$  pixel  $Y = \{x_{ij}\}$  fonksiyonu ve  $m \times m$  pixel boyutunda  $N_b = \left(\frac{N}{m} \times \frac{N}{m}\right)$  alt bloklar ile temsil edilsin  $X = \{x_i, i = 1, 2, \dots, N_b\}$ . L blok büyüklüğü olan  $m \times m$  değerini temsil etsin.  $\mathcal{R}^L$ , L boyutlu öklit uzayı olmak üzere resmi oluşturan alt bloklar  $x_i \in \mathcal{R}^L$  şeklinde tanımlanabilir.  $N_c$ , üretilen vektör listesi içerisindeki vektörleri temsil etsin;  $C = \{C_1, C_2, \dots, C_{N_c}\}$ ,  $C_j \in \mathcal{R}^L$ ,  $j=1, 2, \dots, N_c$ . Orijinal görüntü vektörleri satır vektörleri olarak  $x_i = (x_{i1}, x_{i2}, x_{i3}, \dots, x_{iL})$  şeklinde tanımlanıp vektör listesi içerisindeki i. vektör  $c_i = (c_{i1}, c_{i2}, c_{i3}, \dots, c_{iL})$  olarak ifade edilsin.

VN, resim içerisindeki bir  $m \times m$  alt bloğu vektör listesindeki en uygun vektöre eşleştirme yapma prensibine dayanır. Böylece resim içindeki orijinal bloğun kendisini göndermek yerine LBG ile oluşturulan vektör listesindeki en az hata veren blok numarası gönderilir. Böylece kayıplı bir sıkıştırma yapılmış olur. Vektör listesini oluşturan C kümesi aşağıdaki Ortalama Karesel Hata (OKH) ifadesi (1) minimize edilerek bulunur.

$$OKH(C) = \frac{1}{N_b} \sum_{j=1}^{N_c} \sum_{i=1}^{N_b} \mu_{ij} \|x_i - c_j\|^2 \quad (1)$$

$$\sum_{j=1}^{N_c} \mu_{ij} = 1, \quad i \in \{1, 2, \dots, N_b\} \quad (2)$$

$$\mu_{ij} = \begin{cases} 1 & \text{eğer } x_i, j. \text{ küme içerisinde ise} \\ 0 & \text{diğer durumlarda} \end{cases} \quad (3)$$

$\|x_i - c_j\|$  ifadesi vektör listesindeki  $c_j$  vektörü ile  $x_i$  görüntü bloğu arasındaki öklit uzaklığını ifade etmektedir. Local optimum vektör listesi oluşturmak için aşağıdaki iki kural uygulanır.

a) Vektör kümesi  $R_j, j=1,2,\dots,N_c$  aşağıdaki koşulu sağlamalıdır

$$R_j \supset \{x \in X: d(x, c_j) < d(x, c_k), \forall k \neq j\} \quad (4)$$

b)  $R_j$  vektör kümesinin  $c_j$  merkezi aşağıdaki formül ile hesaplanır

$$c_j = \frac{1}{N_j} \sum_{i=1}^{N_j} x_i, x_i \in R_j \quad (5)$$

Burada  $N_j, R_j$  vektör kümesine ait toplam vektör sayısını ifade etmektedir. Görüntüye ait vektörleri  $x_i, i = 1, 2, \dots, N_b$ , öklit mesafesini  $d$  ve başlangıç anlık vektör listesi  $c_j(0), j = 1, 2, \dots, N_c$  olarak kabul edilip aşağıdaki ardaşık işlemler uygulandığında lokal optimum vektör listesi oluşturulmuş olur:

a) Orijinal görüntü blokları üzerinde öklit uzaklığı kullanılarak istenilen sayıda Voronoi bölgesine ayrılır. Kümeleme sonucu,  $N_b \times N_c$  boyutuna sahip  $U$  gösterge matrisinde saklanır.

$$\mu_{ij} = \begin{cases} 1, & \text{eğer } d(x_i, c_j(k)) = \min d(x_i, c_j(k)) \\ 0, & \text{diğer durumda} \end{cases} \quad (6)$$

b) Herbir Voronoi küme merkezi belirlenir. Eski merkezler aşağıdaki formül kullanılarak yenileri ile değiştirilir.

$$c_j(k+1) = \frac{\sum_{i=1}^{N_b} \mu_{ij} x_i}{\sum_{i=1}^{N_b} \mu_{ij}}, \quad j = 1, 2, \dots, N_c \quad (7)$$

c) Voronoi küme merkezleri  $c_j$  değişmeyene kadar a ve b adımları tekrarlanır.

## 2.2 Levy Dağılım Fonksiyonu

Levy dağılım fonksiyonu, standart Gauss dağılımında standart sapmayı  $x$  in bir fonksiyonu şeklinde yazarak elde edilir.[34-35]. Gauss dağılım fonksiyonu (8) eşitliğinde verilmiştir.

