

**T.C.  
İNÖNÜ ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ**

**SEZGİSEL YÖNTEMLERDE ALTIN ORAN**

**MURAT DEMİR**

**DOKTORA TEZİ**

**ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI**

**MALATYA**

**Temmuz 2015**

Tezin başlığı : “**Sezgisel Yöntemlerde Altın Oran**”

Tezi Hazırlayan: **Murat DEMİR**

Sınav tarihi: **10.07.2015**

Yukarıda adı geçen tez jürimizce değerlendirilerek Elektrik-Elektronik Mühendisliği Anabilim Dalında Doktora Tezi olarak kabul edilmiştir.

**Sınav Jürisi Üyeleri**

Prof. Dr. Mehmet KAYA  
( Jüri Başkanı )

Fırat Üniversitesi

Prof. Dr. Ali KARCI  
( Danışman )

İnönü Üniversitesi

Prof. Dr. Nusret TAN  
( Üye )

İnönü Üniversitesi

Prof. Dr. Serdar Ethem HAMAMCI  
( Üye )

İnönü Üniversitesi

Yrd. Doç. Dr. Burhan ERGEN  
( Üye )

Fırat Üniversitesi

Prof. Dr. Alaattin ESEN  
Enstitü Müdürü

## **ONUR SÖZÜ**

Doktora Tezi olarak sunduđum “ **Sezgisel Yöntemlerde Altın Oran**” başlıklı bu çalışmanın, bilimsel ahlak ve geleneklere aykırı düşecek bir yardıma başvurmaksızın tarafımdan yazıldığını ve yararlandığım bütün kaynakların, hem metin içinde hem de kaynakçada yöntemine uygun biçimde gösterilenlerden oluştuđunu belirtir, bunu onurumla doğrularım.

**Murat DEMİR**

# ÖZET

Doktora Tezi

## SEZGİSEL YÖNTEMLERDE ALTIN ORAN

Murat DEMİR

İnönü Üniversitesi  
Fen Bilimleri Enstitüsü  
Elektrik-Elektronik Mühendisliği Anabilim Dalı

129+xi sayfa

2015

Danışman: Prof. Dr. Ali KARCI

Sezgisel algoritmalar, çözüme yaklaşımsal yöntemlerle giden ve doğada var olan sistemleri taklit eden yöntemlerdir. Genellikle çözümü barındıracak popülasyonları temsil eden matrislerle hedef çözüme ulaşılmaya çalışılır. Bir çok farklı metot kullanan sezgisel yöntem vardır. Bu tez çalışmasında optimizasyon amaçlı kullanılan sezgisel algoritmalar, Yapay atom algoritması, Parçacık sürü optimizasyon algoritması, Ateş böceği algoritması, Yarasa algoritması ve Fidan gelişim algoritması yöntemleri veri kümeleme üzerine uygulanmıştır. Sezgisel algoritmalarda genellikle, çözüm için oluşturulan başlangıç popülasyonları rastgele oluşturulur. Bu çalışmada, Fidan gelişim Algoritması yönteminin başlangıç popülasyonunu oluşturmadaki tekniğinden faydalanılmıştır ve genellikle doğruluk oranını daha iyi değerlere taşıdığı görülmüştür.

Altın oran değeri, kainatta birçok yerde eşsiz bir şekilde yaratma sanatını gösteren bir orandır. Bundan hareketle yapılan bu tez çalışmasında, veri kümeleme için, uygulama yazılımları geliştirilirken, altın oranın parametrelerdeki etkileri incelenmiştir. Sezgisel algoritmalarda hesapsal parametreler genellikle rastgele seçilir. Bu tez çalışmasında uygulama yazılımlarında parametreler, öncelikle rastgele denenmiştir. Daha sonra parametreler, altın oran değeri, altın oran değerine yakın değerler ve altın oran değerine uzak değerler etrafında denenmiştir. Uygulama yazılımlarında, parametrelerin tanımlı olduğu aralıklarda altın oran değerinin genelde en iyi sonuçları verdiği görülmüştür.

Veri kümeleme için kullanılan yöntemlerde, sezgisel algoritmalar ve altın orana bir arada şu ana kadar rastlanmamıştır. Bu tez çalışmasında altın oran ve sezgisel algoritmalar başarılı bir şekilde veri kümeleme üzerine uygulanmıştır.

**ANAHTAR KELİMELELER:** Sezgisel Algoritmalar, Altın Oran ve Veri Kümeleme

## **ABSTRACT**

PhD. Thesis

**GOLDEN RATIO ON HEURISTIC METHODS**

Murat DEMİR

Inonu University  
Graduate School of Natural and Applied Sciences  
Department of Electrical-Electronics Engineering

129+xi pages

2015

Supervisor: Professor Dr. Ali KARCI

Heuristic algorithms which mimic the natural processes, try to find approximate solutions. Generally, they try to approximate to solutions by using matrices which represent the candidate solutions. There are a lot of heuristic methods which use more than one method. In this thesis, the heuristic algorithms which are used at the aim of optimization, such as Artificial Atom Algorithm, Particle Swarm Optimization, Firefly Algorithm and Bat Algorithms were applied to clustering problems. Generally, initial populations are generated for heuristic algorithms. In this thesis, initial population generating method for Sapling Growing-up Algorithm was used and it was seen that this method made success rate increase.

The golden ratio depicts unique art of creation in nature. By this way, the effect of golden ratio on the parameters of heuristic algorithms was investigated by developing software. In general, the values of parameters in heuristic algorithms are generated randomly. In this thesis, the random values were tried for parameters firstly. After that the values equal to golden ratio, near to golden ratio and far away to golden ratio were tried for parameters. In general, it was seen that the values of parameters near or equal to golden ratio yielded better results.

There were no studies used golden ratio and heuristic algorithms together. In this thesis, golden ratio and heuristic algorithms were applied to clustering problems successfully.

**KEYWORDS:** Heuristic Algorithms, Golden Ratio and Data Clustering

## TEŐEKKÜR

Bu alıőmanın her aőamasında yardım, öneri ve desteęini esirgemeden beni yönlendiren danıőman hocam Sayın Do. Dr. Ali KARCI' ya,

Bugünlere gelmemde büyük emeęi olan ve desteklerini hiçbir zaman esirgemeyen sevgili annem Őahide DEMİR ve babam Adem DEMİR' e,

Tüm bu süreçte bana sürekli destek olan sevgili eőim Ülkü DEMİR' e ve sevgili kızım Rana Nur DEMİR' e,

Tez alıőmalarım sırasında hayırlı temennilerini eksik etmeyen tüm dostlarıma teşekkürlerimi sunarım.

## İÇİNDEKİLER

<b>ÖZET</b> .....	i
<b>ABSTRACT</b> .....	ii
<b>TEŞEKKÜR</b> .....	iii
<b>İÇİNDEKİLER</b> .....	iv
<b>ŞEKİLLER LİSTESİ</b> .....	vi
<b>TABLolar LİSTESİ</b> .....	viii
<b>1. GİRİŞ</b> .....	1
<b>2. SEZGİSEL YÖNTEMLER</b> .....	4
2.1. Yapay Atom Algoritması ( A <sup>3</sup> : Artificial Atom Algorithm ).....	5
2.2. Parçacık Sürü Optimizasyonu Algoritması.....	9
2.2.1. Parçacık sürü optimizasyonu.....	9
2.2.2. Parçacık sürü optimizasyon algoritması.....	10
2.2.3. Parçacık sürü optimizasyonu çözüm yöntemi.....	11
2.3. Ateş Böceği Algoritması.....	13
2.3.1. Ateş böcekleri.....	13
2.3.2. Yapay ateş böcekleri.....	13
2.3.3. Işık şiddeti.....	13
2.3.4. Çekicilik.....	14
2.4. Yarasa Algoritması.....	16
2.4.1. Yarasalar.....	16
2.4.2. Yarasa algoritmasının adımları.....	17
2.4.3. Yapay yarasaların hareketleri.....	17
2.4.4. Gürültü ve sinyal yayma oranı.....	18
2.5. Fidan Gelişim Algoritması.....	19
<b>3. ALTIN ORAN</b> .....	23
<b>4. VERİ KÜMELEME</b> .....	26
4.1. Uzaklık Matrisi Hesaplama Yöntemleri.....	27
4.1.1. Öklid uzaklık ölçüsü.....	27
4.1.2. Manhattan uzaklık ölçüsü.....	28
4.1.3. Minkowski uzaklık ölçüsü.....	28
<b>5. ÖNERİLEN YÖNTEMLER ve UYGULAMA SONUÇLARI</b> .....	29
5.1. Yapay Atom Algoritması Uygulama Yazılımı ve Sonuçları.....	34
5.2. Parçacık Sürü Optimizasyon Algoritması Uygulama Yazılımı ve Sonuçları.....	45
5.3. Ateş Böceği Algoritması Uygulama Yazılımı ve Sonuçları.....	74
5.4. Yarasa Algoritması Uygulama Yazılımı ve Sonuçları.....	93
<b>6. TARTIŞMA ve SONUÇ</b> .....	122

<b>KAYNAKLAR</b> .....	125
<b>ÖZGEÇMİŞ</b> .....	<b>Hata! Yer işareti tanımlanmamış.</b>



## ŞEKİLLER LİSTESİ

Şekil 1.1. Minimum ve maksimum değer gösterimi .....	1
Şekil 1.2. Yerel ve global minimum ve maksimum değerlerin gösterimi .....	2
Şekil 2.1. Kovalent bağ temsili .....	6
Şekil 2.2. İyonik bağ temsili .....	6
Şekil 2.3. Yapay atom algoritması elektron ortaklığının temsili .....	7
Şekil 2.4. Parçacık yaklaşımı şekilsel temsili .....	10
Şekil 2.5. Fidan gelişim algoritması başlangıç fidanları .....	20
Şekil 2.6. Fidan gelişim algoritması başlangıç popülasyonu $k=2$ için yeni fidanlar .	20
Şekil 2.7. Fidan gelişim algoritması başlangıç popülasyonu $k=3$ için yeni fidanlar .	20
Şekil 2.8. Fidan gelişim algoritması başlangıç popülasyonu $k=4$ için yeni fidanlar .	21
Şekil 3.1. Doğru parçasında altın oran .....	23
Şekil 3.2. Altın dikdörtgen geometrik temsili .....	23
Şekil 3.3. Altın dikdörtgende spiral yapısının temsili .....	24
Şekil 3.4. Doğadaki altın spirale örnekler .....	25
Şekil 4.1. Kümeleme adımları blok temsili .....	26
Şekil 4.2. Verileri kümelemenin grafiksel temsili .....	27
Şekil 5.1. Manhattan uzaklıkları hesabının şekilsel temsili .....	32
Şekil 5.2. Yapay atom algoritması tüm çözüm durumları için ortalama doğruluk oranları grafik temsili .....	44
Şekil 5.3. Parçacık sürü optimizasyon algoritması uygulama yazılımı, ilk 11 adımın tüm durumlarının doğruluk oranlarının grafik temsili .....	70
Şekil 5.4. Parçacık sürü optimizasyon algoritmasında fidan gelişim algoritmasının etkilerinin grafik temsili .....	73
Şekil 5.5. Popülasyon değerleri ile eğitim verilerinin farklarının temsili .....	75
Şekil 5.6. Ateş böceği algoritması ilk 6 adımın ortalama doğruluk oranlarının grafiği .....	80
Şekil 5.7. Ateş böceği algoritması Adım 7, 8, 9 ve 10 için ortalama doğruluk oranlarının grafiği .....	83
Şekil 5.8. Ateş böceği algoritması Adım 11, 12 ve 13 için ortalama doğruluk oranlarının grafiği .....	85
Şekil 5.9. Ateş böceği algoritması Adım 14,15 ve 16 için ortalama doğruluk oranlarının grafiği .....	87
Şekil 5.10. Fidan gelişim algoritması gruplanmış başlangıç popülasyonu temsili ....	87
Şekil 5.11. Ateş böceği algoritması Adım 17,18, 19 ve 20 için ortalama doğruluk oranlarının grafiği .....	90
Şekil 5.12. Ateş böceği algoritması Adım 21, 22 ve 23 için ortalama doğruluk oranlarının grafiği .....	92
Şekil 5.13. Yarasa algoritması Adım 1 için sonuç grafiği .....	96
Şekil 5.14. Yarasa algoritması Adım 2 için sonuç grafiği .....	97
Şekil 5.15. Yarasa algoritması Adım 3 için sonuç grafiği .....	98
Şekil 5.16. Yarasa algoritması Adım 4 için sonuç grafiği .....	99
Şekil 5.17. Yarasa algoritması Adım 5 için sonuç grafiği .....	100
Şekil 5.18. Yarasa algoritması Adım 6 için sonuç grafiği .....	101
Şekil 5.19. Yarasa algoritması Adım 1-Adım 6 arası ortalama doğruluk oranları grafik temsili .....	101
Şekil 5.20. Yarasa algoritması Adım 7 için sonuç grafiği .....	102
Şekil 5.21. Yarasa algoritması Adım 8 için sonuç grafiği .....	103
Şekil 5.22. Yarasa algoritması Adım 9 için sonuç grafiği .....	104

Şekil 5.23. Yarasa algoritması Adım 7-Adım 9 arası ortalama doğruluk oranları grafik temsili .....	104
Şekil 5.24. Yarasa algoritması Adım 10 için sonuç grafiği .....	105
Şekil 5.25. Yarasa algoritması Adım 11 için sonuç grafiği .....	106
Şekil 5.26. Yarasa algoritması Adım 10 ve Adım 11 ortalama doğruluk oranları grafik temsili .....	107
Şekil 5.27. Yarasa algoritması Adım 12 için sonuç grafiği .....	108
Şekil 5.28. Yarasa algoritması Adım 13 için sonuç grafiği .....	109
Şekil 5.29. Yarasa algoritması Adım 14 için sonuç grafiği .....	110
Şekil 5.30. Yarasa algoritması Adım 15 için sonuç grafiği .....	111
Şekil 5.31. Yarasa algoritması Adım 12-Adım 15 arası ortalama doğruluk oranları grafik temsili .....	111
Şekil 5.32. Yarasa algoritması Adım 16 için sonuç grafiği .....	112
Şekil 5.33. Yarasa algoritması Adım 17 için sonuç grafiği .....	113
Şekil 5.34. Yarasa algoritması Adım 18 için sonuç grafiği .....	114
Şekil 5.35. Yarasa algoritması Adım 17 ve Adım 18 ortalama doğruluk oranları grafik temsili .....	115
Şekil 5.36. Yarasa algoritması Adım 19 için sonuç grafiği .....	116
Şekil 5.37. Yarasa algoritması Adım 20 için sonuç grafiği .....	117
Şekil 5.38. Yarasa algoritması Adım 21 için sonuç grafiği .....	118
Şekil 5.39. Yarasa algoritması Adım 22 için sonuç grafiği .....	120
Şekil 5.40. Yarasa algoritması Adım 19-Adım 22 arası ortalama doğruluk oranları grafik temsili .....	120
Şekil 6.1. Iris-Plants veri setini kullanan uygulama yazılımlarının en yüksek sonuçlarının karşılaştırılma grafiği.....	122

## TABLolar LİSTESİ

Tablo 2.1. Yapay atom algoritması çözüm matrisi temsili .....	6
Tablo 2.2. Yapay atom algoritması popülasyon temsili.....	6
Tablo 2.3. Yapay atom algoritması kovalent bağ temsili.....	7
Tablo 2.4. Yapay atom algoritması iyonik bağ temsili .....	7
Tablo 2.5. Fidan gelişim algoritması başlangıç popülasyonu oluşturma hesapsal temsili .....	21
Tablo 3.1. Gezegenler arası nispi altın oran değerleri (milyon km.) .....	25
Tablo 5.1. Başlangıç popülasyonlarının genel temsili .....	30
Tablo 5.2. $p_i$ değer matrisinin temsili.....	31
Tablo 5.3. $p_i$ değerlerinin toplamının matris temsili .....	31
Tablo 5.4. Örnek uygulama BEST matris değerleri.....	32
Tablo 5.5. Örnek uygulama manhattan uzaklık değerleri örnek temsili .....	33
Tablo 5.6. Yapay atom algoritması uygulama yazılımı BEST matris temsili .....	34
Tablo 5.7. Iris Dataset veri örnekleri temsili.....	34
Tablo 5.8. Kovalent bağ çözüm temsili .....	35
Tablo 5.9. İyonik bağ çözüm temsili.....	36
Tablo 5.10. Yapay atom algoritması uygulama yazılımı farklar matrisinin temsili ..	36
Tablo 5.11. Yapay atom algoritması uygulama yazılımı farklar matrisinin toplamının temsili .....	36
Tablo 5.12. Yapay atom algoritması Adım 1 için 1 özellik değiştiğinde sonuçlar ....	37
Tablo 5.13. Yapay atom algoritması Adım 1 için 2 özellik değiştiğinde sonuçlar ...	37
Tablo 5.14. Yapay atom algoritması Adım 1 için 3 özellik değiştiğinde sonuçlar ....	38
Tablo 5.15. Yapay atom algoritması Adım 1 için 4 özellik değiştiğinde sonuçlar ....	38
Tablo 5.16. Yapay atom algoritması Adım 2 için 1 özellik değiştiğinde sonuçlar ....	38
Tablo 5.17. Yapay atom algoritması Adım 2 için 2 özellik değiştiğinde sonuçlar ....	39
Tablo 5.18. Yapay atom algoritması Adım 2 için 3 özellik değiştiğinde sonuçlar ....	39
Tablo 5.19. Yapay atom algoritması Adım 2 için 4 özellik değiştiğinde sonuçlar ....	39
Tablo 5.20. Yapay atom algoritması Adım 3 için 1 özellik değiştiğinde sonuçlar ....	40
Tablo 5.21. Yapay atom algoritması Adım 3 için 2 özellik değiştiğinde sonuçlar ....	40
Tablo 5.22. Yapay atom algoritması Adım 3 için 3 özellik değiştiğinde sonuçlar ....	40
Tablo 5.23. Yapay atom algoritması Adım 3 için 4 özellik değiştiğinde sonuçlar ....	41
Tablo 5.24. Yapay atom algoritması Adım 4 için 1 özellik değiştiğinde sonuçlar ....	41
Tablo 5.25. Yapay atom algoritması Adım 4 için 2 özellik değiştiğinde sonuçlar ....	41
Tablo 5.26. Yapay atom algoritması Adım 4 için 3 özellik değiştiğinde sonuçlar ....	42
Tablo 5.27. Yapay atom algoritması Adım 4 için 4 özellik değiştiğinde sonuçlar ....	42
Tablo 5.28. Yapay atom algoritması Adım 5 için 1 özellik değiştiğinde sonuçlar ....	42
Tablo 5.29. Yapay atom algoritması Adım 5 için 2 özellik değiştiğinde sonuçlar ....	43
Tablo 5.30. Yapay atom algoritması Adım 5 için 3 özellik değiştiğinde sonuçlar ....	43
Tablo 5.31. Yapay atom algoritması Adım 5 için 4 özellik değiştiğinde sonuçlar ....	43
Tablo 5.32. Parçacık sürü optimizasyon algoritması uygulama yazılımı BEST matris temsili .....	45
Tablo 5.33. Parçacık sürü optimizasyon algoritması uygulama yazılımı farklar matrisinin temsili.....	45
Tablo 5.34. Parçacık sürü optimizasyon algoritması uygulama yazılımı farklar matrisinin toplamının temsili .....	46
Tablo 5.35. Parçacık sürü optimizasyonu Adım 1 a. için sonuçlar.....	47
Tablo 5.36. Parçacık sürü optimizasyonu Adım 1 b. için sonuçlar .....	48
Tablo 5.37. Parçacık sürü optimizasyonu Adım 1 c. için sonuçlar.....	48