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (8)$$

Bu fonksiyonda  $\mu=0$  ve standard sapmayı  $\sigma, x$  değişkeninin bir fonksiyonu yapalım.

$$\sigma = \sqrt{\frac{x^3}{c}} \quad (9)$$

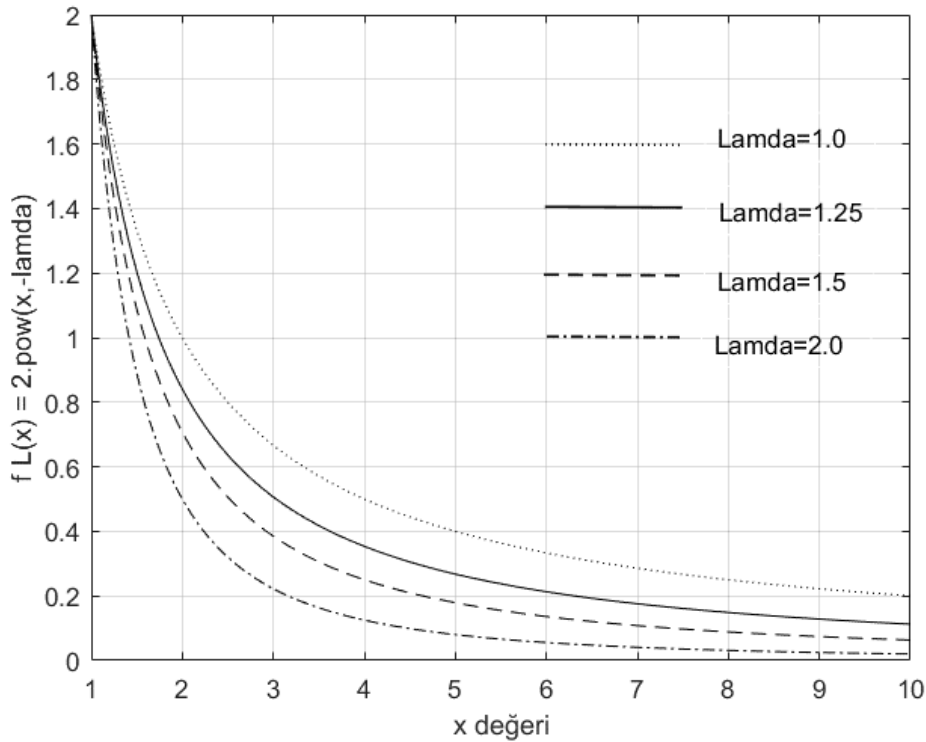
Gauss dağılım fonksiyonunda ortalama değeri  $\mu=0$  alıp standart sapmayı (9) nolu eşitlik ile  $x$ 'in bir fonksiyonuna dönüştürürsek,  $f(x)$  fonksiyonu (10) nolu eşitlik ile Levy dağılım fonksiyonuna dönüştürülmüş olur.

$$f_L(x) = \sqrt{\frac{c}{2\pi}} \left( \frac{e^{-\frac{c}{2x}}}{x^{\frac{3}{2}}} \right) \quad (10)$$

Levy fonksiyonunda  $x$  değerini artırdıkça eksponensiyel kısım 1 değerine yaklaşacak ve Levy fonksiyonu yaklaşık olarak (11) eşitliğindeki ifadeye dönüşecektir.

$$f_L(x) \cong k \cdot x^{-\lambda} \quad (11)$$

Bu  $f_L(x)$ , yaklaşık Levy dağılımı Şekil 1. de görüldüğü gibi  $x$ 'in küçük değerlerinde büyük,  $x$ 'in büyük değerlerinde ise sifıra yakın bir değer üretecektir. Lamda( $\lambda$ ) nın  $\lambda_1= 1.0$ ,  $\lambda_2= 1.25$ ,  $\lambda_3= 1.5$ ,  $\lambda_4= 2.0$  değerleri için Levy fonksiyonu Şekil 1. de verilmiştir.



Şekil 1. Değişik lamda( $\lambda$ ) değerleri için Levy dağılım fonksiyonu  $f_L(x)$

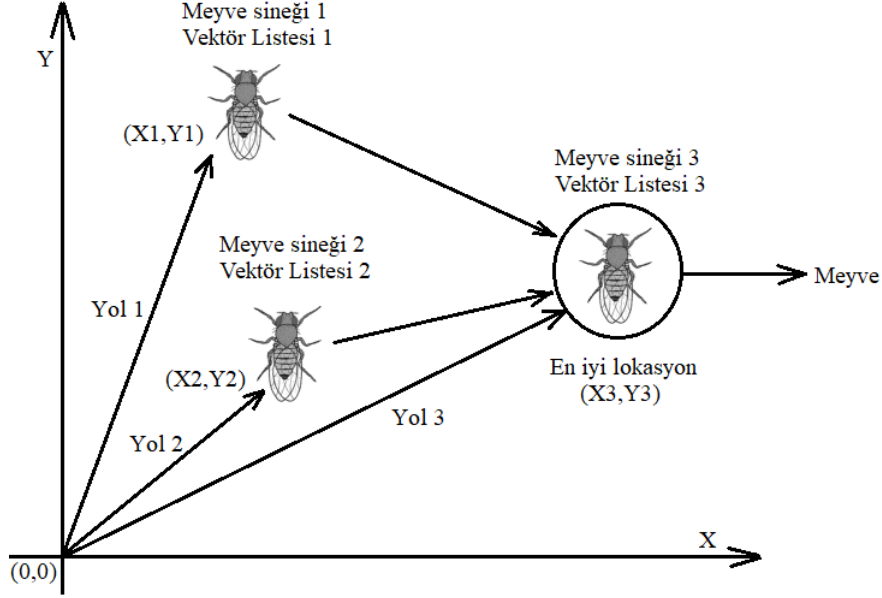
### 3. Levy Uçuşlu Meyve Sineği Algoritması