Tablo 5.88. Parçacık sürü optimizasyonu Adım 11 d. için sonuçlar .....	68
Tablo 5.89. Parçacık sürü optimizasyonu Adım 11 e. için sonuçlar.....	69
Tablo 5.90. Parçacık Sürü Optimizasyon algoritması uygulama yazılımı ilk 11 adımın tüm durumlarının doğruluk tabloları .....	69
Tablo 5.91. Parçacık sürü optimizasyonu Adım 12 a. için sonuçlar.....	71
Tablo 5.92. Parçacık sürü optimizasyonu Adım 12 b. için sonuçlar .....	71
Tablo 5.93. Parçacık sürü optimizasyonu Adım 12 c. için sonuçlar.....	72
Tablo 5.94. Parçacık sürü optimizasyonu Adım 12 d. için sonuçlar .....	72
Tablo 5.95. Parçacık sürü optimizasyonu Adım 12 e. için sonuçlar.....	72
Tablo 5.96. Parçacık sürü optimizasyon algoritması için fidan gelişim algoritması öncesi ve sonrası durum mukayesesi .....	73
Tablo 5.97. Wisconsin Diagnosis Breast Cancer veri tabanından alınan ham bilgilerin temsili .....	74
Tablo 5.98. Ateş böceği algoritması uygulama yazılımında işlenecek olan verinin yeni hali.....	75
Tablo 5.99. Ateş böceği algoritması uygulama yazılımı farklar matrisinin temsili...	76
Tablo 5.100. Ateş böceği algoritması uygulama yazılımı farklar matrisinin toplamının temsili .....	76
Tablo 5.101. Ateş böceği algoritması uygulama yazılımı BEST matris temsili.....	77
Tablo 5.102. Ateş böceği algoritması Adım 1 için sonuçlar.....	77
Tablo 5.103. Ateş böceği algoritması Adım 2 için sonuçlar.....	78
Tablo 5.104. Ateş böceği algoritması Adım 3 için sonuçlar.....	78
Tablo 5.105. Ateş böceği algoritması Adım 4 için sonuçlar.....	79
Tablo 5.106. Ateş böceği algoritması Adım 5 için sonuçlar.....	79
Tablo 5.107. Ateş böceği algoritması Adım 6 için sonuçlar.....	80
Tablo 5.108. Ateş böceği algoritması Adım 7 için sonuçlar.....	81
Tablo 5.109. Ateş böceği algoritması Adım 8 için sonuçlar.....	81
Tablo 5.110. Ateş böceği algoritması Adım 9 için sonuçlar.....	82
Tablo 5.111. Ateş böceği algoritması Adım 10 için sonuçlar.....	82
Tablo 5.112. Ateş böceği algoritması Adım 11 için sonuçlar.....	83
Tablo 5.113. Ateş böceği algoritması Adım 12 için sonuçlar.....	84
Tablo 5.114. Ateş böceği algoritması Adım 13 için sonuçlar.....	84
Tablo 5.115. Ateş böceği algoritması Adım 14 için sonuçlar.....	85
Tablo 5.116. Ateş böceği algoritması Adım 15 için sonuçlar.....	86
Tablo 5.117. Ateş böceği algoritması Adım 16 için sonuçlar.....	86
Tablo 5.118. Ateş böceği algoritması Adım 17 için sonuçlar.....	88
Tablo 5.119. Ateş böceği algoritması Adım 18 için sonuçlar.....	88
Tablo 5.120. Ateş böceği algoritması Adım 19 için sonuçlar.....	89
Tablo 5.121. Ateş böceği algoritması Adım 20 için sonuçlar.....	89
Tablo 5.122. Ateş böceği algoritması Adım 21 için sonuçlar.....	90
Tablo 5.123. Ateş böceği algoritması Adım 22 için sonuçlar.....	91
Tablo 5.124. Ateş böceği algoritması Adım 23 için sonuçlar.....	91
Tablo 5.125. Yarasa algoritması uygulama yazılımı BEST matris temsili.....	93
Tablo 5.126. Yarasa algoritması uygulama yazılımı farklar matrisinin temsili.....	94
Tablo 5.127. Yarasa algoritması uygulama yazılımı farklar matrisinin toplamının temsili .....	94
Tablo 5.128. Yarasa algoritması Adım 1 için başlangıç koşulları .....	95
Tablo 5.129. Yarasa algoritması Adım 1 uygulama sonuçları.....	95
Tablo 5.130. Yarasa algoritması Adım 2 için başlangıç koşulları .....	96
Tablo 5.131. Yarasa algoritması Adım 2 uygulama sonuçları.....	96

Tablo 5.132. Yarasa algoritması Adım 3 için başlangıç koşulları .....	97
Tablo 5.133. Yarasa algoritması Adım 3 uygulama sonuçları .....	97
Tablo 5.134. Yarasa algoritması Adım 4 için başlangıç koşulları .....	98
Tablo 5.135. Yarasa algoritması Adım 4 uygulama sonuçları .....	98
Tablo 5.136. Yarasa algoritması Adım 5 için başlangıç koşulları .....	99
Tablo 5.137. Yarasa algoritması Adım 5 uygulama sonuçları .....	99
Tablo 5.138. Yarasa algoritması Adım 6 için başlangıç koşulları .....	100
Tablo 5.139. Yarasa algoritması Adım 6 uygulama sonuçları .....	100
Tablo 5.140. Yarasa algoritması Adım 7 için başlangıç koşulları .....	101
Tablo 5.141. Yarasa algoritması Adım 7 uygulama sonuçları .....	102
Tablo 5.142. Yarasa algoritması Adım 8 için başlangıç koşulları .....	102
Tablo 5.143. Yarasa algoritması Adım 8 uygulama sonuçları .....	103
Tablo 5.144. Yarasa algoritması Adım 9 için başlangıç koşulları .....	103
Tablo 5.145. Yarasa algoritması Adım 9 uygulama sonuçları .....	104
Tablo 5.146. Yarasa algoritması Adım 10 için başlangıç koşulları .....	105
Tablo 5.147. Yarasa algoritması Adım 10 uygulama sonuçları .....	105
Tablo 5.148. Yarasa algoritması Adım 11 için başlangıç koşulları .....	106
Tablo 5.149. Yarasa algoritması Adım 11 uygulama sonuçları .....	106
Tablo 5.150. Yarasa algoritması Adım 12 için başlangıç koşulları .....	107
Tablo 5.151. Yarasa algoritması Adım 12 uygulama sonuçları .....	108
Tablo 5.152. Yarasa algoritması Adım 13 için başlangıç koşulları .....	108
Tablo 5.153. Yarasa algoritması Adım 13 uygulama sonuçları .....	109
Tablo 5.154. Yarasa algoritması Adım 14 için başlangıç koşulları .....	109
Tablo 5.155. Yarasa algoritması Adım 14 uygulama sonuçları .....	110
Tablo 5.156. Yarasa algoritması Adım 15 için başlangıç koşulları .....	110
Tablo 5.157. Yarasa algoritması Adım 15 uygulama sonuçları .....	111
Tablo 5.158. Yarasa algoritması Adım 16 için başlangıç koşulları .....	112
Tablo 5.159. Yarasa algoritması Adım 16 uygulama sonuçları .....	112
Tablo 5.160. Yarasa algoritması Adım 17 için başlangıç koşulları .....	113
Tablo 5.161. Yarasa algoritması Adım 17 uygulama sonuçları .....	113
Tablo 5.162. Yarasa algoritması Adım 18 için başlangıç koşulları .....	114
Tablo 5.163. Yarasa algoritması Adım 18 uygulama sonuçları .....	114
Tablo 5.164. Yarasa algoritması Adım 19 için başlangıç koşulları .....	115
Tablo 5.165. Yarasa algoritması Adım 19 uygulama sonuçları .....	116
Tablo 5.166. Yarasa algoritması Adım 20 için başlangıç koşulları .....	117
Tablo 5.167. Yarasa algoritması Adım 20 uygulama sonuçları .....	117
Tablo 5.168. Yarasa algoritması Adım 21 için başlangıç koşulları .....	118
Tablo 5.169. Yarasa algoritması Adım 21 uygulama sonuçları .....	118
Tablo 5.170. Yarasa algoritması Adım 22 için başlangıç koşulları .....	119
Tablo 5.171. Yarasa algoritması Adım 22 uygulama sonuçları .....	119
Tablo 6.1. Iris-Plants veri setini kullanan uygulama yazılımlarının en yüksek sonuçlarının karşılaştırılması .....	122

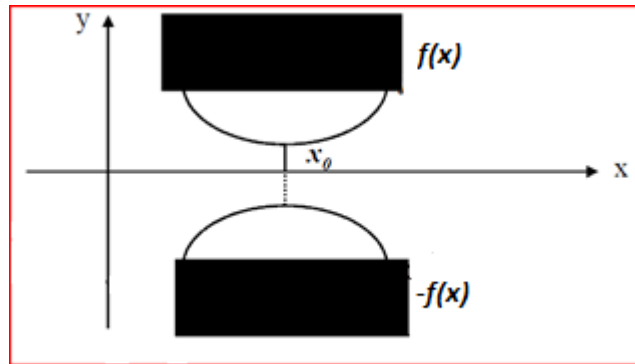
## 1. GİRİŞ

Çok hızlı bir şekilde değişim gösteren teknolojilerin, giderek daha da karmaşıklaşan sistemlerin beraberinde getirmiş olduğu problemlerin, matematiksel ya da analitik bilinen klasik yöntemlerle çözülmesi güçleşmiştir. Bu sebeplerden dolayı optimizasyon kavramı son zamanlarda en çok bahsi geçen alanlardan olmuştur. Tüm bilim dallarının içerisinde optimizasyon kavramı yerini almıştır.

Optimum kelimesi Latince kökenli bir kelimedir. Olabilmesi mümkün olan en ideal manasındadır. Optimizasyon ise, bir problemin en iyi çözümünü veya tasarımını bulma işlemi olarak tanımlanabilir. Bir başka deyişle mümkün olan alternatifler arasında en iyisini seçebilme işlemidir. Mühendislik alanında, tasarım, imalat ve bakım aşamalarında kararlar alınır. Bu kararların amacı, gerekli çaba, sermaye, malzeme veya teknolojinin minimum tutulması ve faydanın maksimum yapılabilmesidir. Bu bakış açısıyla optimizasyon, hedeflenen amacı maksimum veya minimum yapacak şartları bulmak olarak ifade edilebilir [1]. Belirli sınırlamaları sağlayacak şekilde, bilinmeyen parametrelerin değerlerinin hesaplanmasını içeren herhangi bir problem, optimizasyon problemi olarak adlandırılabilir [2].

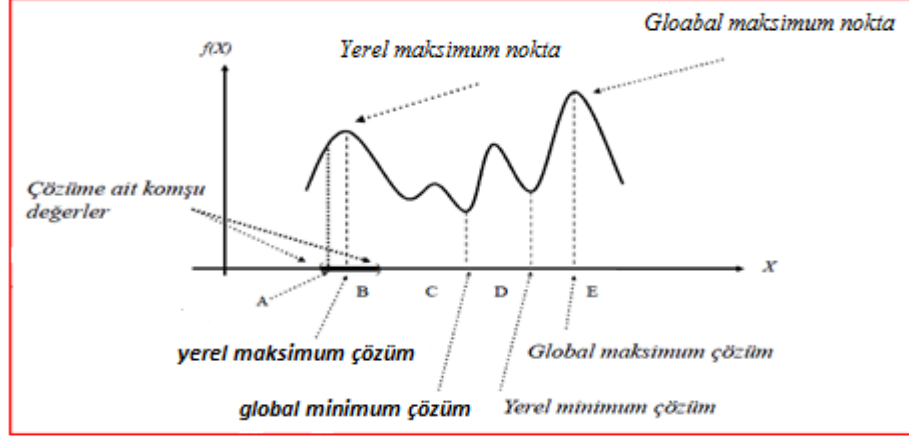
Optimizasyon alanındaki en önemli gelişmeler 18. yüzyılda Newton ve Lagrange tarafından yapılmıştır. Optimizasyon sürecinde fonksiyonun minimum veya maksimum değerlerini belirleyen şartlar tespit edilir.

Bir  $x_0$  noktası  $f(x)$  fonksiyonunun minimum değeri ise, aynı nokta  $-f(x)$  fonksiyonunun maksimum değeridir [3]. Bu durum Şekil 1.1.'de gösterilmiştir.



Şekil 1.1. Minimum ve maksimum değer gösterimi [3]

Çok küçük pozitif ve negatif bütün  $h$  değerleri için  $f(x_0) \leq f(x_0 + h)$  ise  $f(x)$  fonksiyonu  $x_0$  'da yerel minimuma sahiptir.  $f(x_0) \geq f(x_0 + h)$  ise  $f(x)$  fonksiyonu  $x_0$  'da yerel maksimuma sahiptir [3]. Şekil 1.2.'de yerel ve global minimum ve maksimuma örnek gösterilmiştir.



Şekil 1.2. Yerel ve global minimum ve maksimum değerlerin gösterimi [3]

Hedef fonksiyon, optimize edilecek büyüklük olarak adlandırılır. Tasarım değişkenleri, optimum değeri bulmak için değerleri değiştirilen parametrelerdir. Kısıtlayıcı şartlar ise, parametrelerin değer alması üzerine konulan sınırlamalardır.

Optimizasyon problemleri, kısıtların varlığına göre kısıtlayıcı veya kısıtlayıcısız, tasarım değişkenlerinin yapısına göre statik veya dinamik, amaç fonksiyonunun ve kısıtlamalar ile ilgili fonksiyonların doğrusal olup olmamasına göre doğrusal veya doğrusal olmayan, tasarım değişkenlerinin sürekli yada ayrık olmasına göre sürekli veya ayrık, içerdikleri değişkenlerin deterministik olup olmamasına göre deterministik veya stokastik olarak sınıflandırılır. Çözüme yönelik sınıflandırma metotları ise; analitik, nümerik ve sezgisel yöntemler olarak sınıflandırılır. Bu tez çalışmasında çözüme yönelik metotlardan sezgisel yöntemler üzerinde durulmuştur.

Sezgisel yöntemler çoğunlukla optimizasyon amacıyla kullanılmıştır ve bu yöntemlerin hemen-hemen hepsinde çözüm için oluşturulan popülasyon rastgele oluşturulur. Her sezgisel yöntemin kendine özgü operatörleri ve parametreleri bulunmaktadır. Problemin çözümünde kullanılan operatörlerin hesapsal modelleri içerisinde rastgele değerler kullanarak işlem yapmak ortak bir karakteristiktir. Bu yöntemlerin hiçbirinde altın oranın etkisi bu tez çalışmasındaki gibi çalışılmamıştır.

Bu tez çalışmasında sırasıyla, Yapay Atom Algoritması, Parçacık Sürü Optimizasyon Algoritması, Ateş Böceği Algoritması, Yarasa Algoritması yöntemleri ile çalışılmıştır. Bu yöntemlerde, sezgisel başka bir yöntem olan Fidan Gelişim Algoritmasının



başlangıç popülasyonunu oluşturmadaki etkileri denenmiştir. Bu yöntemler kullanılarak iki farklı veri seti üzerinde kümeleme yapılmıştır.

Tezin uygulama amacı, altın oranın sezgisel yöntemlerdeki etkilerini görmektir. Bu sebeple uygulamada altın oranın yöntemlerdeki etkilerini anlamak için öncelikle uygulamalar altın oran olmaksızın gerçekleştirilmiş, daha sonra altın oran ile birlikte sonuçların değişimi gözlenmiştir. Bunun sonucunda altın oranın sezgisel yöntemler üzerindeki etkileri ortaya konulmuştur ve bu sonuçlar bu tez çalışmasının ileriki bölümlerinde detaylı bir şekilde ele alınmıştır. Her bölümdeki uygulamalarda, test için kullanılan veri setinin, uygulama yazılımı ile elde edilen sonuçlar tarafından ne kadar kümelenebildiği doğruluk tabloları ile adım adım verilmiştir.

Bu tez çalışmasında üzerinde durulan sezgisel yöntemler, kümeleme ve altın oran ile birlikte kullanılmıştır. Literatürde her üçünün birlikte kullanıldığı hiçbir uygulamaya rastlanmamıştır.

1. bölüm giriş bölümüdür. Optimizasyon ve sezgisel yöntemlerle ilgili kısa bir genel bilgi sunulmuştur.

2. bölümde, sezgisel yöntemler hakkında genel bilgiler sunulmuştur. Bu tez çalışmasında kullanılan dört yöntem, bölüm 2.1., 2.2., 2.3. ve 2.4.'de detaylarıyla verilmiştir.

3. bölümde, altın oran hakkında genel bilgiler sunulmuştur. Altın oranın sayısal ve geometrik anlamı bu bölümde verilmiştir. Ayrıca altın oranının evrendeki müthiş düzeni gösteren örneklerinden bu bölümde bahsedilmiştir.

4. bölümde, veri kümeleme hakkında bilgi sunulmuştur. Veri kümelemede kullanılan uzaklık ölçüsü kavramı ve uzaklık ölçüsü hesaplama yöntemleri bu bölümde verilmiştir.

5. bölümde tez çalışmasında kullanılan dört sezgisel yöntemin uygulama yazılımlarının teorik alt yapıları ve uygulama sonuçları sunulmuştur.

6. bölümde bu tez çalışmasından elde edilen sonuçların tartışması yapılmış ve bu sonuçlar ışığında elde edilen kazanımlar sunulmuştur.

## 2. SEZGİSEL YÖNTEMLER

Gerçek dünyada optimizasyon problemlerinin çoğu doğrusal olmayan yapıdadır. Bu problemlerin çözümü için birçok teknik ve yaklaşım mevcuttur. Özellikle çok parametre içeren sistemlerin optimizasyon aşamalarında, parametre sayısına ve veri tipine bağlı olarak problemlerin zorluk derecesi değişiklik gösterebilmektedir. Bu tip problemlerin klasik yöntemler ile çözümü, hem modellemede, hem de çözüm aşamasında güçlükler getirmektedir. Bu tür zorlukları aşabilmek için doğada var olan sistemleri ve olayları taklit eden sezgisel yöntemler geliştirilmiştir.

Sezgisel yöntemler, büyük boyutlu optimizasyon problemleri için, makul bir sürede optimuma yakın çözümler sunabilen yöntemlerdir. Genel amaçlı sezgisel optimizasyon algoritmaları, biyoloji tabanlı, fizik tabanlı, sürü tabanlı, sosyal tabanlı, müzik tabanlı ve kimya tabanlı olmak üzere altı farklı grupta değerlendirilmektedir[4].

Fizik tabanlı: Çok-noktalı benzetimli tavlama algoritması [5], Yerçekimi arama algoritması [6], Elektromanyetizma algoritması [7,8],Büyük patlama-Büyük çöküş algoritması [9].

Biyoloji tabanlı: Yarasa algoritması [10-12], Genetik algoritma [13,14], Karınca koloni algoritması [15], Arı koloni algoritması [16,17], Yapay bağışıklık algoritması [18], Fidan gelişim algoritması[19-24], İnvazif ot (Invasive weed) optimizasyon algoritması [25], Maymun arama algoritması [26], Bakteri yiyecek arama (Bacterial foraging) algoritması [27], Cırcır böceği algoritması [28], Evrimsel algoritma [29].

Sosyal tabanlı: Çok-noktalı tabu arama algoritması[30], Emperyalist rekabetçi algoritma [31].

Kimya tabanlı: Yapay kimyasal reaksiyon optimizasyon algoritması [32], Yapay atom algoritması [33-35].

Sürü tabanlı: Ateş böceği algoritması [36-39], Parçacık sürü optimizasyon algoritması [40-44], Kedi sürü optimizasyon algoritması [45].

Müzik tabanlı: Harmony search algoritması [39,46].

Sezgisel yöntemler bilgisayar bilimlerinde, bir problem çözme tekniğidir. Genelde problemlerin çözümünde iyiye yakın olacak şekilde çözümler elde ederler. Elde edilen sonuçların doğruluğunun kanıtlanabilir olup olmadığını önemsememektedir. Sezgisel yöntemler, en iyi sonucu bulabileceklerini garanti etmezler, fakat makul bir sürede bir çözüm elde edeceklerini garanti ederler. Genellikle en iyiye yakın olan çözüme bilinen matematiksel yada analitik çözümlerden daha hızlı ve kolay bir şekilde ulaşırlar. Bu nedenle çözümü güç ve karmaşık olan sistemlerin çözümünde tercih edilirler.

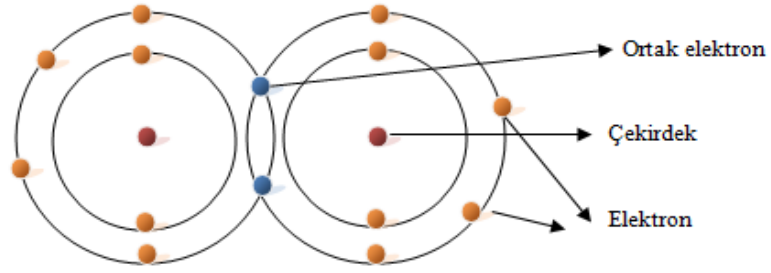
Sezgisel algoritmalar, anlaşılabilirlik yönünden çok daha basittir. Optimizasyon problemlerinin kesin çözüme giden yolları bazen çok karmaşık olabilmektedir. Bu sebeplerden dolayı çeşitli problemlerin çözümünde sezgisel algoritmalara ihtiyaç duyulmaktadır [47].