#### 3.1 Klasik Meyve Sineği Optimizasyon Algoritması

MSO algoritması doğada Drosophila ismi ile bilinen meyve sineğinin nektar bulabilmek için izlediği stratejiyi taklit eder [21]. Bir meyve sineği nektara ulaşmak için iki yol izler. Birincisi koku takibi yaparak yön tayin eder. İkinci olarak keskin görüşünü kullanarak daha iyi bir yiyecek kaynağı için diğer sinekleri izler. Meyve sineğinin beslenmesini üç adımda özetleyebiliriz;

- i) Başlangıç sinek popülasyonunu üretme
- ii) Koku duyu organı ile araştırma fazı
- iii) Görüş duyu organını kullanarak araştırma fazı.

Şekil 2 de sineklerin koku sensörü ile araştırması ve ardından görme duyusu ile en iyi meyveyi bulan sineğin lokasyonunda toplanmaları temsil edilmiştir.



Şekil 2. Meyve sineği beslenmesi

MSO tekniğinin adımları aşağıda sıralanmıştır.

**Adım 1.** Maksimum iterasyon sayısı, meyve sineği sayısı, araştırma yarıçapı belirlenir.

**Adım 2.** Araştırma bölgesindeki meyve sineklerinin lokasyonları rasgele olarak belirlenir.

**Adım 3.** Koku yoğunluğu araştırma fazı; önceden belirlenen araştırma bölgesinde meyve sinekleri rasgele hareket ettirilip meyve nektarı araştırması yapılır.

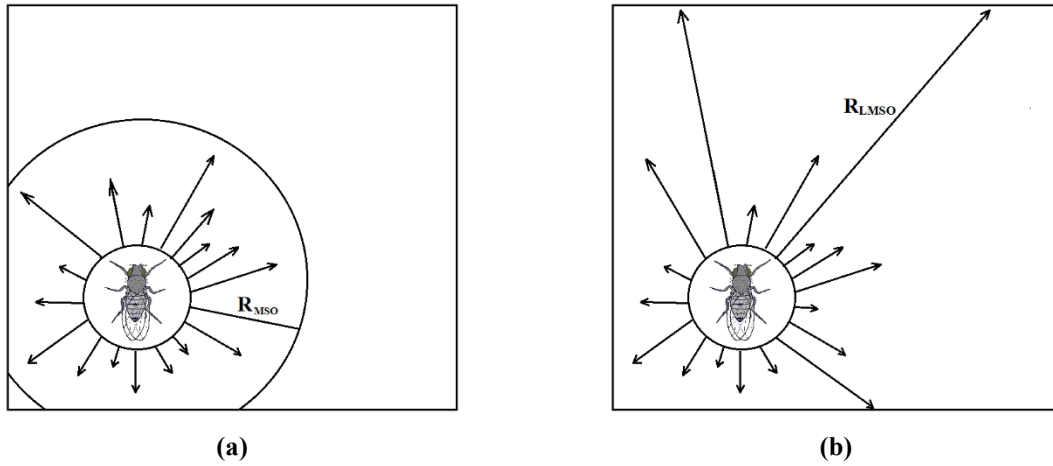
**Adım 4.** Meyve sineklerinin ulaştıkları nektar ile doğru orantılı olarak herbir sineğe ait koku konsantrasyonu belirlenir

**Adım 5.** Görüş araştırma fazı; En yüksek koku yoğunluğuna sahip meyve sineği tespit edilir. Bölge içerisindeki tüm sinekler, en yüksek koku yoğunluğuna sahip sineğin konumuna götürülür. Bu konum bir sonraki iterasyonda meyve sineklerinin ortak başlangıç noktası yapılır.

**Adım 6.** Maksimum iterasyon sayısına ulaşıldığında algoritma durdurulur, aksi durumda adım 3, 4, 5 ve 6 ardaşık olarak tekrar edilir.

### 3.2 Levy uçuşlu Meyve Sineği Optimizasyon Algoritması

Klasik MSO algoritmasında meyve sinekleri sabit bir yarıçap içerisinde rasgele hareket ederek ilerlerler. Bu durumda uzayın tamamını araştıramazlar. Çoğu kez yerel minimum noktalara takılırlar ve oradan kurtulamazlar. Bu makalede meyve sineklerinin uçuşu Levy fonksiyonu olarak belirlenmiştir. Böylelikle sinekler çoğu kez kısa mesafe uçuşları yaparken nadir olarak arada uzun Levy uçuşları da yapar, böylelikle uzayın tamamına bu sayede ulaşıp yerel minimumlardan kurtulur, global en iyi noktaya ulaşırlar (Şekil 3)



**Şekil 3. a)** Standart MSO algoritmasında sineklerin yiyecek araştırması **b)** LMSO algoritmasında sineklerin yiyecek araştırması