İyi bilinen sezgisel yöntemlere örnek olarak:

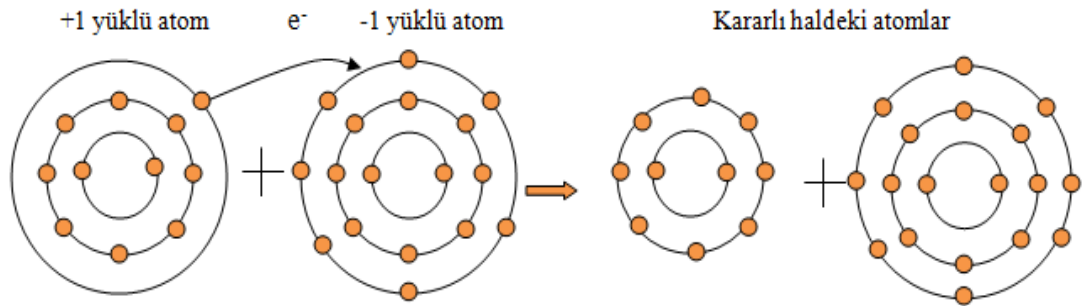
- 1- Genetik Algoritma
- 2- Yapay sinir ağları
- 3- Karınca koloni algoritması
- 4- Ateş böceği algoritması
- 5- Yarasa algoritması
- 6- Yapay atom algoritması
- 7- Fidan gelişim algoritması
- 8- Parçacık sürü optimizasyon algoritması
- 9- Destek vektör makineleri
- 10- Benzetimli tavlama
- 11- Arı koloni algoritması
- 12- Kurt koloni algoritması
- 13- Yapay balık sürüsü algoritması v.b. verilebilir.

## **2.1. Yapay Atom Algoritması ( A<sup>3</sup> : Artificial Atom Algorithm )**

Bu metot, atomlar arası çekimlerde etkili olan kovalent ve iyonik bağları esas almaktadır. Atomların yapısında çekirdek ve etrafında elektronlar vardır. Son yörüngedeki elektronların bir kısmı bağ esnasında ortaklaşa kullanılır. Kovalent bağ elektron ortaklaşmasından, iyonik bağ ise zıt yüklü çekim kuvvetlerinden oluşur. Pozitif yüklü atom elektron verecek, negatif yüklü atom ise elektron alacak şekilde bağ oluştururlar. Atomlarda her iki bağ çeşidinin de etkileri mevcuttur [33-35].



Şekil 2.1. Kovalent bağ temsili



Şekil 2.2. İyonik bağ temsili

Yapay atom algoritmasında kovalent ve iyonik bağların çözümlerini temsil eden çözüm matrisleri mevcuttur. Bu matrisler çözüm değerlerini içerisinde barındırır.

**Tablo 2.1.** Yapay atom algoritması çözüm matrisi temsili

$K_1$	$K_2$	$K_3$	$K_4$	$K_5$	...	$K_n$	$\dot{I}_1$	$\dot{I}_2$	$\dot{I}_3$	...	$\dot{I}_n$
-------	-------	-------	-------	-------	-----	-------	-------------	-------------	-------------	-----	-------------

$K_1, K_2, \dots, K_n$ 'ler kovalent bağ çözümlerini,  $\dot{I}_1, \dot{I}_2, \dots, \dot{I}_n$ 'ler ise iyonik bağ çözümlerini ifade etmektedir. Bu temsil yardımı ile bir popülasyon oluşturulur [33-35]. Kovalent bağ ve iyonik bağ çözümlerinin temsil edildiği çözümlerin özellik sayıları (n) eşit olmak zorunda değildir. Fakat bu tez çalışmasında geliştirilen yöntemde farklı matriste temsil edildiklerinden özellik sayıları eşit seçilmiştir.

**Tablo 2.2.** Yapay atom algoritması popülasyon temsili

$K_{11}$	$K_{12}$	$K_{13}$	...	$K_{1n}$	$\dot{I}_{11}$	$\dot{I}_{12}$	$\dot{I}_{13}$	...	$\dot{I}_{1n}$
$K_{21}$	$K_{22}$	$K_{23}$	...	$K_{2n}$	$\dot{I}_{21}$	$\dot{I}_{22}$	$\dot{I}_{23}$	...	$\dot{I}_{2n}$
$K_{31}$	$K_{32}$	$K_{33}$	...	$K_{3n}$	$\dot{I}_{31}$	$\dot{I}_{32}$	$\dot{I}_{33}$	...	$\dot{I}_{3n}$
$K_{41}$	$K_{42}$	$K_{43}$	...	$K_{4n}$	$\dot{I}_{41}$	$\dot{I}_{42}$	$\dot{I}_{43}$	...	$\dot{I}_{4n}$
.	.	.	...	.	.	.	.	...	.
.	.	.	...	.	.	.	.	...	.
$K_{n1}$	$K_{n2}$	$K_{n3}$	...	$K_{nn}$	$\dot{I}_{n1}$	$\dot{I}_{n2}$	$\dot{I}_{n3}$	...	$\dot{I}_{nn}$

Bu tez çalışmasında Yapay atom algoritmasında aynı matriste temsil edilen kovalent ve iyonik bağ çözümleri Tablo 2.3. ve Tablo 2.4.'de gösterildiği gibi farklı iki matriste temsil edilmiştir.

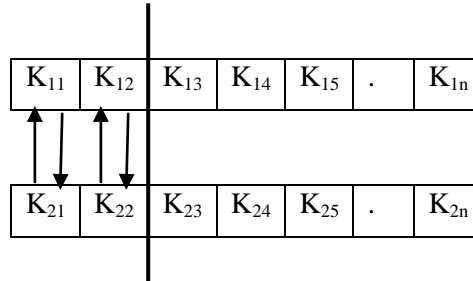
**Tablo 2.3.** Yapay atom algoritması kovalent bağ temsili

$K_{11}$	$K_{12}$	$K_{13}$	$K_{14}$	$K_{15}$	.	$K_{1n}$
$K_{21}$	$K_{22}$	$K_{23}$	$K_{24}$	$K_{25}$	.	$K_{2n}$
$K_{31}$	$K_{32}$	$K_{33}$	$K_{34}$	$K_{35}$	.	$K_{3n}$
$K_{41}$	$K_{42}$	$K_{43}$	$K_{44}$	$K_{45}$	.	$K_{4n}$
.	.	.	.	.	.	.
.	.	.	.	.	.	.
$K_{m1}$	$K_{m2}$	$K_{m3}$	$K_{m4}$	$K_{m5}$	.	$K_{mn}$

**Tablo 2.4.** Yapay atom algoritması iyonik bağ temsili

$\dot{I}_{11}$	$\dot{I}_{12}$	$\dot{I}_{13}$	$\dot{I}_{14}$	$\dot{I}_{15}$	.	$\dot{I}_{1n}$
$\dot{I}_{21}$	$\dot{I}_{22}$	$\dot{I}_{23}$	$\dot{I}_{24}$	$\dot{I}_{25}$	.	$\dot{I}_{2n}$
$\dot{I}_{31}$	$\dot{I}_{32}$	$\dot{I}_{33}$	$\dot{I}_{34}$	$\dot{I}_{35}$	.	$\dot{I}_{3n}$
$\dot{I}_{41}$	$\dot{I}_{42}$	$\dot{I}_{43}$	$\dot{I}_{44}$	$\dot{I}_{45}$	.	$\dot{I}_{4n}$
.	.	.	.	.	.	.
.	.	.	.	.	.	.
$\dot{I}_{m1}$	$\dot{I}_{m2}$	$\dot{I}_{m3}$	$\dot{I}_{m4}$	$\dot{I}_{m5}$	.	$\dot{I}_{mn}$

Popülasyonun kovalent bölgedeki tüm çözüm matrislerinin uygunluk ( fitness ) değerleri hesaplanıp sıralanır. Kovalent bölgede uygunluğu en zayıf elemanlar içerisinde seçilen özellik (attribute) sayısı kadar özellik uygunluğu yüksek olan bireyler ile değer değiştirilir. Böylelikle karşılıklı eleman değişikliği ile çeşitlilik sağlanır.



**Şekil 2.3.** Yapay atom algoritması elektron ortaklığının temsili

Böylelikle çözüm çeşitliliği de sağlanabilmektedir. Daha sonrasında iyonik bağları temsilen oluşturulan çözümlerde işin içine katılarak tüm popülasyon tekrar bir uygunluk hesaplamasından geçirilir. Sıralama işlemi tekrar gerçekleştirilir. İyonik bağlar için oluşturulan rastgele çözümler, çözüm uzayına eklenerek, çözüm uzayındaki çeşitlilik artırılır. Yeni bireylerle beraber oluşan popülasyon sıralandıktan sonra işlemler belirli bir durdurma kriteri sağlanan kadar uygulanır.

Yapay atom algoritmasında kullanılan yöntemin bir zahiri kodu aşağıda verilmiştir.

```
Başla
Kovalent ve iyonik çözüm sayılarını altın orana denk gelmeyecek şekilde belirle
Kovalent çözümleri üret ( $K_{in}$ )
İyonik çözümleri üret ( $I_{in}$ )
While sayaç<adım sayısı
    Kovalent ve iyonik çözümleri tek popülasyonda birleştir
    Her çözüm adayının uygunluk değerini hesapla
    Eğer uygunluk( $pop_i$ ) > uygunluk( $pop_{i+1}$ ) ise
        Kaç özelliğın ortaklaşılacağını altın oran olmaksızın belirle
         $Pop_i$ 'den  $pop_{i+1}$ 'e değer kopyala
    End eğer
    Eğer uygunluk( $pop_i$ ) < uygunluk( $pop_{i+1}$ ) ise
        Kaç özelliğın ortaklaşılacağını altın oran olmaksızın belirle
         $Pop_{i+1}$ 'den  $pop_i$ 'ye değer kopyala
    End eğer
    Uygunluk değerlerine göre çözümleri sırala
    İyi olanları kovalent çözüm kümesinde tut
    İyonik çözümleri rastgele tekrar üret
End while
Son işlemleri yap ve çözümü sun
End
```

Aynı zahiri kod altın oran ile birleştirildiğinde sıradaki zahiri kodda olduğu gibi kullanılmıştır.

```
Başla
Kovalent ve iyonik çözüm sayılarını altın orana denk gelecek şekilde belirle
Kovalent çözümleri üret ( $K_{in}$ )
İyonik çözümleri üret ( $I_{in}$ )
While sayaç<adım sayısı
  Kovalent ve iyonik çözümleri tek popülasyonda birleştir
  Her çözüm adayının uygunluk değerini hesapla
  Eğer uygunluk( $pop_i$ ) > uygunluk( $pop_{i+1}$ ) ise
    Yüzdesi altın oran değerine denk gelecek kadar özellik ortaklaş
     $Pop_i$ 'den  $pop_{i+1}$ 'e değer kopyala
  End eğer
  Eğer uygunluk( $pop_i$ ) < uygunluk( $pop_{i+1}$ ) ise
    Yüzdesi altın oran değerine denk gelecek kadar özellik ortaklaş
     $Pop_{i+1}$ 'den  $pop_i$ 'ye değer kopyala
  End eğer
Uygunluk değerlerine göre çözümleri sırala
İyi olanları kovalent çözüm kümesinde tut
İyonik çözümleri rastgele tekrar üret
End while
Son işlemleri yap ve çözümü sun
End
```

## 2.2. Parçacık Sürü Optimizasyonu Algoritması

### 2.2.1. Parçacık sürü optimizasyonu

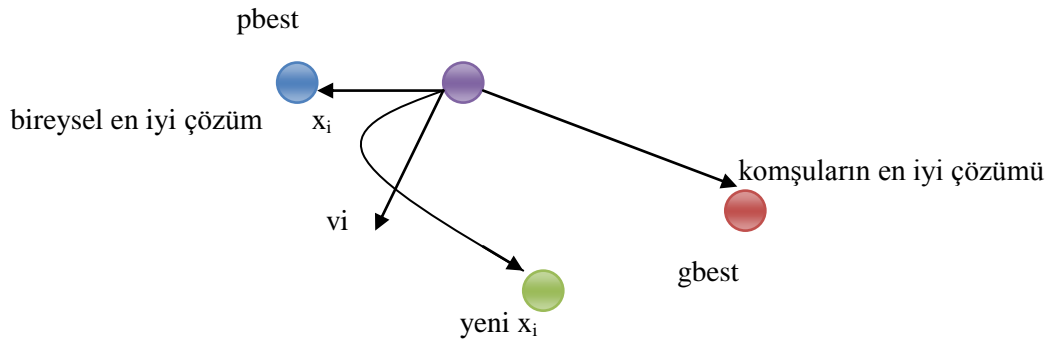
1995 yılında Russell Eberhart ve James Kennedy tarafından bulunmuştur. Popülasyon temelli, evrimsel bir optimizasyon algoritmasıdır [40]. Kuş ve balık sürülerinin yiyecek bulma ve tehlikelerden kaçmaları anındaki birlikte yaptıkları hareketlerden esinlenerek ortaya konulmuştur. Sonuca çabuk ulaşması, fazla parametre olmaması ve yerel optimumlara takılmaması sebebiyle diğer birçok optimizasyon algoritmasına göre üstündür [42].

Popülasyondaki bütün bireylerin optimum kaynağın yerini bilmediği durumda bile, popülasyondaki tüm bireylerinin başarılı bir şekilde kaynağa ulaşabilmesinden esinlenilerek geliştirilmiş olan bu yöntem;

- 1) Her bir bireyin kendi tecrübelerinde bulunan iyi konuma gitme eğilimi olarak düşünülen bilişsel (cognitive) davranış biçimi (pbest)
- 2) Her bir bireyin iyi konumlarda bulunan diğer bireyleri takip etme eğilimi olarak tanımlanabilecek sosyal (social) davranış biçimi (gbest)

3) Her bir bireyin rastgele olarak arama yapma eğilimi olarak tanımlanabilecek keşifsel (exploratory) davranış biçimleri arasında bir denge olduğu varsayımına dayanmaktadır [43].

Bu yöntemde parçacıklar, sürü içinde optimum çözüme en yakın olan parçacığa doğru yönelmektedir. Bu amaçla parçacıkların konum bilgilerinin popülasyonda paylaşılması gerekmektedir. Her bir parçacık kendi pozisyonunu, önceki tecrübesinden faydalanarak sürüdeki en iyi pozisyon değerine göre doğru ayarlar. Temelde popülasyondaki bireylerin pozisyonlarının, popülasyonun en iyi bireyine yaklaştırılmasına dayanır [42].



Şekil 2.4. Parçacık yaklaşımı şekilsel temsili

### 2.2.2. Parçacık sürü optimizasyon algoritması

Algoritma temel olarak aşağıdaki basamaklardan oluşur [44].

1. Rastgele üretilen başlangıç pozisyonları, hızları ve başlangıç popülasyonu oluşturulur.
2. Popülasyon içerisindeki bütün parçacıkların uygunluk değerleri hesaplanır.
3. Her bir parçacık için mevcut jenerasyondan yerel en iyi "pbest" değeri bulunur.
4. Mevcut jenerasyondaki yerel eniyiler içerisinde küresel en iyi "gbest" seçilir.
5. Pozisyon ve hızlar güncellenerek sonlanma adımına kadar işlem devam ettirilir.



### 2.2.3. Parçacık sürü optimizasyonu çözüm yöntemi

D adet parametreden oluşan n adet parçacık olduğunu varsayalım. Bu durumda popülasyon parçacık matrisi aşağıdaki gibidir.

$$X = \begin{bmatrix} x_{11} & x_{12} & \dots & \dots & \dots & x_{1d} \\ x_{21} & x_{22} & \dots & \dots & \dots & x_{2d} \\ x_{31} & x_{32} & \dots & \dots & \dots & x_{3d} \\ \vdots & \vdots & & & & \vdots \\ \vdots & \vdots & & & & \vdots \\ x_{n1} & x_{n2} & \dots & \dots & \dots & x_{nd} \end{bmatrix} \quad n \times D$$

Yukarıdaki matriste, i. parçacık  $x_i = [x_{i1}, x_{i2}, \dots, x_{id}]$  olarak ifade edilir. Yerel en iyi uygunluk değerini veren i. parçacığın değeri  $pbest_i = [p_{i1}, p_{i2}, \dots, p_{id}]$  olarak ifade edilir. “gbest” ise her iterasyonda bütün parçacıklar için bir tanedir ve  $gbest = [p_1, p_2, \dots, p_d]$  şeklinde gösterilir. i. parçacığın hızı yada diğer bir ifadeyle her boyuttaki değerinin değişim miktarı  $v_i = [v_{i1}, v_{i2}, \dots, v_{id}]$  olarak ifade edilir. İki en iyi değer bulunmasından sonra parçacık hızları ve konumları aşağıda denklem (2.1) ve denklem (2.2) ‘ye göre güncellenir [40,43].

$$v_i^{k+1} = v_i^k + c_1 \cdot rand_1^k (pbest_i^k - x_i^k) + c_2 \cdot rand_2^k (gbest^k - x_i^k) \quad (2.1)$$

$$\text{eğer } v_i^{k+1} > v_{\max}, v_i^{k+1} = v_{\max} \quad (2.2)$$

$$\text{eğer } v_i^{k+1} < v_{\min}, v_i^{k+1} = v_{\min} \quad (2.3)$$

$$x_i^{k+1} = x_i^k + v_i^{k+1} \quad (2.4)$$

Denklem (2.1)’deki , c1 ve c2 öğrenme faktörleridir. c1 ve c2, her çözümü pbest ve gbest çözümlerine doğru çeken, hızlanma sabitlerdir. c1, çözümün o jenerasyondaki çözümlere göre hareket etmesini, c2 ise o ana kadarki diğer çözümlerin tecrübelerine göre hareket etmesini sağlar. Düşük değerler çözümün hedef bölgeye doğru hareketinden önce, bu bölgeden uzak yerlerde dolaşmalarına imkân verir. Ancak çözüme kavuşma süresi uzayabilir. Yüksek değerler seçilmesi ise, çözüme kavuşmayı çabuklaştırırken, beklenmedik hareketlerin ortaya çıkmasına ve hedef bölgenin atlanmasına sebep olabilir. Bu algoritma üzerinde araştırmacıların yaptığı denemelerde  $c_1=c_2= 2$  olarak almanın iyi sonuçlar verdiği belirtilmiştir. Bu tez çalışmasında yapılmış olan uygulamada, bu göz önüne alınmıştır. c1 ve c2 değerlerinin 2 değerinin etrafındaki ve altın oran etrafındaki sonuçları verilmiştir. Parçacık sürü optimizasyon algoritmasında kullanılan yöntemin bir zahiri kodu aşağıda verilmiştir.

```

Başla
Vmax değerleri belirle
Parçacık sürü optimizasyonu popülasyonunu oluştur  $X_i$  ( $i=1,2,\dots,n$ )
Hız değerlerini belirle
Pbest ve gbest değerleri popülasyonun ilk çözümlerine ayarla
While sayaç<adım sayısı
Pbest ve gbest çözümlerin uygunluk değerlerini hesapla
Uygunluk değerlerini sırala
Yeni pbest ve gbest değerleri hesapla
C1, C2,  $rand_1$  ve  $rand_2$  değerlerini altın orana bağlı kalmaksızın seç
 $v_i^{k+1} = v_i^k + c_1 \cdot rand_1^k (pbest_i^k - x_i^k) + c_2 \cdot rand_2^k (gbest^k - x_i^k)$  formülüne göre
eğer  $v_i^{k+1} > v_{max}$  ise  $v_i^{k+1} = v_{max}$ 
end eğer
eğer  $v_i^{k+1} < v_{min}$  ise  $v_i^{k+1} = v_{min}$ 
end eğer
 $x_i^{k+1} = x_i^k + v_i^{k+1}$  değerini hesapla
End while
end

```

Aynı zahiri kod altın oran ile birleştirildiğinde sıradaki zahiri kodda olduğu gibi kullanılmıştır.

```

Başla
Vmax değerleri belirle
Parçacık sürü optimizasyonu popülasyonunu oluştur  $X_i$  ( $i=1,2,\dots,n$ )
Hız değerlerini belirle
Pbest ve gbest değerleri popülasyonun ilk çözümlerine ayarla
While sayaç<adım sayısı
Pbest ve gbest çözümlerin uygunluk değerlerini hesapla
Uygunluk değerlerini sırala
Yeni pbest ve gbest değerleri hesapla
C1, C2,  $rand_1$  ve  $rand_2$  değerlerini altın oran ve altın oran etrafında seç
 $v_i^{k+1} = v_i^k + c_1 \cdot rand_1^k (pbest_i^k - x_i^k) + c_2 \cdot rand_2^k (gbest^k - x_i^k)$  formülüne göre
eğer  $v_i^{k+1} > v_{max}$  ise  $v_i^{k+1} = v_{max}$ 
end eğer
eğer  $v_i^{k+1} < v_{min}$  ise  $v_i^{k+1} = v_{min}$ 
end eğer
 $x_i^{k+1} = x_i^k + v_i^{k+1}$  değerini hesapla
End while
end

```

## 2.3. Ateş Böceği Algoritması

### 2.3.1. Ateş böcekleri

Ateş böcekleri çoğunlukla tropikal ve ılıman olan bölgelerde yaşarlar. Ateş böceklerinin parlayan ışıkları bu bölgelerde sıklıkla görülür. Çoğu, kısa ve ritmik olacak şekilde ışık yayarlar. Bu ışık biyoluminesans denilen bir işlem ile üretilir. Biyoluminesans, canlı bir organizmada kimyasal bir reaksiyon sırasında kimyasal enerjinin ışık enerjisine dönüşmesi ve bu yolla ışık üretilmesi ve ışık yayılmasıdır. Ateş böceklerinde ışık şiddeti ve çekicilik oranı iki temel özelliktir. Işığın parlaklığı ve yayılma sıklığı diğer cinsleri tarafından çekilmesini artırır [36,37].

### 2.3.2. Yapay ateş böcekleri

Gerçek ateş böceklerinin davranışlarını taklit eden bu algoritma Xin-She Yang tarafından Firefly algorithm olarak sezgisel algoritmalara kazandırılmıştır. Ateş böcekleri cinsiyet ayrımı yapmadan, yaydıkları ışığın parlaklığına göre birbirlerini çekmektedirler. Bu çekim iki temel fonksiyona dayanmaktadır.

- 1- Işık şiddeti
- 2- Çekicilik

### 2.3.3. Işık şiddeti

Her ateş böceğinin yaymış olduğu ışığın bir şiddeti vardır. Birinin ötekini çekmesi bu ışığın şiddetine göre değişmektedir. Her ateş böceği ışık şiddeti fazla olan öteki ateş böceğine doğru hareket etmektedir. Aşağıda ışık şiddetinin formülü verilmiştir [36-38].

$$I = I_0 * e^{-\gamma r} \quad (2.5)$$

$I_0$  , orijinal ışık şiddeti olarak isimlendirilmekte olup ,tasarımcı tarafından belirlenmektedir.

$\gamma$ , ışık emme katsayısı olup, bu parametrede tasarımcı tarafından belirlenmektedir.

$r$ , kartezyen uzaklık olup, iki ateş böceği arasındaki mesafeyi temsil etmektedir. Aşağıdaki formül ile ifade edilir [36,37].

$$r_{ij} = \| x_i - x_j \| = \sqrt{\sum_{k=1}^d (x_{i,k} - x_{j,k})^2} \quad (2.6)$$

#### 2.3.4. Çekicilik

Işık şiddetine bağlı olarak değerlendirilecek olan çekicilik, bir ateş böceği tarafından, öteki ateş böceğinin kendine yaklaşması için kullanılan bir özellik olup aşağıdaki formül ile ifade edilmektedir [36,37].

$$\beta = \beta_0 * e^{-\gamma r^2} \quad (2.7)$$

$\beta_0$ ,  $r = 0$  için belirlenmiş olan ilk çekicilik olup, tasarımcı tarafından belirlenmektedir.  $\gamma$ , yine ışık emme katsayısı olup, bu parametrede tasarımcı tarafından belirlenmektedir.  $r$ , yine kartezyen uzaklık olup, iki ateş böceği arasındaki mesafeyi temsil etmektedir.

Her iki fonksiyonu da hesaba katarak yeni çözüm değerleri aşağıdaki gibi hesaplanmaktadır [37,38].

$$x_i = x_i + \beta_0 * e^{-\gamma r_{ij}^2} (x_j - x_i) + \alpha * \epsilon_i \quad (2.8)$$

$\alpha * \epsilon_i$ , gaussian dağılım vektör değerini temsil etmekte olup çözümün çeşitliliğine sağlamak üzere düşünülmüştür.  $\alpha$  değeri tasarımcı tarafından belirlenmektedir.

Bu tez çalışmasında, uygulama yazılımında aşağıdaki şekilde seçilmiştir.

$$x_i = x_i + \beta_0 * e^{-\gamma r_{ij}^2} (x_j - x_i) + \alpha \quad (2.9)$$

Sıradaki zahiri kod ateş böceği algoritmasının çalışma mantığını vermektedir [36,37].

```
başla
amaç fonksiyon  $f(x)$  ,  $X=(X_1,X_2,\dots,X_d)^T$ 
Ateş böceği başlangıç popülasyonunu üret  $X_i$  ( $i=1,2,\dots,n$ )
 $f(X_i)$  ile belirlenen  $I_i$  ışık şiddetlerini altın orana bağlı kalmadan belirle
emme katsayısı  $\gamma$  değerini altın orana bağlı kalmadan belirle
(sayaç<adım sayısı)
for i=1: ateşböceği sayısı
for j=1: ateş böceği sayısı
eğer ( $I_i > I_j$ )
ışık şiddeti yüksek olanı düşük olana yaklaştır
end if

r uzaklık değerine göre çekiciliği hesaplarken beta değerinde
altın orana bağlı kalma

yeni çözümleri oluştururken alpha değerinde de altın orana bağlı kalma
ışık şiddetlerini güncelle
end for i
end for j
çözümleri sırala ve en iyiyi bul
end while
son işlemleri yap çözümü sun
end
```

Aynı zahiri kod altın oran ile birleştirildiğinde sıradaki zahiri kodda olduğu gibi kullanılmıştır.

```
başla
amaç fonksiyon  $f(x)$  ,  $X=(X_1,X_2,\dots,X_d)^T$ 
Ateş böceği başlangıç popülasyonunu üret  $X_i$  ( $i=1,2,\dots,n$ )
 $f(X_i)$  ile belirlenen  $I_i$  ışık şiddetlerini altın oran değerinde belirle
emme katsayısı  $\gamma$  değerini altın oran değerinde belirle
(sayaç<adım sayısı)
for i=1: ateşböceği sayısı
for j=1: ateş böceği sayısı
eğer ( $I_i > I_j$ )
ışık şiddeti yüksek olanı düşük olana yaklaştır
end if

r uzaklık değerine göre çekiciliği hesaplarken beta değerinde
altın oran değerini kullan

yeni çözümleri oluştururken alpha değerinde de altın oran değerini kullan
ışık şiddetlerini güncelle
end for i
end for j
çözümleri sırala ve en iyiyi bul
end while
son işlemleri yap çözümü sun
end
```

Algoritmanın çalışma mantığı şu şekildedir: Başlangıç koşulları olan parametreler, ışık emme kat sayısı ve çözüm popülasyonu oluşturulduktan sonra ışık her ateş böceği için ışık şiddetleri belirlenir. Işık şiddetleri kıyaslanır ve ışık şiddeti az olan ateş böceğinin çekiciliğinin çözüm değerleri, fazla olan ateş böceğinin çekiciliğinin çözüm değerlerine doğru yaklaştırılır.

Esasında ışık şiddeti, çekicilik değerlerini değiştirmede bir ölçüdür. Esas hesaplar çekicilik değerleri üzerinden yapılır. Sonra bu çekicilik değerleri üzerinden yeni çözüm değerleri hesaplanır ve yeni değerler popülasyona yollanır. En uygun olan değer hesabı da, sıralama işleminden sonra belli olur. En iyi çözüm değeri seçilmiş olur.

## **2.4. Yarasa Algoritması**

### **2.4.1. Yarasalar**

Yarasalar son derece etkileyici hayvanlardır. Yeryüzünde yaşayan yaklaşık 4500 memeli türü vardır. Bunların yaklaşık 1000 tanesini yarasalar oluşturmaktadır. Çıkardıkları çok yüksek frekanslı ses dalgalarının, etraflarında bulunan cisimlere çarpıp geri dönmesi sonucu karanlıkta olsalar bile yönlerini bulurlar, engellerden kaçarlara, avlanırlar ve hareket ederler. Bu sonar sistemlere benzeyen özelliklerine, ekolokasyon denir [10-12].

Bu sesler birçok zaman insanlar tarafından duyulamazlar. Yarasaların gözleri görür ama buna rağmen, bu radar sistemine benzer özellikleri sayesinde yönlerini tayin eder, avlanır ve hareket ederler. Yapılan deneyler, kör edilen yarasaların hiçbir engele çarpmadan hareket edebildiğini ve avlanabildiğini göstermiştir. Yarasalar bu sesleri burunları ve ağızları ile çıkarırlar. Tıpkı bir çanağa benzeyen kulakları ve hissi organlarıyla algılar. Yarasalar avlanırken uçuş esnasında, saniyede yaklaşık 200 çığlık çıkarırlar [48].

Yarasalar çıkarmış oldukları bu seslerin cisimlere çarpıp geri dönmesinde dalga boyu ve hız arasındaki ilişkiyi kullanırlar [10-12].

$$\lambda = \frac{v}{f} \quad (2.10)$$

### 2.4.2. Yarasa algoritmasının adımları

Yarasa algoritmasının 3 temel kuralı aşağıdaki gibidir [10-12].

- 1- Tüm yarasalar ekolokasyonu kullanırlar ve bu sayede avları veya yiyecekleri ile aralarındaki mesafeyi hissederler.
- 2- Her yarasanın bir hızı ( $v_i$ ), bir pozisyonu ( $x_i$ ), bir frekans değeri ( $f_i$ ), bir dalga boyu ( $\lambda$ ), gürültü değeri ( $A_0$ ) ve bir sinyal yayma oranı ( $r$ ) olacaktır.
- 3- Gürültü bir  $A_0$  değerinden avı veya yiyeceği yakalayınca minimum olacak olan bir  $A_{min}$  değerine kadar değişir.

### 2.4.3. Yapay yarasaların hareketleri

Yapay yarasaların hareketlerini tanımlamada 3 temel formül vardır.

$$f_i = f_{min} + (f_{max} - f_{min}) * \beta \quad (2.11)$$

$$v_i^{t+1} = v_i^t (x_i^t - x^*) * f_i \quad (2.12)$$

$$x_i^{t+1} = x_i^t + v_i^t \quad (2.13)$$

Burada  $f_i$  değerleri, her yarasa için tanımlanan frekans değerleridir. Her yarasanın çıkarmış olduğu seslerin çeşitli frekans aralıkları vardır. Bu tasarımcının belirleyeceği aralıkta değerlerdir. Bu tez çalışmasında frekans değerleri [0,1] aralığında seçilmiştir.

Burada  $\beta$ , [0,1] aralığında rastgele bir değerdir. Bu tez çalışmasında altın oran değerinin  $\beta$  üzerindeki etkileri de denenmiş ve olumlu sonuçlar alınmıştır.

$x^*$  ise tüm yarasa çözüm adayları içerisindeki global en iyi çözümdür. Algoritmanın adımlarında başka en iyiler çıktıkça bu değer güncellenmektedir.  $x$  pozisyon değeri bir işlem adımından daha geçerek yeniden hesaplanır.

$$x_{new} = x_{old} + \epsilon * A^t \quad (2.14)$$

Yeni oluşturulan  $x_{new}$  değeri yarasanın yeni pozisyon değeri olacaktır. Burada  $\epsilon$  değeri ise [-1,1] aralığında rastgele bir sayı olacaktır [10-12].

#### 2.4.4. Gürültü ve sinyal yayma oranı

Gürültü, yarasanın avını veya yiyeceği yakalayınca kadar çıkarmış olduğu sestir. Bu ses yardımıyla avının veya yiyeceğinin yerini tespit eder. Ekolokasyon sayesinde av veya yiyeceğe çarpıp geri dönen sinyaller yardımıyla cisim hareket etse bile yarasa cismini bulacaktır. Gürültü av veya yiyecek yakalanınca minimum olacak bir değerdir ve aşağıda formül ile ifade edilir [10-12].

$$A_i^{t+1} = \alpha * A_i^t, r_i^{t+1} = r_i^0 [1 - \exp(-\gamma t)] \quad (2.15)$$

Burada  $\alpha$ , [0,1] aralığında rastgele bir sayıdır.  $\gamma$  ise tasarımcının belirleyeceği sıfırdan büyük bir değerdir. Ava veya yiyeceğe yaklaştıkça sinyal yayma oranı arttıkça gürültü azalacaktır. Gürültü av veya yiyeceğe ulaşıncaya sonlanacaktır [10-12]. Bu tez çalışmasında sinyal yayma oranı ve gürültü değeri için, hem başlangıç koşulları hem de artma azalma oranları altın oran olmadan ve altın orana bağlı olacak şekilde denemiştir. Altın oran ile yapılan denemelerde olumlu sonuçlar alınmıştır.

Aşağıda Yarasa algoritmasının bir zahiri kodu verilmiştir [10-12].

```
Uygunluk fonksiyonu f(x), x = (x1, ..., xd)T
Yarasa çözüm popülasyonunu oluştur xi (i = 1, 2, ..., n)
vi hız değerlerini oluştur
xi'lerin fi frekans değerlerini belirlerken beta değerini altın orana bağlı kalmadan seç
Sinyal yayma oranı r ve gürültü oranı A değerlerini altın orana bağlı kalmadan seç
while (sayaç < adım sayısı)
    Frekans değeri etrafında yeni çözümler oluştur, xi ve vi değerlerini güncelle
    eğer (rastgele üretilen bir sayı > ri) ise
        BEST çözüm etrafında altın orana bağlı kalmadan bir yeni çözüm üret
    end eğer
    Rastgele yeni bir çözüm üret
    eğer (rand < Ai & f(xi) < f(x*))
        Yeni üretilen çözümü kabul et
        ri değerini altın orana bağlı kalmadan artır
        Ai değerini altın orana bağlı kalmadan azalt
    end eğer
    Bulunan çözümleri uygunluk fonksiyonuna göre sırala
end while
son işlemleri yap ve çözümü sun
end
```



Aynı zahiri kod altın oran ile birleştirildiğinde sıradaki zahiri kodda olduğu gibi kullanılmıştır.

```
Uygunluk fonksiyonu  $f(x)$ ,  $x = (x_1, \dots, x_d)^T$   
Yarasa çözüm popülasyonunu oluştur  $x_i$  ( $i = 1, 2, \dots, n$ )  
 $v_i$  hız değerlerini oluştur  
 $x_i$ 'lerin  $f_i$  frekans değerlerini belirlerken beta değerini altın oran değerinde seç  
Sinyal yayma oranı  $r$  ve gürültü oranı  $A$  değerlerini altın oran değerinde seç  
while (sayaç < adım sayısı)  
    Frekans değeri etrafında yeni çözümler oluştur,  $x_i$  ve  $v_i$  değerlerini güncelle  
    eğer (rastgele üretilen bir sayı  $> r_i$ ) ise  
        BEST çözüm etrafında altın oran değerinde bir katsayı ile yeni çözüm üret  
    end eğer  
    Rastgele yeni bir çözüm üret  
    eğer ( $\text{rand} < A_i$  &  $f(x_i) < f(x_*)$ )  
        Yeni üretilen çözümü kabul et  
         $r_i$  değerini altın oran değerinde bir katsayı ile çarparak artır  
         $A_i$  değerini altın oran değerinde bir katsayı ile çarparak azalt  
    end eğer  
    Bulunan çözümleri uygunluk fonksiyonuna göre sırala  
end while  
son işlemleri yap ve çözümü sun  
  
end
```

Burada  $f$  frekans değeri  $0, f_{\max}$  aralığında,  $r$  değeri ise  $[0,1]$  aralığında değişir.

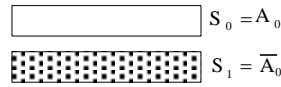
Yukarıda verilen zahiri kodda geçen BEST çözüm etrafında bir yeni çözüm üret adımı en iyi çözüm etrafında üretilen yeni bir çözüm olacaktır. Bu tez çalışmasında bu adım hem rastgele olacak şekilde hem de altın orana bağlı olacak şekilde denemiştir. Altın orana bağlı olarak yapılan adımlarda elde edilen sonuçlardan olumlu sonuçlar elde edilmiştir.

## 2.5. Fidan Gelişim Algoritması

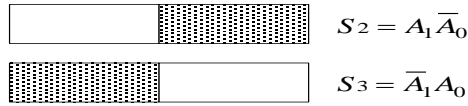
Fidan gelişim algoritması fidanların gelişiminden esinlenerek geliştirilmiş bir algoritmadır. Fidanların gelişim süreçlerini en iyi bir şekilde taklit eder [19-24]. Bu yöntemde problemin çözümünü teşkil edebilecek olan değerler fidanlara yerleştirilip daha sonra yapılacak olan işlemler bu fidanlar üzerine uygulanmaktadır.

Geliştirilmiş olan yöntemdeki operatörler, fidan gelişiminin mümkün olduğunca en iyi bir şekilde taklit edilmesi amacıyla fidanların gelişim esnasında nasıl bir değişikliğe uğradıkları operatörler olarak ele alınmıştır. Bundan dolayı, fidan ekimi bir operatör, eşleştirme bir operatör, dallanma bir operatör ve aşılama da bir operatör olarak ele alınmıştır.

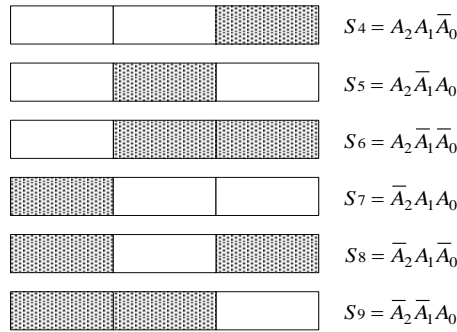
Fidan ekimi operatörü, arama uzayına düzgün dağılmış fidanların (çözümlerin) oluşturulmasını sağlamaktadır. Zaten Fidan gelişim algoritmasının diğer optimizasyon problemlerinden üstün olan tarafı düzenli popülasyon oluşturmaktır. Amaç mümkün olabilecek çözüm aralığını oluştururken, en küçük ve en büyük değerleri ve bunlardan oluşan bir popülasyon oluşturmaktır. Bu sayede çözüm aralığını mümkün olduğunca en iyi şekilde dağıtmaktır. Bu sağlanınca en iyi çözümlere yakınsamak daha kısa sürede ve daha iyi bir başarı ile olabilecektir. Diğer algoritmalarda bazen yerel minimum veya maksimumlara takılabilmeler olmaktadır. Fidan gelişim algoritmasının başlangıç popülasyonunu oluşturmada kullandığı bu yöntem bu problemleri ortadan kaldırabilmektedir.