Aşağıda standart MSO algoritmasındaki meyve sineğinin hareket yarıçapı ile Levy uçuşu yapan meyve sineğinin yarıçap formülleri sırası ile (5) ve (6) nolu denklemler ile ifade edilmiştir.

$$R_{MSO} = R \quad (5)$$

$$R_{LMSO} = R \cdot x^{-\lambda} \quad (6)$$

MSO algoritmasında meyve sinekleri seçilen R yarıçapı içerisinde rasgele hareket eder ve tanımlı uzayı tam olarak araştıramazlar, yerel minimuma takılıp global minimum çözüme ulaşamazlar. Oysa LMSO algoritmasında çoğunlukla kısa uçuşlar yapan meyve sinekleri nadir olarak uzun uçuşlar yaparak tanımlı uzayın değişik bölgelerine geçiş yapar ve tüm uzayı gezebilirler, buna bağlı olarak farklı daha iyi çözümlere ulaşabilirler. Böylelikle önerilen LMSO algoritmasında meyve sinekleri yerel minimum noktalara takılmadan global minimum çözüme ulaşabilirler.

### 3.3 Parametreler

Bu çalışmada görüntü sıkıştırma yaygın olarak kullanılan için LBG algoritması MSO tekniği ile birleştirilmiştir. Algoritma içerisindeki her bir meyve sineği bir çözüm kümesi olan vektör listesini temsil etmektedir  $C = \{C_1, C_2, \dots, C_{N_c}\}$ ,  $C_j \in \mathcal{R}^L$ ,  $j=1,2,\dots,N_c$ . Simülasyonlarda 0.122 bpp sıkıştırma oranı ve 8 vektöre sahip vektör listesi seçilmiş, önerilen LMSO tekniği literatürdeki 256x256 piksel büyüklüğünde 256 gri seviyeye sahip standart resimler üzerinde uygulanmıştır. Resim 4x4 piksel boyutuna sahip 4096 adet alt vektöre bölünmüştür. Dolayısı ile LBG tekniğinde resmi temsil eden vektörler de 4x4 piksel boyutundadır. Maksimum iterasyon sayısı 1500 ve R yarıçapı 2 piksel olarak belirlenmiştir. Levy dağılımındaki  $\lambda=1.5$  seçilmiştir. Levy fonksiyonu maksimum yarıçap  $R=2$  değerinden hızla  $R=0$  değerine indiğinden (6) nolu denklemdeki x yatay eksen  $x \in [1,9]$  olarak sınırlandırılmıştır. Tüm parametreler LMSO tekniğinin performansını en iyi yapan değerler olarak belirlenmiştir. Standart resimlerde varyans ve ortalama değerler kullanılarak (7) nolu değişim katsayıları elde edilmiştir. Simülasyonda meyve sineklerinin araştırmada kullandıkları yarıçap fonksiyonları Çizelge 1. de verilmiştir.

$$\text{Değişim Katsayısı (DK)} = \frac{\sqrt{\frac{1}{N} \sum_{i=1}^N (X_i - \text{ortalama})^2}}{\text{ortalama}} \quad (7)$$

**Çizelge 1.** Meyve sineklerinin simülasyonda kullandıkları yarıçap fonksiyonları

Yarıçap Çeşidi	Yarıçap Fonksiyonu	Parametreleri
Sabit	$Y=R$	$R=2$
Merdiven	$Y=R \Rightarrow Y=0$ (Merdiven şeklinde azalma)	$R=2$ Merdiven sayısı=10
Lineer Azalan	$Y = R - \frac{R \cdot (\text{iter. num.})}{N}$	$R=2$ $N=1500$
Levy	$Y=R \cdot x^{-p}$	$R=2$ $p=1.5, x \in [1,9]$
Üstel	$Y=R \cdot \exp[-a \cdot (\text{iter. num.})]$	$R=2$ $a=0.002$

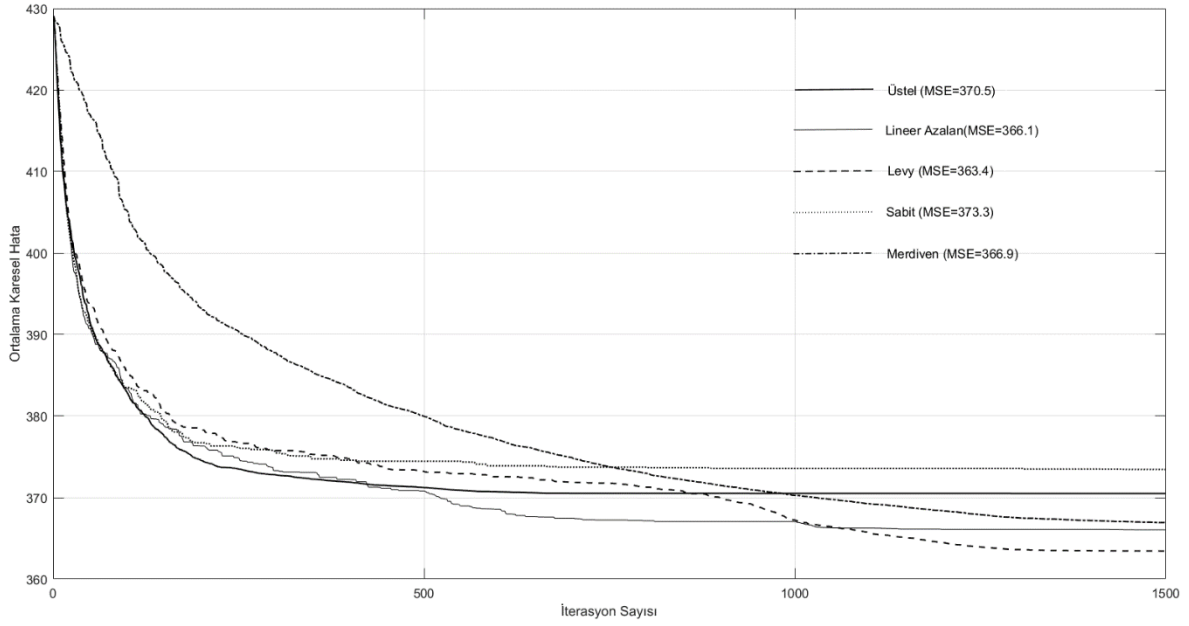
#### 4. Simulasyon Sonuçları

Simülasyonlarda 256x256 piksel büyüklüğünde 256 gri seviyeye sahip Çizelge 2 deki standart görüntüler kullanılmıştır. Meyve sineği için sabit, lineer azalan, Levy, rastgele, sinüs ve merdiven azalan yarıçap çeşitleri kullanılmış olup Çizelge 1 de gösterilmiştir.

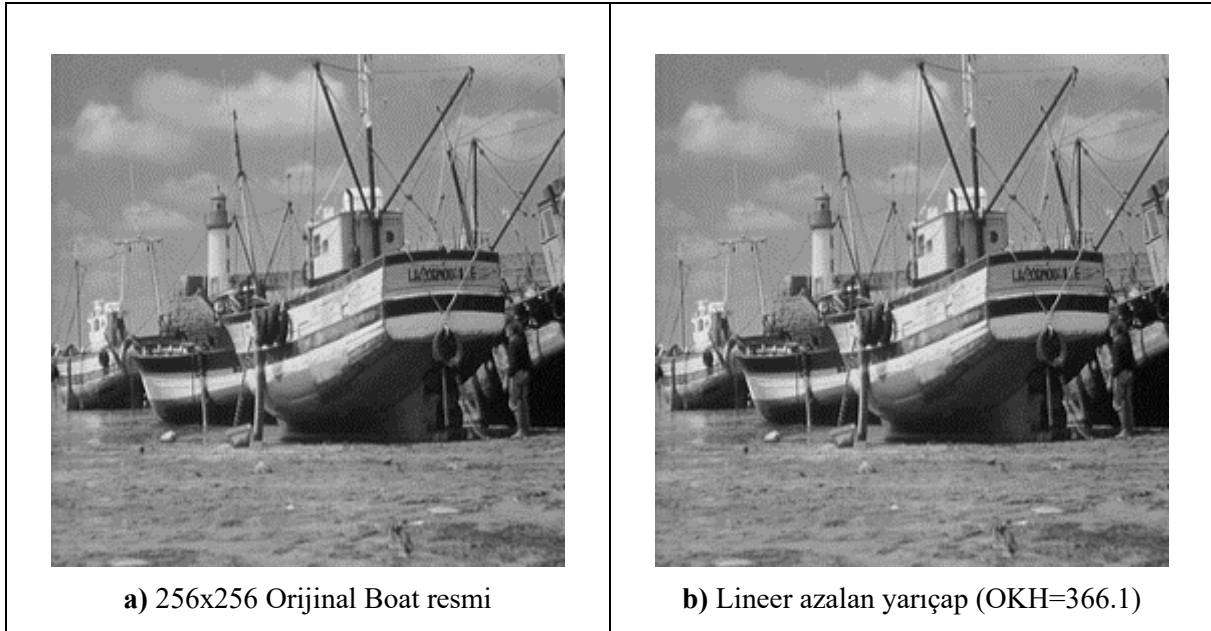
**Çizelge 2.** Sıkıştırma oranı=0.122 bpp için değişik yarıçap çeşitlerinde MSO algoritmasının OKH performansları

Kodlama Oranı 0.122 bpp	DK	(MSO) Sabit Yarıçap	Linear Azalan Yarıçap	Merdiven Azalan Yarıçap	Levy Azalan Yarıçap	Üstel Azalan Yarıçap
Peppers	0.447	335.8	326.5	323.4	318.1	320.2
Clock	0.308	306.3	299.9	298.7	293.4	295.1
Airplane	0.170	126.9	124.0	122.2	118.2	120.3
Boat	0.357	373.3	366.1	366.9	363.4	370.5
Chemical Plant	0.405	359.1	357.5	356.3	353.5	355.9
Cameraman	0.530	411.3	407.1	409.5	402.9	405.4
Barbara	0.454	348.7	343.4	340.2	335.0	338.4
Einstein	0.344	269.3	235.9	238.3	226.1	229.2
Couple	0.313	309.7	304.8	305.0	301.6	303.0
Lena	0.530	324.4	322.8	321.5	318.7	320.8
Moon	0.217	145.6	138.6	137.8	136.6	137.4
Baboon	0.300	432.3	423.2	422.6	415.3	418.7
Aerial	0.322	528.5	524.6	521.1	516.2	518.4

Şekil 4. te ise seçilen BOAT resmi için önerilen LMSO algoritmasının değişik yarıçap değerlerindeki sıkıştırılmış görüntü ve OKH değerleri görülmektedir.