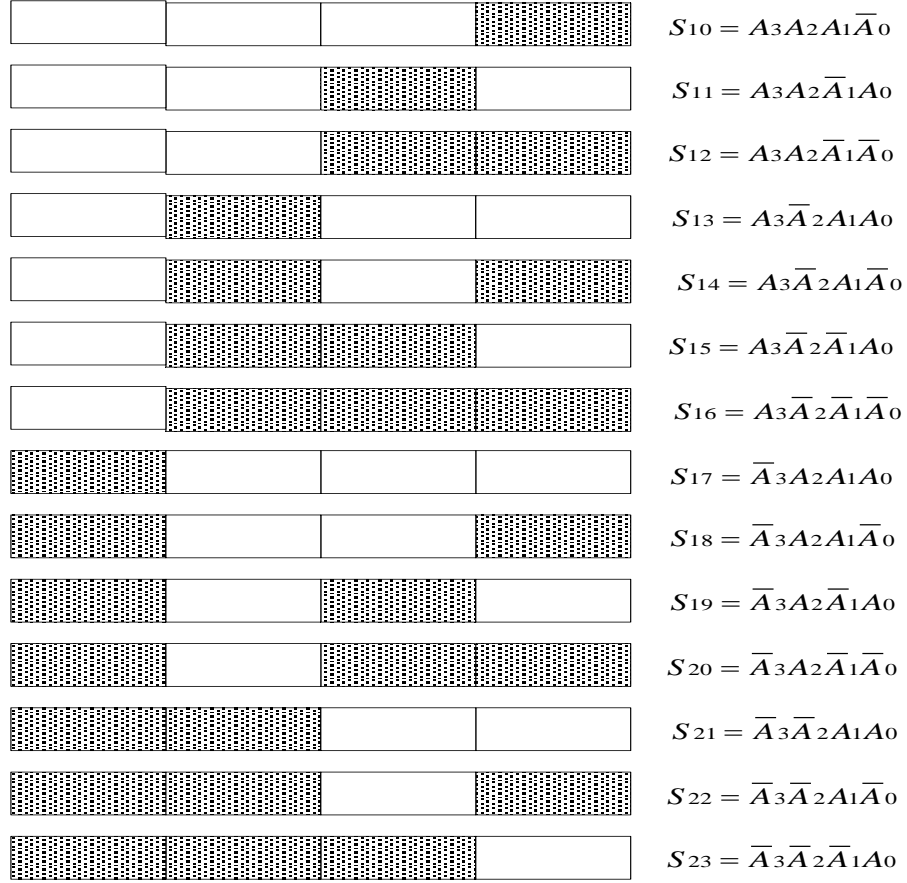
**Şekil 2.5.** Fidan gelişim algoritması başlangıç fidanları



**Şekil 2.6.** Fidan gelişim algoritması başlangıç popülasyonu k=2 için yeni fidanlar



**Şekil 2.7.** Fidan gelişim algoritması başlangıç popülasyonu k=3 için yeni fidanlar



Şekil 2.8. Fidan gelişim algoritması başlangıç popülasyonu  $k=4$  için yeni fidanlar

Tablo 2.5. Fidan gelişim algoritması başlangıç popülasyonu oluşturma hesapsal temsili

Minimum1	Minimum2	Minimum3	Minimum4
$L_1 + (u_1 - l_1) * r$	$L_1 + (u_1 - l_1) * r$	$L_1 + (u_1 - l_1) * r$	$L_1 + (u_1 - l_1) * r$
$L_1 + (u_1 - l_1) * r$	$L_1 + (u_1 - l_1) * r$	$L_1 + (u_1 - l_1) * r$	$L_1 + (u_1 - l_1) * (1 - r)$
$L_1 + (u_1 - l_1) * r$	$L_1 + (u_1 - l_1) * r$	$L_1 + (u_1 - l_1) * (1 - r)$	$L_1 + (u_1 - l_1) * r$
$L_1 + (u_1 - l_1) * r$	$L_1 + (u_1 - l_1) * r$	$L_1 + (u_1 - l_1) * (1 - r)$	$L_1 + (u_1 - l_1) * (1 - r)$
$L_1 + (u_1 - l_1) * r$	$L_1 + (u_1 - l_1) * (1 - r)$	$L_1 + (u_1 - l_1) * r$	$L_1 + (u_1 - l_1) * r$
$L_1 + (u_1 - l_1) * r$	$L_1 + (u_1 - l_1) * (1 - r)$	$L_1 + (u_1 - l_1) * r$	$L_1 + (u_1 - l_1) * (1 - r)$
$L_1 + (u_1 - l_1) * r$	$L_1 + (u_1 - l_1) * (1 - r)$	$L_1 + (u_1 - l_1) * (1 - r)$	$L_1 + (u_1 - l_1) * r$
$L_1 + (u_1 - l_1) * r$	$L_1 + (u_1 - l_1) * (1 - r)$	$L_1 + (u_1 - l_1) * (1 - r)$	$L_1 + (u_1 - l_1) * (1 - r)$
...	...	...	...
...	...	...	...
Maksimum1	Maksimum2	Maksimum3	Maksimum4

Burada L deęerleri özüm kümesi ierisindeki her zellik iin minimum deęerleri, U deęerleri ise özüm kümesi ierisindeki her zellik iin maksimum deęerleri temsil etmektedir [19-22,24].

Bu tez alıřmasında fidan geliřim algoritmasının bařlangı popülasyonu oluřturmadaki teknięinden faydalanılmıřtır. Bu teknik öteki sezgisel algoritmalarda bařlangı popülasyonunu oluřturmada kullanılmıřtır. Sonuları rastgele oluřturulan bařlangı popülasyonlarına göre, özüme ulařmada daha iyi sonular vermiřtir.

### 3. ALTIN ORAN

Altın oran, canlı ve cansızların birçoğunun şeklinde ve yapısında bulunan özel bir orandır. Bir bütünün parçaları arasında keşfedilebilen, sanat ve mimaride de uygulanmış, estetik ve uyum açısından en düzgün denilebilecek boyutları verdiği düşünülen geometrik ve sayısal bir orandır. En belirgin örneklerine insan vücudunda, yapraklarda, ağaç dallarında, deniz kabuklularında rastlanır. Platon'a göre kozmik fiziğin anahtarı bu orandır. Eski Mısırlılar ve Yunanlılar tarafından keşfedilmiş, mimaride ve sanatta kullanılmıştır.

Bir doğru parçasının Altın Oran'a uygun biçimde iki parçaya bölünmesi gerektiğinde, bu doğru öyle bir noktadan bölünmelidir ki; küçük parçanın büyük parçaya oranı, büyük parçanın bütün doğruya oranına eşit olmalıdır. [49].

$$\frac{|\langle \text{-----} 1 \text{-----} \rangle|}{|\langle \text{-----} X \text{-----} \rangle| \langle \text{-----} 1 - X \text{-----} \rangle|}$$

Şekil 3.1. Doğru parçasında altın oran [49]

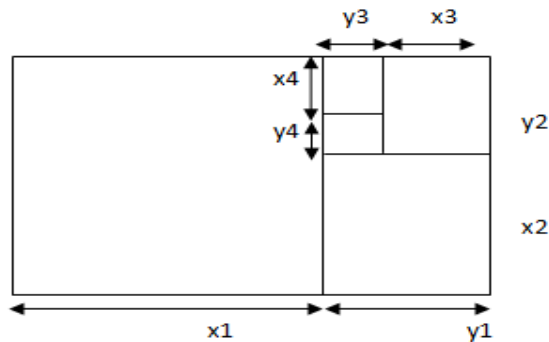
Şekil 3.1.'de uzunluğu 1 olan bir doğru parçası iki parçaya bölünmüştür. Buna göre

$$(1/X) = X/(1-X) \text{ veya } X^2 + X - 1 = 0 \text{ olduğu varsayılır ise,} \quad (3.16)$$

Denklem (3.16)'nın pozitif kökü  $X = (-1 + \sqrt{5})/2 \approx 0.61803398875$ .

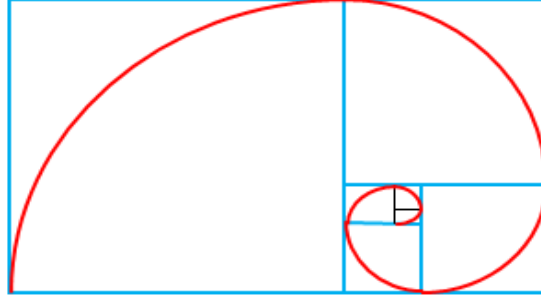
$1/X = (1 + \sqrt{5})/2 \approx 1.61803398875$  olacaktır. [49].

Bir altın dikdörtgenin, "subrectangles" olarak adlandırılan kendisinden daha küçük dikdörtgenlere ayrılabilir olduğu bilinmektedir. Altın dikdörtgen, altın orana bağlı kalınarak kendisinden daha küçük dikdörtgenlere bölündüğünde spiral bir yapı oluşturur. [50].



Şekil 3.2. Altın dikdörtgen geometrik temsili

Şekil 3.2.'deki dikdörtgenlerin iç içe oluşturulduğu düşünülürse; içinden defalarca kareler çıkardığımız bu altın dikdörtgenin karelerinin kenar uzunluklarını yarıçap alan bir çember parçasını her karenin içine çizersek, altın spirali elde ederiz. Altın oranın değeri kullanılarak güzel bir logaritmik spiral ortaya çıkar [50].



**Şekil 3.3.** Altın dikdörtgende spiral yapısının temsili

Sanatçılar, bilim adamları ve tasarımcılar, araştırmalarını yaparken ya da bir ürün ortaya koyarken hesaplamalarını altın orana göre yaratılmış insan bedenini ve doğadaki farklı örnekleri ölçü olarak alırlar. Leonardo da Vinci ve Corbusier tasarımlarını yaparken altın orana göre belirlenmiş insan vücudunu ölçü almışlardır.

Fizikçi B. J. West ile doktor A. L. Goldberger, 1985-1987 arasındaki araştırmalarında akciğerlerin yapısında altın oranın varlığını belirlediler [51]. Akciğerleri oluşturan bronş ağacı asimetriktir. Bu asimetrik bölünme, bronşların ardışık dallanmalarında da sürüp gider. İşte bu bölünmelerin hepsinde kısa bronşun uzun bronşa olan oranının yaklaşık olarak  $1/1,618$  değerini verdiği saptanmıştır.



Şekil 3.4. Doğadaki altın spirale örnekler [52]

Tablo 3.1. Gezegenler arası nispi altın oran değerleri (milyon km.) [52]

Planet	Güneşe olan uzaklık (milyon km.)	Nispi oran
Merkür	57.91	1
Venüs	108.21	1.86859
Dünya	149.60	1.38250
Mars	227.92	1.52353
Ceres	413.79	1.81552
Jüpiter	778.57	1.88154
Satürn	1433.53	1.84123
Uranüs	2872.46	2.00377
Neptün	4495.06	1.56488
Plütön	5869.66	1.30580
Toplam		16.18736
Ortalama		<b>1.61874</b>

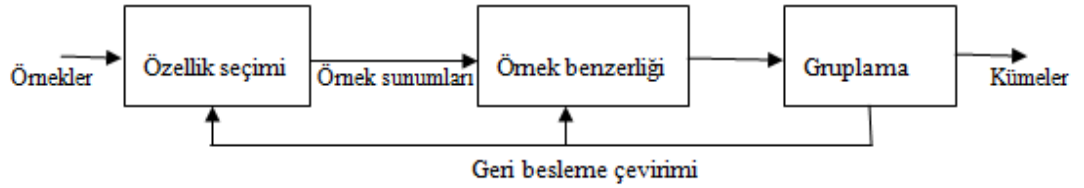
Yukarıda verilenlerden de görüldüğü üzere altın oran değeri ve altın dikdörtgen spiral yapısı, kainatta bir çok yerde muntazam bir şekilde kendini göstermektedir. Zaten bu muntazam ölçüyü koyan yüce yaradan; “Her şeyin bir ölçüye göre yaratıldığını” ve “Bir ölçüyle takdir edildiğini ” bize bildirmektedir.

#### 4. VERİ KÜMELEME

Çoğu zaman kümeleme ve sınıflandırma birbirlerinin yerine kullanılan kavramlar olmalarına rağmen, ikisi farklı şeylerdir. Sınıflandırma, eğitici bir öğrenme metodudur. Fakat kümeleme eğitici bir öğrenme metodudur. Kümeleme, nesnelerin öğreticisiz olarak farklı gruplara (kümelere) ayrıldığı bir sınıflandırma yöntemidir. Aynı küme içerisindeki nesnelere diğer kümelerdeki nesnelere göre birbirlerine daha benzerler [53,54].

Kümeleme işleminde izlenecek yol aşağıdaki gibi tanımlanabilir [54].

- 1- Örneklerin sunulması
- 2- Örneklerin mesafe hesaplamalarının veri seti üzerinden yapılması
- 3- Kümeleme ve gruplama
- 4- Verileri soyutlama
- 5- Çıktının değerlendirilmesi



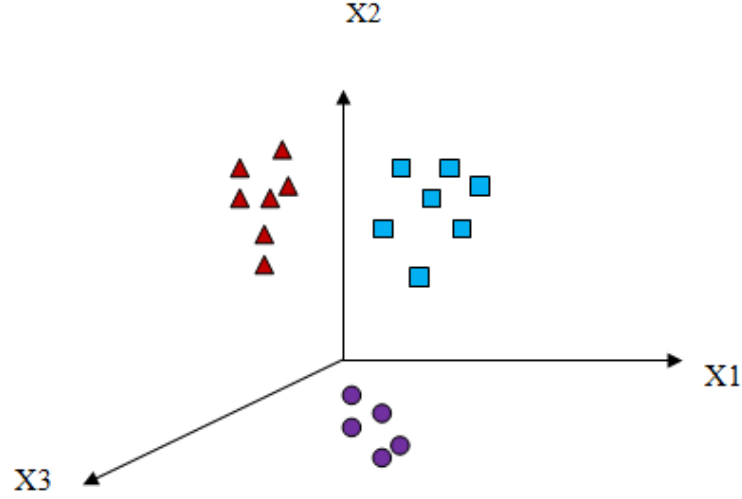
Şekil 4.1. Kümeleme adımları blok temsili [34]

Veri kümeleme yöntemleri, hiyerarşik ve hiyerarşik olmayan yöntemler olarak ikiye ayrılır. Hiyerarşik kümeleme yöntemleri: Tek bağlantı yöntemi, tam bağlantı yöntemi, ortalama bağlantı yöntemi, merkezi bağlantı yöntemi ve varyans yöntemi gibi yöntemlerdir. Hiyerarşik olmayan kümeleme yöntemleri ise: K-ortalamlar yöntemi ve en çok olabilirlik yöntemi gibi yöntemlerdir.

Veri kümelemede kullanılan birçok algoritma vardır [54-57]. Bu algoritmalarda amaç, birbirine çok benzediği düşünülen, fakat özellikleri bakımından birbirinden farklı olan verilerin birbirinden ayrılması ve farklı kümelere bölünmesidir.

Bu tez çalışmasında kümeleme için yukarıda belirtilen klasik kümeleme yöntemleri yerine sezgisel algoritmalar kullanılmıştır. Fakat uzaklık değerleri hesaplamada yine bilinen hesapsal yöntemlerden faydalanılmıştır.





**Şekil 4.2.** Verileri kümelemenin grafiksel temsili

Şekil 4.2.'de X1,X2 ve X3 veri setinin özellikleridir. Bunlar ölçü alındığında benzer özellikleri gösteren veriler aynı gruba denk düşecektir. Böylelikle, veri kümelerine ayrılacaktır. Yukarıda verilen adımlarda geçen “veri soyutlama” işlemi ise bazı farklılıkları göz ardı ederek kümeleri tamamen birbirinden ayırma işlemidir.

#### 4.1. Uzaklık Matrisi Hesaplama Yöntemleri

Verilerin birbirlerine olan benzerlikleri veriler arası uzaklık matrisleri ile belirlenir. Uzaklık matrislerinin hesaplanmasında farklı yöntemler mevcuttur. Her uzaklık ölçütü aşağıdaki şartları sağlar [58].

- 1-  $d(i,j) \geq 0$ , uzaklık fonksiyonunun değeri negatif olamaz.
- 2-  $d(i,i) = 0$ , her nokta çözümün kendisine olan uzaklığı sıfırdır.
- 3-  $d(i,j) = d(j,i)$ , uzaklık fonksiyonunun simetri özelliği vardır.
- 4-  $d(i,j) \leq d(i,h) + d(h,j)$ , Üçgen Eşitsizliği: İki nokta çözümün arasındaki uzaklık bu iki noktanın üçüncü bir noktaya olan uzaklıkları toplamından küçük olamaz.

##### 4.1.1. Öklid uzaklık ölçüsü

Öklid uzaklığında standartlaştırılmış verilerle değil, işlenmemiş verilerle çalışılır. Öklid uzaklıkları yeni nesnelerin eklenmesinden etkilenmezler. Öklid uzaklık formülü en yaygın olarak kullanılan uzaklık hesaplama formülüdür [58].

Öklid uzaklık ölçüsünde iki nokta arasındaki uzaklık denklem (4.17)'deki gibi hesaplanır [58].

$$d(i, j) = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \dots + (x_{ip} - x_{jp})^2} \quad (4.17)$$

#### 4.1.2. Manhattan uzaklık ölçüsü

Manhattan uzaklığı değerler arasındaki mutlak uzaklığa eşittir. Farkın karesi alınmadığı için sıra dışılıkların etkisi azalır. Manhattan uzaklık ölçütüne, “city block uzaklık ölçütü” adı da verilir. Manhattan uzaklık ölçütü kullanılarak iki birim arasındaki uzaklık denklem (4.18)’deki gibi hesaplanır [58].

$$d(i, j) = \left( |x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| + \dots + |x_{ip} - x_{jp}| \right) \quad (4.18)$$

#### 4.1.3. Minkowski uzaklık ölçüsü

Minkowski uzaklık ölçüsü aslında genel bir uzaklık ölçüsü formülüdür. Formüldeki m’nin alacağı farklı değerlere göre yeni formüller üretilir. Minkowski uzaklık ölçüsü kullanılarak iki birim arasındaki uzaklık denklem (4.19)’daki gibi hesaplanır [58].

$$d(i, j) = \left[ |x_{i1} - x_{j1}|^m + |x_{i2} - x_{j2}|^m + \dots + |x_{ip} - x_{jp}|^m \right]^{1/m} \quad (4.19)$$

m = 1 değerini alırsa, formül, Manhattan uzaklık ölçüsünün formülüne, m = 2 değerini alırsak, formül Öklid uzaklık ölçüsü formülüne dönüşür [58].

## 5. ÖNERİLEN YÖNTEMLER ve UYGULAMA SONUÇLARI

Bu tezde geliştirilen 4 yöntemde de BEST matris adlı bir çözüm kümesi oluşturulmaktadır. BEST matrisin boyutu kullanılan veri setinin özellik sayısına göre uygulama yazılımında belirlenmiştir. BEST matrisin elde edilme yöntemi her bölümde ayrıca sunulmuştur.

Uygulamalarda iki farklı veri olan, Iris Plants database ve Wisconsin Breast Cancer Database ile çalışılmıştır.

Literatürde Iris plants veri setini kümele için kullanan bazı yöntemler ve başarı oranları aşağıdaki gibidir :

- 1- K-Dimensional Tree yöntemi, başarı oranı %74 [59]
- 2- Hibrit Fuzzy – Ateş böceği sınıflandırıcı yöntemi, başarı oranı %94,53 [60]
- 3- Modifiye edilmiş Hill Climbing K-Means metodu, başarı oranı 85,30 [61]
- 4- K-Means Optimal Kümeleme için otomatik bölmeleme, başarı oranı %82,31 [62]
- 5- En yakın komşu yöntemi, başarı oranı %97,3 [63]
- 6- Yapay sinir ağları yöntemi, başarı oranı %96,66 [64]
- 7- Yarasa algoritması, başarı oranı %96 [65]
- 8- Parçacık Sürü Optimizasyonu ve Entropi Temelli Bulanık Mantık, başarı oranı %90,67 [66]
- 9- Kernel metot, başarı oranı %97 [67]

Literatürde Wisconsin Breast Cancer veri setini kümele için kullanan bazı yöntemler ve başarı oranları aşağıdaki gibidir :

- 1- Kernel metot yöntemi, başarı oranı %94,7 [67]
- 2- Uyarlamalı Yapay Sinir Ağları, başarı oranı %82,64 [68]
- 3- Fuzzy c-means yöntemi, başarı oranı %93,5 [69]
- 4- Hibrit Fuzzy – Ateş böceği sınıflandırıcı yöntemi, başarı oranı %97,38 [60]
- 5- Destek Vektör Makineleri, başarı oranı 97,08 [70]
- 6- Hibrit Fuzzy Kümeleme, Arı Algoritması ve Destek Vektör Makineleri yöntemi, başarı oranı %98,85 [71]
- 7- Hibrit Ateş böceği ve K-Means yöntemi, başarı oranı %96,13 [72]

Bu çalışmada sunulan yöntemlerde, öncelikle her algoritma için bir başlangıç popülasyonu oluşturulur. Popülasyon BEST matris adayı olabilecek çözümler topluluğudur. Algoritmaya göre popülasyonda oluşturulacak çözüm satır sayısı tasarımcı tarafından belirlenir. Sütun sayısı çalışılacak olan veri setinin özellik sayısı ile aynı olmak zorundadır. Tablo 5.1. genel popülasyon temsili ifade etmektedir. Tablo 5.1.'deki  $m$  tasarımcıya bağlıdır.  $n$  ise veri setindeki özellik sayısına bağlıdır.

**Tablo 5.1.** Başlangıç popülasyonlarının genel temsili

özellik <sub>1</sub>	özellik <sub>2</sub>	.	.	.	özellik <sub>n</sub>
pop(1,1)	pop(1,2)	.	.	.	pop(1,n)
pop(2,1)	pop(2,2)	.	.	.	pop(2,n)
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
pop(m,1)	pop(m,2)	.	.	.	pop(m,n)

BEST matris her uygulama yazılımında farklı yöntemlerle elde edilir. BEST matris elde edilirken uygulama yazılımında oluşturulan popülasyon ile eğitim kümesinin özelliklerinin farkları alınır. Algoritmanın yapısına göre sonuca gidilir ve BEST matris elde edilir.

for  $i=1,2,3,\dots,m$

$$p_i = \sum_{k=1}^n | (\text{popülasyon}(i, k) - \text{eğitim\_kümesi\_özellik}(i, k) |$$

$p_i$  , popülasyondaki her bir çözümün manhattan uzaklık değeri,  
 $\text{popülasyon}(i,k)$ , uygulama yazılımına özel oluşturulmuş çözüm değeri,  
 $\text{eğitim\_kümesi\_özellik}(i,k)$  , eğitim veri seti değerlerinin her biri

Amaç  $p_i$  değerinin minimum olduğu değeri tespit etmektir. Çünkü  $p_i$ 'nin en küçük olduğu çözüm eğitim setinin tümüne en yakın olan optimum çözümdür. Algoritma sonlanma adımına kadar, toplam uzaklık değeri ( hata değeri ) en küçük olan çözüm, popülasyondan çıkan en iyi, yani BEST çözümdür.

**Tablo 5.2.**  $p_i$  değer matrisinin temsili

$p_i(1,1)$	$p_i(1,2)$	$p_i(1,3)$	.	.	.	$p_i(1,n)$
$p_i(2,1)$	$p_i(2,2)$	$p_i(2,3)$	.	.	.	$p_i(2,n)$
.	.	.	.	.	.	.
.	.	.	.	.	.	.
$p_i(m,1)$	$p_i(m,2)$	$p_i(m,3)$	.	.	.	$p_i(m,n)$

$\underbrace{\hspace{15em}}_{\Sigma (\text{fark}(n,:))}$

**Tablo 5.3.**  $p_i$  değerlerinin toplamının matris temsili

$p_{i\_top}(1,1)$	$p_{i\_top}(1,2)$	.	.	.	$p_{i\_top}(1,m)$
-------------------	-------------------	---	---	---	-------------------

Her çözüm için bir uzaklıklar toplamı hesaplanacak olursa toplam  $m$  tane farklar toplamı ortaya çıkar. Tablo 5.3' ün değerleri içerisinde en küçük olan değer, indeks değerine karşılık gelen popülasyondaki çözüm matrisi o anki adımın BEST matrisidir. Örneğin Tablo 5.3.'deki en küçük değer indeks değeri 7. olan çözüm olsun; bu durumda çözüm popülasyonunun 7. satırı o anki adımın BEST matrisi olur. Algoritma sonlanan kadar bu adımlar tekrar edilir. Her seferinde BEST matris güncellenir. En son elde edilen BEST nihai sonuçtur ve BEST matris olarak çıktı verilir.

BEST matris elde edildikten sonra, BEST matrisin test için ayrılan veriler ile manhattan uzaklık farklarının toplamı alınır. Aslında bu uzaklıkların toplamı bizim için ,BEST matrise göre, test verisinin hata değerlerinin toplamı olmaktadır. Bu fark değerlerinin sonuç aralığına göre sınıflar belirlenir. Bu durum şekil 5.1.'de şekilsel olarak temsil edilmiştir.

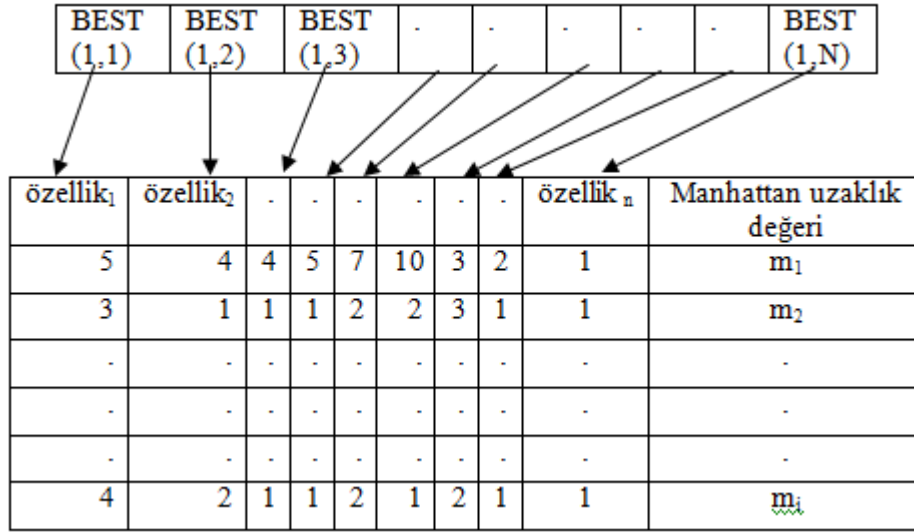
for  $i=1,2,3,\dots,t$

$$m_i = \sum_{k=1}^n | (\text{BEST}(i, k) - \text{test\_kümesi\_özellik}(i, k) |$$

$m_i$  , test değerinin manhattan uzaklık değeri,

$\text{BEST}(i,k)$ , yazılımın üretmiş olduğu BEST matrisin her bir satır elemanı,

$\text{test kümesi özellik}(i,k)$  , test için ayrılan veri setinin her bir satır elemanıdır.



Şekil 5.1. Manhattan uzaklıkları hesabının şekilsel temsili

Geliştirilen yöntemlerin sonuçları göstermiştir ki, BEST matris test gurubunu manhattan uzaklıklarına göre kümelemeyi başarılı bir şekilde yapmıştır. Tablo 5.5.'de uygulama yazılımlarından birinin BEST matrisi ile, test gurubunun arasındaki manhattan uzaklıklarının hesaplamasından elde edilen sonuçlar sunulmuştur. Bu deneme için BEST matris değerleri Tablo 5.4.'de verildiği gibidir.

Tablo 5.4. Örnek uygulama BEST matris değerleri

BEST(1,1)= 4,3321	BEST(1,2)= 2,1088	BEST(1,3)= 1,1588	BEST(1,4)= 0,2431
----------------------	----------------------	----------------------	----------------------

**Tablo 5.5.** Örnek uygulama manhattan uzaklık değerleri örnek temsili

1.test sınıfı hata değerleri					2.test sınıfı hata değerleri					3.test sınıfı hata değerleri				
4,4	3	1,3	0,2	1,1434	5	2,3	3,3	1	3,7572	5,8	2,7	5,1	1,9	7,6572
4,6	3,2	1,4	0,2	1,6434	5,5	2,4	3,7	1	4,7572	6	3	4,8	1,8	7,7572
4,8	3	1,4	0,3	1,6572	5,5	2,4	3,8	1,1	4,9772	6,3	2,7	4,9	1,8	7,8572
4,6	3,4	1,4	0,3	1,8572	5,5	2,5	4	1,3	5,4572	6,4	2,7	5,3	1,9	8,4572
4,9	3,1	1,5	0,1	2,0434	5,7	2,8	4,1	1,3	6,0572	6,5	3	5,2	2	8,8572
4,9	3,1	1,5	0,1	2,0434	5,6	3	4,1	1,3	6,1572	6,5	3,2	5,1	2	8,9572
5	3,3	1,4	0,2	2,1434	5,7	2,9	4,2	1,3	6,2572	6,5	3	5,5	1,8	8,9572
5	3,5	1,3	0,3	2,2572	6,1	2,8	4	1,3	6,3572	6,4	3,1	5,5	1,8	8,9572
5	3,4	1,5	0,2	2,3434	5,7	2,8	4,5	1,3	6,4572	6,4	2,8	5,6	2,1	9,0572
5,1	3,4	1,5	0,2	2,4434	5,6	3	4,5	1,5	6,7572	6,7	3	5,2	2,3	9,3572
5,1	3,5	1,4	0,3	2,4572	6	2,9	4,5	1,5	7,0572	6,9	3,1	5,1	2,3	9,5572
4,8	3,4	1,9	0,2	2,5434	6,4	2,9	4,3	1,3	7,0572	6,3	3,4	5,6	2,4	9,8572
5	3,4	1,6	0,4	2,5572	6,1	2,9	4,7	1,4	7,2572	6,7	3,3	5,7	2,1	9,9572
5,2	3,5	1,5	0,2	2,6434	6,3	2,5	4,9	1,5	7,3572	6,9	3,2	5,7	2,3	10,2572
5,1	3,7	1,5	0,4	2,8572	6,6	2,9	4,6	1,3	7,5572	7,2	3,2	6	1,8	10,3572
5,4	3,4	1,5	0,4	2,8572	6	2,7	5,1	1,6	7,5572	7,3	2,9	6,3	1,8	10,4572
5,1	3,8	1,6	0,2	2,9434	6,7	3,1	4,4	1,4	7,7572	7,7	2,8	6,7	2	11,3572
5,4	3,7	1,5	0,2	3,0434	6,8	2,8	4,8	1,4	7,9572	7,6	3	6,6	2,1	11,4572
5,4	3,9	1,7	0,4	3,5572	6,7	3,1	4,7	1,5	8,1572	7,9	3,8	6,4	2	12,2572
5,7	3,8	1,7	0,3	3,6572	6,9	3,1	4,9	1,5	8,5572	7,7	3,8	6,7	2,2	12,5572

Tablo 5.5. dikkatlice incelenecek olursa, her test sınıfının örnekleri için manhattan uzaklık değerleri belirli bir değer aralığına çekilmektedir. Tablo 5.5.'de ilk sınıfın manhattan uzaklık değeri (1,1434-3,6572) aralığında, ikinci sınıfın manhattan uzaklık değeri (3,7572-8,5572) aralığında, üçüncü sınıfın manhattan uzaklık değeri ise (8,8572-12,5572) aralığında çıkmaktadır. Bu durumda üçüncü gurubun 4 değeri hatalı kümelenebilmektedir. Doğruluk oranı aşağıdaki gibi hesaplanmaktadır.

$$\text{Doğruluk oranı} = (100/\text{Örnek sayısı}) * \text{Doğru hesaplanan örnek sayısı} \quad (5.20)$$

Tablo 5.5.'deki testin doğruluk oranı %93,33 çıkmaktadır.

Bu çözüm yöntemiyle 4 farklı yöntem için elde edilen sonuçlar bölüm 5.1., 5.2., 5.3. ve 5.4.'de sunulmuştur.

## 5.1. Yapay Atom Algoritması Uygulama Yazılımı ve Sonuçları

Tez çalışmasının bu bölümünde Iris Plants Database ile çalışılmıştır [73].

Geliştirilen yazılım sonuç olarak BEST matris isimli 1X4 boyutlu bir matris vermektedir. Çünkü veri seti (dataset) 4 farklı özellik içermektedir. Bu matrisin eğitim kümesi ile farklarının toplamı hata değerleri olacaktır.

**Tablo 5.6.** Yapay atom algoritması uygulama yazılımı BEST matris temsili

BEST(1,1)	BEST(1,2)	BEST(1,3)	BEST(1,4)
-----------	-----------	-----------	-----------

Veri setinin özellikleri Tablo 5.7.'de gösterilmiştir.

**Tablo 5.7.** Iris Dataset veri örnekleri temsili

sepal length	sepal width	petal length	petal width	class
5,4	3,7	1,5	0,2	1
5,4	3,4	1,5	0,4	1
5,7	3,8	1,7	0,3	1
5	2,3	3,3	1	2
5,5	2,4	3,8	1,1	2
5,5	2,4	3,7	1	2
5,8	2,7	5,1	1,9	3
6	3	4,8	1,8	3
6,3	2,7	4,9	1,8	3
6,3	3,4	5,6	2,4	3

Sepal length : çanak yaprak uzunluğunu,

Sepal width : çanak yaprak genişliğini,

Petal length : taç yaprak uzunluğunu,

Petal width : taç yaprak genişliğini ifade etmektedir.

Bu örnek değerlere karşılık class değeri ise bitkinin sınıfını göstermektedir. Uygulama yazılımında bu özellik kullanılmayacaktır. Çünkü kümeleme işleminde, sınıf özellik değeri kullanılmadan kümeler belirlenmeye çalışılır.



Veri setinde 150 farklı örnek mevcuttur. Her bir sınıftan 50 adet örnek vardır. Geliştirilen uygulama yazılımında, her sınıf için 30 örnek algoritmanın eğitim verileri için kullanılmıştır. Her sınıftan 20 örnekte test için bırakılmıştır.

Uygulama yazılımı için kovalent ve iyonik bağları temsilen oluşturulan toplam 100 elemanlı bir çözüm kümesi oluşturulmuştur. Popülasyon oluşturulurken her kümenin en küçük ve en büyük değerlerinden faydalanılır. Amaç sınır değerleri taşımayan bir popülasyon çözümü oluşturabilmektir.

```
kovalent(:,1)=4.3+(7.7-4.3).*rand(n,1)
kovalent(:,2)=2+(4.4-2).*rand(n,1)
kovalent(:,3)=1+(6.9-1).*rand(n,1)
kovalent(:,4)=0.1+(2.5-0.1).*rand(n,1)
```

```
iyonik(:,1)=4.3+(7.7-4.3).*rand(m,1)
iyonik(:,2)=2+(4.4-2).*rand(m,1)
iyonik(:,3)=1+(6.9-1).*rand(m,1)
iyonik(:,4)=0.1+(2.5-0.1).*rand(m,1)
```

Burada n ve m değerleri oluşturulacak 100 elemanlı popülasyonun indeks değerleri olup; uygulamada değişik kombinasyonlarda denenmişlerdir.

**Tablo 5.8.** Kovalent bağ çözüm temsili

K <sub>11</sub>	K <sub>12</sub>	K <sub>13</sub>	K <sub>14</sub>
K <sub>21</sub>	K <sub>22</sub>	K <sub>23</sub>	K <sub>24</sub>
K <sub>31</sub>	K <sub>32</sub>	K <sub>33</sub>	K <sub>34</sub>
K <sub>41</sub>	K <sub>42</sub>	K <sub>43</sub>	K <sub>44</sub>
.	.	.	.
.	.	.	.
K <sub>n1</sub>	K <sub>n2</sub>	K <sub>n3</sub>	K <sub>n4</sub>

**Tablo 5.9.** İyonik bağ çözüm temsili

$\dot{I}_{11}$	$\dot{I}_{12}$	$\dot{I}_{13}$	$\dot{I}_{14}$
$\dot{I}_{21}$	$\dot{I}_{22}$	$\dot{I}_{23}$	$\dot{I}_{24}$
$\dot{I}_{31}$	$\dot{I}_{32}$	$\dot{I}_{33}$	$\dot{I}_{34}$
$\dot{I}_{41}$	$\dot{I}_{42}$	$\dot{I}_{43}$	$\dot{I}_{44}$
.	.	.	.
.	.	.	.
$\dot{I}_{m1}$	$\dot{I}_{m2}$	$\dot{I}_{m3}$	$\dot{I}_{m4}$

Denklem (5.21)'de verildiği üzere rastgele oluşturulan popülasyonun eğitim kümesi ile manhattan uzaklıklarının toplamları hesaplanır.

$$\|x_i - x_j\| = \sum_{k=1}^d |(x_{i,k} - x_{j,k})| \quad (5.21)$$

**Tablo 5.10.** Yapay atom algoritması uygulama yazılımı farklar matrisinin temsili

fark(1,1)	fark (1,2)	fark(1,3)	fark (1,4)
fark (2,1)	fark (2,2)	fark(2,3)	fark (2,4)
.	.		.
.	.		.
fark (100,1)	fark(100,2)	fark(100,3)	fark(100,4)

$\underbrace{\hspace{15em}}_{\sum (\text{fark}(k,:))}$

Her çözüm için bir farklar toplamı hesaplanacak olursa toplam 100 tane farklar toplamı ortaya çıkar.

**Tablo 5.11.** Yapay atom algoritması uygulama yazılımı farklar matrisinin toplamının temsili

farktop(1,1)	farktop(1,2)	.	.	.	farktop(1,100)
--------------	--------------	---	---	---	----------------

farktop matrisinin değerleri küçükten büyüğe doğru sıralanır ve buna uygun olarak popülasyon çözümleri yeni baştan düzenlenirse elimizde toplam fark değerlerine göre sıralanmış bir popülasyon oluşur. Bu sıralanmış popülasyonunda 1. çözümü en iyi çözüm olur. Bu işleme algoritma sonlanana kadar devam edilir. Sonuçta en son adımın elde ettiği popülasyonun 1. çözümü BEST matris olarak seçilmiş olur.

Yapay atom algoritması yapısında parametre barındırmayan bir algoritmadır. Yeni değerleri oluşturmada hesapsal bir yöntem mevcut değildir. En iyi çözümleri bir sonraki jenerasyona taşıırken, bir çözüm kendisinden daha iyi olan bir çözüm değerini üzerine

kopyalar. Yani bir nevi elektron ortaklaşması gibi iyi çözümler ortaklaşırlar. İyonik çözümler ise her jenerasyonda belirli bir oranda yeniden üretilir.

Bu amaçla uygulamada Yapay atom algoritmasının altın oran ile denemeleri kovalent ve iyonik bağ çözümlerinin oranlarını ayarlamakta denenmiştir. Yani Tablo 5.8.ve Tablo 5.9.'da verilen n ve m değerlerinin oranının, altın oran etrafındaki etkileri gözlemlenmiştir. Ayrıca her bir çözüm 4 özellikten oluştuğundan, ortaklaşılan çözümlerin sayısının çözüme etkisini görmek amacıyla, her adım için ortaklaşılan 4 özellik değerinin değişimi de ayrı ayrı ele alınmıştır.

**Adım 1:** Bu adımda popülasyondaki kovalent çözüm oranı %30, iyonik çözüm oranı ise %70'dir. 4 farklı özelliğin sadece ilki değiştiğinde doğruluk oranları Tablo 5.12.'de, iki özellik değiştiğinde doğruluk oranları Tablo 5.13.'de, üç özellik değiştiğinde doğruluk oranları Tablo 5.14.'de ve 4 özellik de değiştiğinde doğruluk oranları Tablo 5.15.'de verildiği gibidir.

**Tablo 5.12.** Yapay atom algoritması Adım 1 için 1 özellik değiştiğinde sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Doğruluk oranı (%)
5,8127	3,0527	4,2423	1,3318	91,67
5,8352	2,9950	4,2515	1,3678	90
5,7522	2,9615	4,2277	1,3749	90
5,6593	2,9996	4,2249	1,3205	91,67
5,7577	3,0303	4,2955	1,3846	90
5,6871	2,9989	4,2131	1,2308	90
5,7702	3,0094	4,2932	1,3013	90
5,7968	3,0349	4,2135	1,2300	91,67
5,6658	3,0198	4,2139	1,3235	90
5,8024	3,0312	4,2004	1,2658	90
				Ortalama = 90,50

**Tablo 5.13.** Yapay atom algoritması Adım 1 için 2 özellik değiştiğinde sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Doğruluk oranı (%)
5,7630	3,0089	4,2808	1,3006	90
5,7784	3,0163	4,2112	1,3005	90
5,7248	2,9863	4,2497	1,3299	90
5,7350	3,0016	4,2108	1,2954	91,67
5,7863	3,0035	4,2576	1,3102	90
5,7989	3,0036	4,2727	1,3057	90
5,7803	2,9974	4,2373	1,2947	90
5,7808	3,0003	4,2996	1,2799	90
5,7514	2,9902	4,2127	1,3007	90
5,7748	3,0009	4,2314	1,3000	90
				Ortalama = 90,17

**Tablo 5.14.** Yapay atom algoritması Adım 1 için 3 özellik deđiřtiđinde sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Dođruluk oranı (%)
5,7532	3,0064	4,2072	1,3021	90
5,7909	2,9986	4,2979	1,3063	90
5,7851	3,0042	4,2741	1,2963	90
5,7817	2,9987	4,2871	1,2884	90
5,8000	2,9985	4,2574	1,3070	90
5,7980	2,9994	4,2563	1,3000	90
5,7636	2,9969	4,2236	1,2998	90
5,7991	3,0023	4,2689	1,3066	90
5,7757	3,0047	4,2827	1,3026	90
5,7874	2,9999	4,2931	1,3093	90
				Ortalama = 90

**Tablo 5.15.** Yapay atom algoritması Adım 1 için 4 özellik deđiřtiđinde sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Dođruluk oranı (%)
5,7830	2,9989	4,3074	1,2446	90
5,7328	3,0079	4,2811	1,2943	90
5,7013	3,0004	4,2125	1,3239	91,67
5,7901	3,0085	4,2898	1,2713	90
5,7920	2,9876	4,2864	1,2879	90
5,7518	2,9969	4,2722	1,3064	90
5,7489	3,0201	4,2881	1,2972	90
5,7868	3,0073	4,2442	1,2698	90
5,7615	3,0039	4,2465	1,3032	90
5,7728	3,0081	4,2791	1,3061	90
				Ortalama = 90,17

**Adım 2:** Bu adımda popülasyondaki kovalent çözüm oranı %40, iyonik çözüm oranı ise %60'dir. 4 farklı özelliđin sadece ilki deđiřtiđinde dođruluk oranları Tablo 5.16.'da, iki özellik deđiřtiđinde dođruluk oranları Tablo 5.17.'de, üç özellik deđiřtiđinde dođruluk oranları Tablo 5.18.'de ve 4 özellik de deđiřtiđinde dođruluk oranları Tablo 5.19.'da verildiđi gibidir.