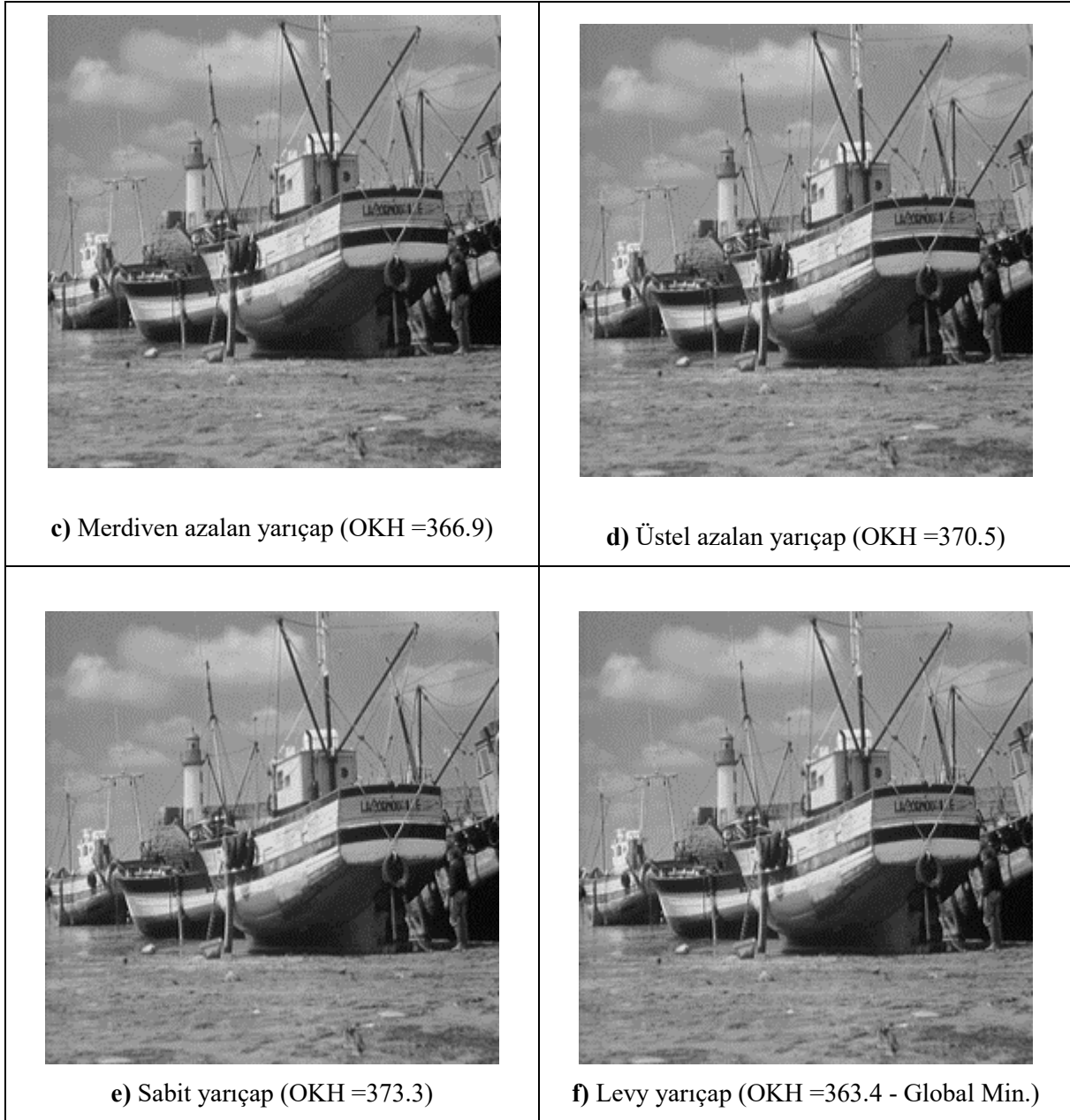


**Şekil 4.** 256x256 pixel boyutunda BOAT resmi için, 0.122bpp sıkıştırma oranında MSO algoritmasının değişik yarıçap değerlerinde OKH performans karşılaştırması



**Şekil 5.** 256x256 pixel boyutunda BOAT resmi için, 0.122bpp sıkıştırma oranında MSO algoritmasının değişik yarıçap değerlerinde OKH performans karşılaştırması





Şekil 5. Devamı

Önerilen LMSO tekniği 0.122bpp sıkıştırma oranında, 256x256 pixel boyutlarındaki değişik DK değerlerine sahip standart resimler üzerinde uygulanmış, sonuçlar Çizelge 2 de gösterilmiştir. Levy yarıçap değeri ile hareket eden meyve sineklerinin global en küçük hata değerine sahip vektör listesine ulaşmış olduğu ve diğer yarıçap fonksiyonlarına göre hareket eden meyve sineklerine göre daha iyi bir performans elde ettiği gözlenmiştir.