**Tablo 5.16.** Yapay atom algoritması Adım 2 için 1 özellik deđiřtiđinde sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Dođruluk oranı (%)
5,7723	2,9846	4,2414	1,3304	90
5,7276	2,9817	4,2308	1,3456	90
5,7559	2,9582	4,2653	1,2932	90
5,8321	2,9596	4,2163	1,3018	90
5,7638	2,9658	4,2805	1,3055	90
5,7043	3,0287	4,2093	1,3857	90
5,7234	3,0006	4,2771	1,4187	90
5,8538	2,9984	4,2138	1,3574	90
5,7454	3,0156	4,2582	1,3748	90
5,8072	3,0455	4,2079	1,2507	90
				Ortalama = 90

**Tablo 5.17.** Yapay atom algoritması Adım 2 için 2 özellik deđiřtiđinde sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Dođruluk oranı (%)
5,7743	3,0071	4,2548	1,2991	90
5,7840	2,9843	4,2604	1,2936	90
5,7475	2,9901	4,2925	1,2972	90
5,7301	2,9968	4,2059	1,2995	91,67
5,7726	2,9853	4,2491	1,2999	90
5,8127	2,9947	4,2826	1,3471	90
5,8076	3,0126	4,2657	1,29992	90
5,7896	2,9975	4,2831	1,2760	90
5,7508	2,9989	4,2878	1,3214	90
5,7671	2,9970	4,2738	1,2911	90
				Ortalama = 90,17

**Tablo 5.18.** Yapay atom algoritması Adım 2 için 3 özellik deđiřtiđinde sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Dođruluk oranı (%)
5,7971	2,9989	4,2718	1,3070	90
5,7909	2,9999	4,2679	1,2938	90
5,7968	3,0048	4,2954	1,3022	90
5,8015	3,0018	4,2491	1,3016	90
5,7908	2,9985	4,2078	1,2942	90
5,7957	3,0004	4,2614	1,3039	90
5,7659	2,9954	4,2610	1,2945	90
5,8005	3,0141	4,2506	1,3033	90
5,7790	2,9956	4,2375	1,3286	90
5,7892	3,0027	4,2289	1,2984	90
				Ortalama = 90

**Tablo 5.19.** Yapay atom algoritması Adım 2 için 4 özellik deđiřtiđinde sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Dođruluk oranı (%)
5,7457	2,9974	4,2810	1,3290	90
5,7697	3,0080	4,2531	1,3410	90
5,8037	2,9978	4,2515	1,2883	90
5,8051	3,0001	4,2585	1,3389	90
5,7871	3,0029	4,2865	1,3594	90
5,7762	2,9886	4,2086	1,3041	90
5,7975	3,0004	4,2639	1,2961	90
5,6886	3,0213	4,2961	1,3218	90
5,7879	3,0182	4,3090	1,3004	90
5,8076	2,9926	4,2046	1,2900	90
				Ortalama = 90

**Adım 3:** Bu adımda popülasyondaki kovalent çözüm oranı %50, iyonik çözüm oranı ise %50'dir. 4 farklı özelliğin sadece ilki değiştiğinde doğruluk oranları Tablo 5.20.'de, iki özellik değiştiğinde doğruluk oranları Tablo 5.21.'de, üç özellik değiştiğinde doğruluk oranları Tablo 5.22.'de ve 4 özellik de değiştiğinde doğruluk oranları Tablo 5.23.'de verildiği gibidir.

**Tablo 5.20.** Yapay atom algoritması Adım 3 için 1 özellik değiştiğinde sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Doğruluk oranı (%)
5,7826	3,0179	4,2815	1,2799	90
5,7748	2,9933	4,2861	1,2933	90
5,6820	3,0049	4,2505	1,2920	91,67
5,7777	2,9760	4,2248	1,3160	90
5,6515	3,0267	4,2150	1,4084	90
5,7490	3,0116	4,2383	1,3320	90
5,7584	3,0059	4,2003	1,1634	91,67
5,7901	2,9890	4,2031	1,2459	90
5,7842	2,9971	4,2218	1,2425	90
5,7671	3,0092	4,2967	1,3145	90
				Ortalama = 90,33

**Tablo 5.21.** Yapay atom algoritması Adım 3 için 2 özellik değiştiğinde sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Doğruluk oranı (%)
5,8027	2,9981	4,2140	1,2987	90
5,6723	3,0005	4,2542	1,3117	91,67
5,8082	3,0058	4,2944	1,3014	90
5,7662	3,0071	4,2249	1,2900	90
5,7312	2,9939	4,2688	1,2977	90
5,8066	2,9903	4,2980	1,2877	90
5,7925	3,0020	4,2694	1,2951	90
5,7486	2,9983	4,2927	1,3078	90
5,7662	3,0169	4,2568	1,2809	90
5,7723	2,9858	4,2399	1,3026	90
				Ortalama = 90,17

**Tablo 5.22.** Yapay atom algoritması Adım 3 için 3 özellik değiştiğinde sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Doğruluk oranı (%)
5,8027	2,9987	4,2856	1,2993	90
5,7985	3,0001	4,2293	1,3065	90
5,7737	2,9997	4,2252	1,3098	90
5,7853	2,9939	4,2929	1,3173	90
5,7970	3,0005	4,2857	1,3090	90
5,7927	2,9933	4,2094	1,2955	90
5,7853	3,0001	4,2372	1,3202	90
5,8010	3,0012	4,2376	1,2970	90
5,7743	3,0027	4,2642	1,3198	90
5,7657	2,9963	4,2326	1,3022	90
				Ortalama = 90

**Tablo 5.23.** Yapay atom algoritması Adım 3 için 4 özellik deđiřtiđinde sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Dođruluk oranı (%)
5,7670	3,0001	4,3395	1,3279	90
5,7681	3,0162	4,2111	1,3050	90
5,8089	3,0013	4,3410	1,3154	90
5,7649	2,9909	4,2560	1,3251	90
5,7488	2,9951	4,2059	1,3027	90
5,7395	2,9753	4,3007	1,2911	90
5,7710	3,0235	4,2342	1,2989	90
5,7627	2,9862	4,2125	1,3038	90
5,7727	3,0032	4,2629	1,3239	90
5,7615	3,0081	4,1850	1,2664	91,67
				Ortalama = 90,17

**Adım 4:** Bu adımda popülasyondaki kovalent çözüm oranı %62, iyonik çözüm oranı ise %38'dir. 4 farklı özelliđin sadece ilki deđiřtiđinde dođruluk oranları Tablo 5.24.'de, iki özellik deđiřtiđinde dođruluk oranları Tablo 5.25.'de, üç özellik deđiřtiđinde dođruluk oranları Tablo 5.26.'da ve 4 özellik de deđiřtiđinde dođruluk oranları Tablo 5.27.'de verildiđi gibidir.

**Tablo 5.24.** Yapay atom algoritması Adım 4 için 1 özellik deđiřtiđinde sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Dođruluk oranı (%)
5,8542	3,0121	4,2836	1,4034	88,33
5,7048	2,9736	4,2710	1,2730	90
5,8754	2,9961	4,2070	1,2907	90
5,8672	3,0105	4,2215	1,2998	90
5,7111	3,0646	4,2127	1,3008	91,67
5,7864	3,0198	4,2241	1,2779	90
5,6896	2,9737	4,2576	1,3904	90
5,6501	3,0007	4,2150	1,3832	91,67
5,7361	3,0198	4,2859	1,2817	90
5,8031	2,9363	4,2152	1,2398	90
				Ortalama = 90,17

**Tablo 5.25.** Yapay atom algoritması Adım 4 için 2 özellik deđiřtiđinde sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Dođruluk oranı (%)
5,8093	2,9998	4,2208	1,2916	90
5,7713	3,0008	4,2804	1,3013	90
5,7611	3,0052	4,2862	1,3083	90
5,7169	2,9976	4,2651	1,3083	90
5,7903	3,0026	4,2393	1,3213	90
5,7357	2,9957	4,2516	1,2979	90
5,7982	2,9941	4,2310	1,3232	90
5,7605	3,0097	4,2661	1,2995	90
5,7784	3,0010	4,2639	1,2973	90
5,7901	2,9875	4,2491	1,2992	90
				Ortalama = 90

**Tablo 5.26.** Yapay atom algoritması Adım 4 için 3 özellik deđiřtiđinde sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Dođruluk oranı (%)
5,7917	3,0017	4,2689	1,2972	90
5,7998	2,9990	4,2460	1,3004	90
5,8008	3,0006	4,2153	1,2909	90
5,7928	2,9998	4,2778	1,3062	90
5,7464	3,0012	4,2156	1,3014	90
5,8015	2,9995	4,2220	1,3078	90
5,7905	2,9981	4,2911	1,2962	90
5,7728	2,9990	4,2729	1,3021	90
5,8004	3,0040	4,2513	1,3148	90
5,7999	3,0171	4,2840	1,2890	90
				Ortalama = 90

**Tablo 5.27.** Yapay atom algoritması Adım 4 için 4 özellik deđiřtiđinde sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Dođruluk oranı (%)
5,7375	3,0080	4,2806	1,3313	90
5,6879	2,9970	4,2694	1,3200	90
5,7461	2,9919	4,2608	1,3029	90
5,7793	3,0033	4,4425	1,2932	88,33
5,7480	3,0031	4,2744	1,3426	90
5,7912	2,9841	4,2202	1,2649	90
5,7203	3,0112	4,2183	1,2943	91,67
5,8001	3,0061	4,3453	1,3012	90
5,8055	2,9993	4,3582	1,2992	90
5,7852	3,0094	4,3451	1,3175	90
				Ortalama = 90

**Adım 5:** Bu adımda popülasyondaki kovalent çözüm oranı %80, iyonik çözüm oranı ise %20'dir. 4 farklı özelliđin sadece ilki deđiřtiđinde dođruluk oranları Tablo 5.28.'de, iki özellik deđiřtiđinde dođruluk oranları Tablo 5.29.'da, üç özellik deđiřtiđinde dođruluk oranları Tablo 5.30.'da ve 4 özellik de deđiřtiđinde dođruluk oranları Tablo 5.31.'de verildiđi gibidir.

**Tablo 5.28.** Yapay atom algoritması Adım 5 için 1 özellik deđiřtiđinde sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Dođruluk oranı (%)
5,7493	30462	4,2334	1,2543	90
5,7272	2,9985	4,2120	1,3224	90
5,7738	2,9713	4,2663	1,4242	90
5,8793	3,0090	4,2709	1,2933	90
5,7330	2,9458	4,2516	1,3994	90
5,7505	2,9924	4,2549	1,2987	90
5,7167	2,9373	4,2651	1,3575	90
5,8580	3,0333	4,2872	1,3098	90
5,7738	3,0255	4,3048	1,2734	90
5,6819	2,9948	4,2548	1,2420	90
				Ortalama = 90



**Tablo 5.29.** Yapay atom algoritması Adım 5 için 2 özellik deđiřtiđinde sonuçlar

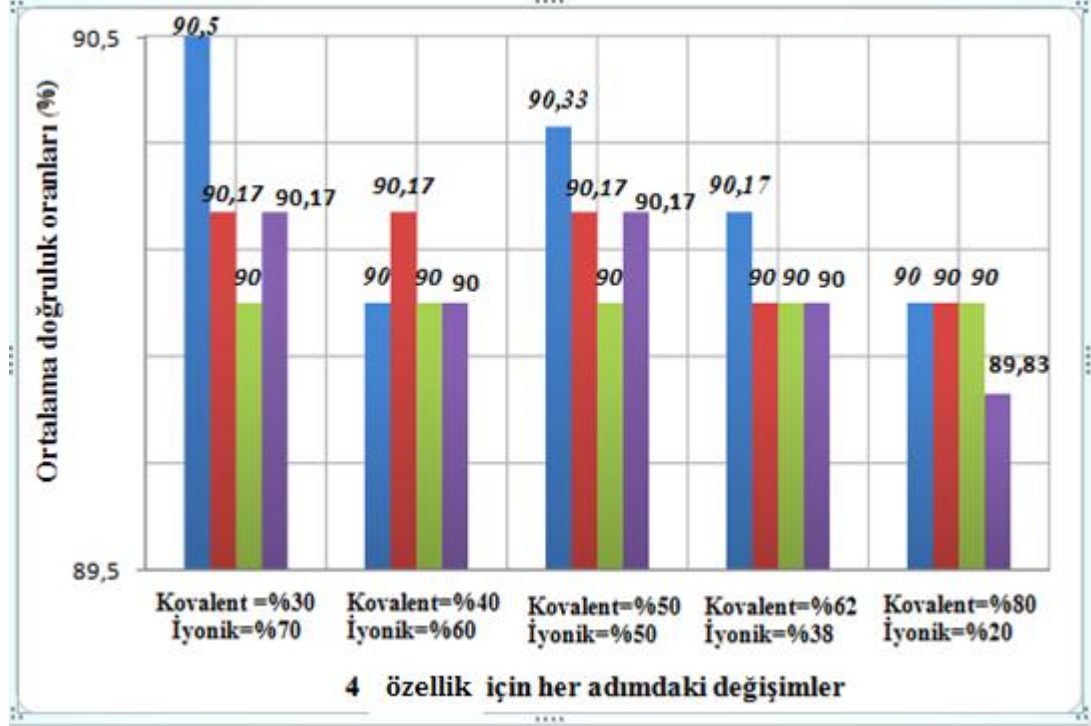
BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Dođruluk oranı (%)
5,7607	3,0098	4,2801	1,2974	90
5,7035	2,9983	4,2574	1,2908	90
5,7859	3,0016	4,2247	1,2993	90
5,7407	3,0022	4,2339	1,2969	90
5,7824	2,9963	4,2213	1,2921	90
5,7572	3,0094	4,2921	1,3150	90
5,7677	3,0063	4,2822	1,3061	90
5,7635	3,0065	4,2357	1,2979	90
5,8318	2,9962	4,2388	1,2974	90
5,7853	2,9995	4,2134	1,2981	90
				Ortalama = 90

**Tablo 5.30.** Yapay atom algoritması Adım 5 için 3 özellik deđiřtiđinde sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Dođruluk oranı (%)
5,7969	2,9931	4,2451	1,3002	90
5,7988	3,0008	4,2717	1,2934	90
5,7788	3,0000	4,2547	1,3122	90
5,7930	3,0044	4,2926	1,3013	90
5,7959	2,9976	4,2025	1,3005	90
5,7692	3,0018	4,2417	1,3018	90
5,7992	2,9974	4,2008	1,3543	90
5,7999	2,9998	4,2465	1,2998	90
5,7853	2,9985	4,2937	1,3042	90
5,8016	2,9996	4,2923	1,2975	90
				Ortalama = 90

**Tablo 5.31.** Yapay atom algoritması Adım 5 için 4 özellik deđiřtiđinde sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Dođruluk oranı (%)
5,7345	3,0005	4,2952	1,2986	90
5,7782	3,0048	4,2857	1,2796	90
5,7716	3,0044	4,4009	1,3153	90
5,7718	2,9960	4,2410	1,3222	90
5,7891	2,9985	4,2900	1,3110	90
5,7268	3,0143	4,2065	1,3612	90
5,7966	2,9520	4,2023	1,2607	90
5,7959	3,0212	4,4019	1,3280	88,33
5,7500	3,0131	4,3019	1,3548	90
5,7921	2,9968	4,2959	1,2941	90
				Ortalama = 89,83



Şekil 5.2. Yapay atom algoritması tüm çözüm durumları için ortalama doğruluk oranları grafik temsili

Uygulamadaki 5 adımın tüm durumları göz önüne alındığına görülmüştür ki; Yapay atom algoritmasının yeni değerleri üretmede, çözüm ortaklaşma yapması başarı oranının %90,5'e kadar gelmesini sağlayabilmiştir.

Sezgisel yöntemlerde, her jenerasyonda oluşan iyi çözümler etrafında oluşturulacak yeni çözümler, belirli bir formül ile oluşturulur. Bu yeni çözümleri oluştururken, en iyi çözüm etrafında bazı parametreler yardımıyla yeni çözümler hesaplanır. Yapay atom algoritmasında parametre yoktur. Sadece popülasyonun kovalent ve iyonik çözüm oranı belirlenebilir. Yeni çözüm, özellik ortaklaşma ile oluşturulur. Buda altın oranın etkisini göstermesine yeterli olmamıştır.

Popülasyonun kovalent ve iyonik çözüm oranının %62-%38 olduğu yerde, altın oran değeri sebebiyle en yüksek doğruluk oranı olması beklenirken; kovalent ve iyonik çözüm oranının %30-%70 olduğu yerde %90,5 başarı oranı ile en yüksek doğruluk oranı gerçekleşmiştir.

## 5.2. Parçacık Sürü Optimizasyon Algoritması Uygulama Yazılımı ve Sonuçları

Tez çalışmasının bu bölümünde, Bölüm 5.1.'de kullanılan Iris-Dataset ile çalışılmıştır. Uygulamada aynı veri seti ile çalışmaktaki amacımız, iki farklı algoritmanın aynı veri gurubu üzerindeki sonuçlarını karşılaştırabilmek içindir. Aynı zamanda altın oran uygulamalarındaki etkileri gözlemleyebilmektir. Geliştirilen yazılım yine sonuç olarak BEST matris isimli 1X4 boyutlu bir matris vermektedir. Bu matrisin eğitim kümesi ile farklarının toplamı yine hata değerleri olacaktır.

Veri setinde 150 farklı örnek mevcuttur. Her bir sınıftan 50 adet örnek vardır. Geliştirilen uygulama yazılımında, her sınıf için 30 örnek algoritmanın eğitim verileri için kullanılmıştır. Her sınıftan 20 örnekte test için bırakılmıştır. Çözüm için 16 elemanlı bir çözüm matrisi oluşturulmuştur.

**Tablo 5.32.** Parçacık sürü optimizasyon algoritması uygulama yazılımı BEST matris temsili

BEST(1,1)	BEST(1,2)	BEST(1,3)	BEST(1,4)
-----------	-----------	-----------	-----------

$$\|x_i - x_j\| = \sum_{k=1}^d |(x_{i,k} - x_{j,k})| \quad (5.22)$$

Denklem (5.22)'de verildiği üzere rastgele oluşturulan popülasyonun eğitim kümesi ile manhattan uzaklıklarının toplamları hesaplanır.

```
for k=1,2,...,16
  for m=1,2,3,4
    fark(k,m)=sum(abs(pop(k,m)-egitim(:,m)));
```

**Tablo 5.33.** Parçacık sürü optimizasyon algoritması uygulama yazılımı farklar matrisinin temsili

fark(1,1)	fark (1,2)	fark(1,3)	fark (1,4)
fark (2,1)	fark (2,2)	fark(2,3)	fark (2,4)
.	.	.	.
.	.	.	.
fark (16,1)	fark(100,2)	fark(100,3)	fark(16,4)

$$\sum (\text{fark}(k,:))$$

Her çözüm için bir farklar toplamı hesaplanacak olursa toplam 16 tane farklar toplamı ortaya çıkar.

**Tablo 5.34.** Parçacık sürü optimizasyon algoritması uygulama yazılımı farklar matrisinin toplamının temsili

farktop(1,1)	farktop(1,2)	.	.	.	farktop(1,16)
--------------	--------------	---	---	---	---------------

Parçacık sürü optimizasyon algoritmasında lokal en iyi olan pbest ve global en iyi olan gbest kavramlarımız vardır. pbest o jenerasyondaki en iyi değer, gbest ise tüm jenerasyonların en iyi değeridir.

```
for k=1,2,...,16
  for m=1,2,3,4
    bestfarklar(k,m)=sum(abs(pbest(k,m)-egitim(:,m)));
```

İlk başta pbest her çözüm için popülasyondaki çözümün kendisine eşittir. Yani pbest çözüm matrisinin boyutu ve değerleri başlangıçta popülasyon ile aynıdır.

$$\begin{bmatrix} \text{pop}(1,1) & \cdots & \text{pop}(1,n) \\ \vdots & & \vdots \\ \text{pop}(n,1) & \cdots & \text{pop}(n,n) \end{bmatrix} = \begin{bmatrix} \text{pbest}(1,1) & \cdots & \text{pbest}(1,n) \\ \vdots & & \vdots \\ \text{pbest}(n,1) & \cdots & \text{pbest}(n,n) \end{bmatrix}$$

Eğer bir önceki jenerasyondaki pbest değerlerden birinin öklit uzaklık farkı, o anki çözümün değerinin öklit uzaklık farkından küçük ise; yeni çözüm pbest matristeki değerin yerine yazılır. Yani eski pbest çözümden daha iyi bir çözüm o jenerasyon için bulunmuştur.

```
for k=1:16
  popfarktoplami(1,k)=sum(popfarklar(k,:));
  bestfarklartoplami(1,k)=sum(bestfarklar(k,:));
  if popfarklartoplami(1,k)<bestfarklartoplami(1,k)
    pbest(k,:)=pop(k,:);
```

O ana kadarki pbest değerlerin içerisinde eğitim kümesi ile farklarının toplamı en küçük olan ise gbest değeridir.

```
[a,b]=min(bestfarklartoplami(1,:))
gbestson(1,:)=pbest(b,:)
```

Parçacık sürü optimizasyon algoritmasının hız ve pozisyon denklemlerini tekrar hatırlayacak olursak;

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

$$\text{eğer } v_i^{k+1} > v_{\max}, v_i^{k+1} = v_{\max}$$

$$\text{eğer } v_i^{k+1} < v_{\min}, v_i^{k+1} = v_{\min}$$

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

Uygulama yazılımında  $v_{\max}$  olarak her özelliğin sınır değerlerinin farklarının pozitif değeri,  $v_{\min}$  olarak da negatif değeri kabul edilmiştir.