Çizelge 3 ile LMSO algoritması, değişik sıkıştırma oranlarında VN, PSO, ABO, YO ve değişik yarıçap fonksiyonlu MSO tekniği ile OKH performans kriterine göre karşılaştırılmış ve daha başarılı olduğu gözlenmiştir. 256x256 pixel boyutunda BOAT resmi için, 0.122bpp sıkıştırma oranında MSO algoritmasının değişik yarıçap değerlerinde OKH performans karşılaştırması Şekil 5 ile gösterilmiştir. MSO içerisinde lineer azalan yarıçap ile OKH=366.1, merdiven azalan yarıçap ile OKH=366.9, üstel

azalan yarıçap ile OKH=370.5, sabit yarıçap ile OKH=373.3 ve levy yarıçap ile global minimum hata OKH=363.4 değerine ulaşılmıştır.

**Çizelge 3.** Değişik sıkıştırma oranlarında orijinal BOAT resmi için algoritmaların OKH performans karşılaştırması

Sıkıştırma Oranı	Vektör Sayısı						MSO	MSO	MSO	MSO
		VN	PSO	ABO	YO	LMSO	Standart	Üstel	Lineer	Merdiven
0.133 bpp	4	492.5	489.2	488.9	489.8	485.9	490.0	489.2	488.7	488.8
0.203 bpp	8	369.54	368.0	368.5	369.8	363.4	367.3	370.5	366.1	366.9
0.281 bpp	16	287.8	285.0	285.7	284.5	280.0	284.4	284.5	282.2	283.0
0.375 bpp	32	229.6	226.7	226.9	225.0	222.1	228.0	227.7	224.5	225.2
0.500 bpp	64	183.6	180.2	181.0	179.1	176.5	182.7	180.0	179.9	180.2
0.688 bpp	128	145.4	143.9	144.2	143.5	139.7	144.5	144.1	142.3	144.0

## 5. Sonuç

Önerilen yeni LMSO tekniğinde meyve sineklerinin sorunsuz bir şekilde global minimum noktaya ulaşabilmesi için MSO, Levy Uçuşu fonksiyonu ile birleştirilmiştir. MSO algoritmasının en büyük sorunlarından biri de lokal minimum noktaya takılıp global minimuma ulaşamamasıdır. Çoğu zaman küçük, nadiren de büyük yarıçap değeri veren Levy Fonksiyonu yardımı ile meyve sineği lokal minimum noktaya hiç takılmayıp global minimum noktaya ulaşmışlardır. Önerilen LMSO tekniği değişik yarıçap fonksiyonlarında VN, PSO, ABO ve YO metotları ile karşılaştırılmış ve aynı sıkıştırma oranlarında MSE ölçütü kullanıldığında diğer metotlardan üstün olduğu Çizelge 3 te gösterilmiştir.

## KAYNAKLAR

- [1] Gray, R. M. Vector quantization. IEEE ASSP Magazine. 1984;1: 4-29.
- [2] Linde, Y., Buzo, A., & Gray, R. M. An algorithm for vector quantizer design. IEEE Transaction on Communications. 1980; 28(1): 84–95.
- [3] 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.
- [4] Patane, G. & Russo, M.. The enhanced LBG algorithm. Neural Networks. 2001; 14; 1219-1237.
- [5] Xu, W., Nandi, A.K., et.al. Novel vector quantiser design using reinforced learning. Signal Processing. 2005; 85; 1315–1333.
- [6] Tsai, C.W., Lee, C.Y., et.al. A fast VQ codebook generation algorithm via pattern reduction. Pattern Recognition Letters. 2009; 30: 653–660.
- [7] Karayiannis, N. B., & Pai, P. I. Fuzzy vector quantization algorithms and their application in image compression. IEEE Transactions in Image Processing. 1995;. 4(9): 1193–1201.
- [8] Karayiannis, N. B., & Bezdek, J. C. An integrated approach to fuzzy learning vector quantization and fuzzy c-means clustering. IEEE Transactions on Fuzzy Systems. 1997; 5(4): 622–628.
- [9] Tsekouras, G.E. A fuzzy vector quantization approach to image compression. Applied Mathematics and Computation. 2005;167: 539–560.
- [10] Kuo, R.C., Wang, H. S. et.al. Application of ant K-Means on clustering analysis. Computers and

- Mathematics with Applications. 2005; 50: 1709-1724.
- [11] Goldberg DE. Genetic algorithms in search, optimization and machine learning. Addison-Wesley; 1989.
- [12] Sun H., Lam, K.Y., et.al. Efficient vector quantization using genetic algorithm. 2005; 14: 203-211.
- [13] L. Zhang, B. Zheng and Z. Yang. Codebook design using genetic algorithm and its application to speaker identification. Electronics Letters. 2005; 41(10): 619-620.
- [14] Yang S. B., Constrained-storage multistage vector quantization based on genetic algorithms. Pattern Recognition. 2008; 41: 689 – 700.
- [15] Huang, H.C., Pan, J.S., et.al. Vector quantization based on genetic simulated annealing. Signal Processing. 2001; 81: 1513-1523.
- [16] Feng H.M., Chen C.Y., Ye, F. Evolutionary fuzzy particle swarm optimization vector quantization learning scheme in image compression. Expert Systems with Applications. 2007; 32: 213–222.
- [17] Horng, M. H., Jiang, T.W. Image vector quantization algorithm via honey bee mating optimization. Expert Systems with Applications. 2011; 38: 1382–1392.
- [18] Rani, M. L. P., Rao, G. S., & Rao, B. P. An efficient codebook generation using firefly Algorithm for optimum medical image compression. Journal of Ambient Intelligence and Humanized Computing, 2020;1-13.
- [19] Tsai, C.W., Tseng, S.P., et.al. PREACO: A fast ant colony optimization for codebook generation. Applied Soft Computing. 2013; 13: 3008–3020.
- [20] Dai, H., Zhao, G., Lu, J., & Dai, S. Comment and improvement on “A new fruit fly optimization algorithm: taking the financial distress model as an example”. Knowledge - Based Systems, 2014; 59: 159-160.
- [21] Li, H., Guo, S., Li, C., Sun, J., A hybrid annual power load forecasting model based on generalized regression neural network with fruit fly optimization algorithm. Knowledge Based Systems. 2013; 37: 378–387.
- [22] S.M. Lin, Analysis of service satisfaction in web auction logistics service using a combination of fruit fly optimization algorithm and general regression neural network, Neural Computational Applications. 2013; 7: 459–465.
- [23] Jiang, W., Wu, X., Gong, Y., Yu, W., & Zhong, X. Holt–Winters smoothing enhanced by fruit fly optimization algorithm to forecast monthly electricity consumption. Energy, 2020; 193: 116779.
- [24] Li, C., Xu, S., Li, W., L. Hu, L. A novel modified fruit fly optimization algorithm for designing the self-tuning proportional integral derivative controller. Journal of Convergence Information Technology. 2012; 7: 69–77.
- [25] Sheng, W., Bao, Y. Fruit fly optimization algorithm based fractional order fuzzy-PID controller for electronic throttle. Nonlinear Dynamics. 2013; 73: 611-619.
- [26] Chen, P.W, Lin, W.Y., Huang, T.H., Pan, W.T. Using fruit fly optimization algorithm optimized grey model neural network to perform satisfaction analysis for e-business service, Applied Mathematics and Information Sciences. 2013; 7(21): 459–465.
- [27] Meng, T., & Pan, Q. K. An improved fruit fly optimization algorithm for solving the multidimensional knapsack problem. Applied Soft Computing, 2017; 50: 79-93.
- [28] Yuan, X., Dai, X., Zhao, J., He, Q. On a novel multi-swarm fruit fly optimization algorithm and its application. Applied Mathematics and Computation. 2014; 233: 260–271.
- [29] Wang, L., Xiong, Y., Li, S., & Zeng, Y. R. New fruit fly optimization algorithm with joint search strategies for function optimization problems. Knowledge-Based Systems, 2019; 176: 77-96.
- [30] Sheng, W, Bao, Y., Fruit fly optimization algorithm based fractional order fuzzy – pid controller for electronic throttle, Nonlinear Dynam. 2013;73 (1–2) : 611–619.
- [31] Li, J. Q., Pan, Q. K., & Mao, K. A hybrid fruit fly optimization algorithm for the realistic hybrid flowshop rescheduling problem in steelmaking systems. IEEE Transactions on Automation Science and Engineering, 2015; 13(2): 932-949.
- [32] Ingaleswar, S., Dharwadkar, N. V., & Jayadevappa, D. Water chaotic fruit fly optimization-based deep convolutional neural network for image watermarking using wavelet transform. Multimedia Tools and Applications, 2021; 1-25.

- [33] Kumar, S. N., Fred, A. L., Kumar, H. A., et.al. BAT Optimization - Based Vector Quantization Algorithm for Compression of CT Medical Images. In ICTMI 2017 ( pp. 53 – 64 ). Springer Singapore. 2019
- [34] Metzler, R., Aleksei, V. C. et.al. Some fundamental aspects of Lévy Flights. *Chaos, Solitons and Fractals*. 2007; 34; 129–142.
- [35] Yang, X.-S. Firefly algorithm, Lévy flights and global optimization. *Research and Development in Intelligent Systems XXVI* (Eds M. Bramer, R. Ellis, M. Petridis), Springer. 2010; 209-218.
- [36] Chiranjeevi, K., Jena, U. R. Image compression based on vector quantization using cuckoo search optimization technique. *Ain Shams Engineering Journal*. 2018; 9(4): 1417-1431.
- [37] Fu, Y., Zhou M., Guo, X., Qi, L. Stochastic multi-objective integrated disassembly-reprocessing reassembly scheduling via fruit fly optimization algorithm. *Journal of Cleaner Production*. 2021; 278: 123364
- [38] Zhang X., Xu, Y., Caiyang Yu, C., et.al. Gaussian mutational chaotic fruit fly - built optimization and feature selection. *Expert Systems With Applications*. 2020; 141; 112976.
- [39] Wang, L., Xiong Y., Li, S., et. al. New fruit fly optimization algorithm with joint search strategies for function optimization problems. *Knowledge-Based Systems*, 2019; 176: 77–96.
- [40] Ding, G., Dong, F., Zou, H. Fruit fly optimization algorithm based on a hybrid adaptive cooperative learning and its application in multilevel image thresholding. *Applied Soft Computing Journal*, 84 (2019) 105704.