Örneğin veri setindeki 1. özellik için en büyük değer 7,7; en küçük değer 4,3'dür.

O halde

$$v_{\max} = (7,7 - 4,3) = 3,4$$

$v_{\min} = -3,4$  olarak kabul edilmiştir.

Tezin bu kısmında geliştirilen uygulama yazılımında C1 ve C2 öğrenme faktörleri, hem altın oran değerinde ve altın oran değerine yakın değerlerde; hem de altın oran değerinden uzak olacak şekilde farklı değerler etrafında denenmiştir. Aynı şekilde denklem (2.1)'de verilen, pbest ve gbest değerlerinin çarpıldığı  $rand_1$  ve  $rand_2$  değerleri hem altın oran değerinde ve altın oran değerine yakın değerlerde; hem de altın oran değerinden uzak olacak şekilde farklı değerler etrafında denenmiştir. Elde edilen sonuçlar sıradaki 12 adımda sunulmuştur.

#### Adım 1:

a. Öğrenme faktörleri  $C1=C2=0,2$ ,  $rand_1=rand$  ve  $rand_2=rand$  olduğunda doğruluk oranları Tablo 5.35.'de verildiği gibidir.

**Tablo 5.35.** Parçacık sürü optimizasyonu Adım 1 a. için sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Doğruluk oranı (%)
5,4259	3,1164	3,7961	1,3523	88,33
5,9493	3,0283	3,8326	1,5583	90
6,1320	2,8606	4,3350	1,2388	90
6,4259	2,6951	4,9364	1,2332	75
5,3128	2,8924	3,6837	0,8858	83,33
5,9774	3,2022	3,9708	1,5748	85
5,5498	3,0135	4,3283	1,7018	85
5,3842	3,2909	4,6179	1,2852	91,67
5,8599	2,8053	4,2641	1,0279	91,67
6,3136	3,2676	4,0476	0,5788	91,67
				Ortalama =87,17

- b. Öğrenme faktörleri  $C1=C2=0,2$ ,  $rand_1=0,2$  ve  $rand_2=0,2$  olduğunda doğruluk oranları Tablo 5.36.'da verildiği gibidir.

**Tablo 5.36.** Parçacık sürü optimizasyonu Adım 1 b. için sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Doğruluk oranı (%)
5,4102	3,1473	4,2226	1,7457	86,67
5,5501	3,1948	4,2893	1,3916	91,67
5,9456	2,9908	4,0255	1,3128	91,67
6,2341	2,6170	4,2325	1,2475	91,67
5,9796	2,9335	4,1985	0,8776	93,33
5,4602	3,3084	3,8241	0,9976	83,33
6,4774	2,4993	4,4518	0,7143	93,33
6,4936	2,4371	4,2384	1,2833	93,33
5,8899	2,9128	3,8338	1,2807	91,67
5,7029	2,9752	4,3182	1,3812	90
				Ortalama = 90,67

- c. Öğrenme faktörleri  $C1=C2=0,2$ ,  $rand_1=0,4$  ve  $rand_2=0,4$  olduğunda doğruluk oranları Tablo 5.37.'de verildiği gibidir.

**Tablo 5.37.** Parçacık sürü optimizasyonu Adım 1 c. için sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Doğruluk oranı (%)
5,8380	2,9932	4,1511	1,4053	90
6,0267	2,8970	4,0551	1,3754	90
6,1468	2,6773	4,3378	1,4329	88,33
5,8938	2,9897	4,1208	1,4651	90
6,1390	2,8044	4,1725	1,3092	91,67
6,0164	2,8587	4,2012	1,3014	88,33
5,4914	3,2260	4,2098	1,1999	91,67
5,4228	3,3183	4,3050	0,5082	75
5,8906	2,9246	4,2150	1,4007	88,33
5,2941	3,0657	4,4719	1,4183	93,33
				Ortalama = 88,67

- d. Öğrenme faktörleri  $C1=C2=0,2$ ,  $rand_1=0,618$  ve  $rand_2=0,618$  olduğunda doğruluk oranları Tablo 5.38.'de verildiği gibidir.

**Tablo 5.38.** Parçacık sürü optimizasyonu Adım 1 d. için sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Doğruluk oranı (%)
5,6284	3,2685	3,9706	1,4870	88,33
5,7955	3,2998	3,4291	1,3401	80
5,8337	2,7658	4,4851	1,4116	90
5,5760	3,2786	4,1924	1,0628	88,33
6,1814	2,6376	4,3089	1,3252	88,33
5,5576	3,1948	4,2850	1,2980	91,67
5,8645	2,8385	4,5154	1,2105	90
6,1915	2,9009	3,9053	1,7975	85
6,2661	2,9338	3,9923	1,2000	91,67
5,7280	3,0180	4,1159	1,3239	90
				Ortalama = 88,33

e. Öğrenme faktörleri  $C1=C2=0,2$ ,  $rand_1=0,85$  ve  $rand_2=0,85$  olduğunda doğruluk oranları Tablo 5.39.'da verildiği gibidir.

**Tablo 5.39.** Parçacık sürü optimizasyonu Adım 1 e. için sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Doğruluk oranı (%)
5,8732	2,9664	4,3010	1,2114	91,67
5,6490	3,0363	4,2152	1,1832	93,33
5,5048	3,3042	4,2653	1,0952	90
5,7881	2,9999	4,2708	1,2728	90
5,4723	3,2361	4,3292	1,1045	91,67
6,0462	2,8642	4,1882	1,3526	88,33
5,6409	3,0016	4,4835	1,3501	90
6,2993	2,8140	4,0885	0,9571	91,67
5,8038	2,9602	4,3360	1,2778	90
6,0090	2,8991	4,1942	1,3958	88,33
				Ortalama = 90,50

**Adım 2:**

a. Öğrenme faktörleri  $C1=C2=0,4$ ,  $rand_1=rand$  ve  $rand_2=rand$  olduğunda doğruluk oranları Tablo 5.40.'da verildiği gibidir.

**Tablo 5.40.** Parçacık sürü optimizasyonu Adım 2 a. için sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Doğruluk oranı (%)
6,0129	2,9777	3,8371	1,5047	91,67
5,7683	2,9823	4,0366	1,5062	90
6,0991	2,8149	4,1316	0,9080	90
5,5149	3,2340	4,0660	0,7568	76,67
5,9969	2,8036	4,1904	1,4243	91,67
5,6575	2,9257	3,9856	1,8235	86,67
6,2375	2,9203	4,1569	1,6209	90
5,7239	2,8749	4,4718	0,9144	93,33
5,8482	3,2580	4,0283	1,4358	91,67
5,2211	3,0123	3,9973	0,9721	83,33
				Ortalama = 88,50

- b. Öğrenme faktörleri  $C1=C2=0,4$ ,  $rand_1=0,2$  ve  $rand_2=0,2$  olduğunda doğruluk oranları Tablo 5.41.'de verildiği gibidir.

**Tablo 5.41 .** Parçacık sürü optimizasyonu Adım 2 b. için sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Doğruluk oranı (%)
5,7865	3,0162	4,0901	1,3749	90
5,6389	3,0963	4,1646	1,3645	91,67
5,5828	3,0657	4,5840	1,3128	88,33
5,8077	2,8666	4,3263	1,4778	90
5,3071	3,0728	4,8622	1,0104	90
5,7917	3,1696	3,5903	0,9176	83,33
5,7215	3,1630	4,1482	0,9822	90
6,0246	2,8206	4,2434	1,2881	88,33
5,6417	3,1031	3,8806	1,6477	88,33
5,8994	2,8695	4,2005	1,4358	88,33
				Ortalama = 88,83

- c. Öğrenme faktörleri  $C1=C2=0,4$ ,  $rand_1=0,4$  ve  $rand_2=0,4$  olduğunda doğruluk oranları Tablo 5.42.'de verildiği gibidir.

**Tablo 5.42.** Parçacık sürü optimizasyonu Adım 2 c. için sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Doğruluk oranı (%)
5,8650	2,7639	4,5514	1,1904	91,67
5,4802	3,0110	4,5244	1,5014	88,33
5,0071	3,4914	4,4473	1,3425	90
5,6693	3,2360	4,1941	1,1302	90
5,9227	2,9106	4,2533	1,2940	90
5,4086	3,1354	4,4264	1,3222	91,67
5,3070	3,4601	4,1971	1,2675	85
5,8588	2,7327	4,3205	1,6116	88,33
5,6980	3,2547	3,9807	0,9864	88,33
5,6174	2,9940	4,2811	1,3992	90
				Ortalama = 89,33

- d. Öğrenme faktörleri  $C1=C2=0,4$ ,  $rand_1=0,618$  ve  $rand_2=0,618$  olduğunda doğruluk oranları Tablo 5.43.'de verildiği gibidir.

**Tablo 5.43.** Parçacık sürü optimizasyonu Adım 2 d. için sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Doğruluk oranı (%)
5,7367	2,9995	4,2692	1,4044	90
5,9318	2,8743	4,3965	1,3477	90
5,7003	3,0552	4,1536	1,2912	91,67
5,7062	3,0008	4,1072	1,5038	90
5,5613	3,1402	4,2407	1,3887	91,67
5,9050	2,9945	4,1240	1,3511	90
6,3159	2,7833	4,1542	1,3343	90
5,7014	3,0094	4,2317	0,9235	90
5,8128	2,9961	4,4253	1,0170	90
5,5016	3,1572	4,3981	1,5198	88,33
				Ortalama = 90,17



- e. Öğrenme faktörleri  $C1=C2=0,4$ ,  $rand_1=0,85$  ve  $rand_2=0,85$  olduğunda doğruluk oranları Tablo 5.44.'de verildiği gibidir.

**Tablo 5.44.** Parçacık sürü optimizasyonu Adım 2 e. için sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Doğruluk oranı (%)
5,9725	2,8746	4,3851	1,2027	88,33
5,5158	3,1763	4,4084	1,1676	93,33
5,4085	3,2764	4,3838	1,2794	90
5,6071	2,9980	4,3119	1,4017	90
6,1181	2,7988	4,3220	1,2572	90
5,7795	3,0090	4,2111	1,0333	91,67
6,0329	2,8959	4,1961	1,2969	88,33
5,4925	3,1827	4,4028	1,2709	91,67
5,1026	3,4101	4,2055	1,4049	86,67
5,7061	3,0976	4,3163	1,1082	91,67
				Ortalama = 90,17

**Adım 3:**

- a. Öğrenme faktörleri  $C1=C2=0,58$ ,  $rand_1=rand$  ve  $rand_2=rand$  olduğunda doğruluk oranları Tablo 5.45.'de verildiği gibidir.

**Tablo 5.45.** Parçacık sürü optimizasyonu Adım 3 a. için sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Doğruluk oranı (%)
5,8661	2,8559	4,3804	1,3421	88,33
5,7857	3,3134	3,9925	1,4805	90
5,6533	2,9960	3,7492	1,0146	86,67
5,5732	2,9460	4,2834	1,2727	91,67
5,7484	2,8663	4,2772	1,2108	91,67
5,8092	2,7947	4,2513	1,3907	90
5,5794	3,2323	4,2940	1,2746	91,67
5,5467	2,9449	4,0909	1,5368	93,33
5,7030	2,9861	4,0723	1,4645	90
5,6223	2,9086	3,8010	1,2467	86,67
				Ortalama = 90

b. Öğrenme faktörleri  $C1=C2=0,58$ ,  $rand_1=0,2$  ve  $rand_2=0,2$  olduğunda doğruluk oranları Tablo 5.46.'da verildiği gibidir.

**Tablo 5.46.** Parçacık sürü optimizasyonu Adım 3 b. için sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Doğruluk oranı (%)
5,8472	2,8418	4,2377	1,3924	90
5,8554	2,7833	4,2930	1,3419	90
5,6661	3,0011	4,1768	1,4364	90
6,0374	2,9218	4,0322	1,3604	91,67
5,4817	3,1649	4,2189	1,3021	91,67
5,9392	3,0085	4,2584	1,5287	86,67
5,8992	2,9991	4,2204	0,5716	86,67
6,0018	2,9562	4,1831	1,4129	88,33
5,7681	3,0511	4,2070	1,3326	91,67
5,6510	3,0873	4,2506	1,1896	93,33
				Ortalama = 90

c. Öğrenme faktörleri  $C1=C2=0,58$ ,  $rand_1=0,4$  ve  $rand_2=0,4$  olduğunda doğruluk oranları Tablo 5.47.'de verildiği gibidir.

**Tablo 5.47.** Parçacık sürü optimizasyonu Adım 3 c. için sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Doğruluk oranı (%)
5,8941	2,8950	4,3732	1,3757	90
5,6006	3,1000	4,2462	1,1927	91,67
5,9582	2,8845	4,3195	1,3933	90
5,5004	3,1493	4,4197	1,2306	91,67
5,6235	3,2007	4,2397	1,1364	90
5,7988	3,0065	4,2376	1,1148	90
6,0950	2,8410	4,0020	1,2942	90
6,0020	2,9997	3,9539	1,1992	91,67
5,5413	3,1875	4,4600	1,1203	91,67
5,3905	3,3153	4,1109	1,2848	90
				Ortalama = 90,67

d. Öğrenme faktörleri  $C1=C2=0,58$ ,  $rand_1=0,618$  ve  $rand_2=0,618$  olduğunda doğruluk oranları Tablo 5.48.'de verildiği gibidir.

**Tablo 5.48.** Parçacık sürü optimizasyonu Adım 3 d. için sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Doğruluk oranı (%)
5,7277	2,9868	4,2337	1,4791	91,67
5,8470	2,9958	4,0902	1,1035	91,67
5,7358	2,9867	4,3696	1,5306	91,67
5,8136	2,9975	4,4058	0,7672	88,33
5,7855	3,0252	4,0649	1,2963	91,67
5,6282	3,1463	4,2124	1,3051	91,67
5,5526	3,0991	4,2375	1,3315	91,67
5,6045	3,0040	4,2159	1,2998	91,67
5,9553	2,8996	4,3523	1,2703	88,33
5,6678	3,0719	4,2429	1,2527	91,67
				Ortalama = 91

e. Öğrenme faktörleri  $C1=C2=0,58$ ,  $rand_1=0,85$  ve  $rand_2=0,85$  olduğunda doğruluk oranları Tablo 5.49.'da verildiği gibidir.

**Tablo 5.49.** Parçacık sürü optimizasyonu Adım 3 e. için sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Doğruluk oranı (%)
5,7988	2,9480	4,3030	1,2953	90
5,6968	3,0844	4,2008	1,1967	93,33
5,7945	3,0090	4,3203	1,1662	90
5,8189	2,9582	4,2957	1,2959	90
5,8040	2,9933	4,2418	1,3886	90
6,0562	2,8065	4,2748	1,2741	91,67
5,7382	2,9999	4,3105	1,4935	88,33
6,0113	2,9987	4,1762	1,1035	91,67
6,1513	2,8858	4,1862	1,1823	90
5,9912	2,9976	4,2326	1,0287	91,67
				Ortalama = 90,67

**Adım 4:**

a. Öğrenme faktörleri  $C1=C2=0,618$ ,  $rand_1=rand$  ve  $rand_2=rand$  olduğunda doğruluk oranları Tablo 5.50.'de verildiği gibidir.

**Tablo 5.50.** Parçacık sürü optimizasyonu Adım 4 a. için sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Doğruluk oranı (%)
5,8377	3,0487	4,2184	1,4529	88,33
5,9412	2,9157	3,6099	1,1305	88,33
5,7996	2,8840	4,3188	1,3182	90
5,7009	2,9684	4,3691	1,3586	90
5,8081	3,0459	4,5621	1,3164	86,67
5,7679	3,1344	4,3381	1,5294	86,67
5,6393	2,7153	4,4782	1,0655	90
5,6250	3,0591	4,3254	1,1767	91,67
5,8399	3,3996	4,1398	1,0774	90
5,8117	3,0996	4,3521	0,9700	91,67
				Ortalama = 89,33

b. Öğrenme faktörleri  $C1=C2=0,618$ ,  $rand_1=0,2$  ve  $rand_2=0,2$  olduğunda doğruluk oranları Tablo 5.51.'de verildiği gibidir.

**Tablo 5.51.** Parçacık sürü optimizasyonu Adım 4 b. için sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Doğruluk oranı (%)
6,2218	2,7515	4,0443	1,3305	91,67
6,0928	2,7395	4,3953	0,9679	90
5,8420	2,9838	4,2327	1,7340	83,33
5,5565	3,2963	4,0541	1,0693	86,67
5,5202	2,9823	4,5884	1,5201	90
5,8405	2,9307	4,2669	1,2974	90
5,1931	3,2791	4,4269	1,2581	91,67
6,0811	2,8392	4,4953	1,0923	90
5,6647	3,3186	3,8730	1,3292	86,67
5,8794	2,7795	4,6179	0,6763	90
				Ortalama = 89

c. Öğrenme faktörleri  $C1=C2=0,618$ ,  $rand_1=0,4$  ve  $rand_2=0,4$  olduğunda doğruluk oranları Tablo 5.52.'de verildiği gibidir.

**Tablo 5.52.** Parçacık sürü optimizasyonu Adım 4 c. için sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Doğruluk oranı (%)
5,4184	3,2978	4,1205	1,2135	90
6,0768	2,8816	4,3608	1,1105	90
5,5141	3,1588	4,2102	1,3005	91,67
5,5885	3,0779	4,3670	1,3176	90
5,6202	3,0972	4,2491	1,2844	91,67
5,8847	2,9692	4,3153	1,3416	88,33
5,5298	3,1646	4,1973	1,4661	90
5,6837	3,0181	4,2666	1,1960	91,67
5,6699	3,0130	4,2805	1,2409	90
5,5860	3,1685	4,2764	1,2520	91,67
				Ortalama = 90,50

d. Öğrenme faktörleri  $C1=C2=0,618$ ,  $rand_1=0,618$  ve  $rand_2=0,618$  olduğunda doğruluk oranları Tablo 5.53.'de verildiği gibidir.

**Tablo 5.53.** Parçacık sürü optimizasyonu Adım 4 d. için sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Doğruluk oranı (%)
5,6178	3,0971	4,2209	1,3437	91,67
5,9017	2,8916	4,3165	1,2987	88,33
5,4386	3,2041	4,2014	1,2918	93,33
5,5086	3,1919	4,3706	1,2946	91,67
5,6488	3,1999	4,1393	1,3000	93,33
5,7862	3,0014	3,9534	1,4768	90
6,0566	3,0266	3,9622	0,9797	91,67
5,6415	3,1877	4,0582	1,3001	90
5,4259	3,2584	4,3316	1,1512	91,67
5,4350	3,1615	4,1793	1,5256	90
				Ortalama = 91,17

e. Öğrenme faktörleri  $C1=C2=0,618$ ,  $rand_1=0,85$  ve  $rand_2=0,85$  olduğunda doğruluk oranları Tablo 5.54.'de verildiği gibidir.

**Tablo 5.54.** Parçacık sürü optimizasyonu Adım 4 e. için sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Doğruluk oranı (%)
5,7432	2,9990	4,2801	1,3217	90
5,7679	2,9935	4,4047	1,4145	88,33
5,6879	3,0091	4,3963	1,3574	90
5,6525	3,0005	4,2108	1,3574	91,67
5,7343	3,0005	4,2739	1,3252	90
5,7907	3,0024	4,2422	1,3005	90
6,0295	2,9095	4,1925	1,3740	88,33
5,7626	3,0045	4,3077	1,2834	90
5,5485	3,1016	4,3826	1,2986	91,67
5,8078	2,9982	4,2540	1,2445	90
				Ortalama = 90

**Adım 5:**

a. Öğrenme faktörleri  $C1=C2=0,64$ ,  $rand_1=rand$  ve  $rand_2=rand$  olduğunda doğruluk oranları Tablo 5.55.'de verildiği gibidir.

**Tablo 5.55.** Parçacık sürü optimizasyonu Adım 5 a. için sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Doğruluk oranı (%)
5,6431	3,0160	4,5422	1,0866	90
5,6839	3,1200	4,6535	1,4767	85
5,6868	2,9134	4,0970	1,2466	91,67
5,3475	3,0030	4,0031	1,2654	88,33
5,5454	3,0472	3,6878	1,2206	86,67
5,9816	3,1033	4,1638	0,9537	95
5,7614	2,8050	4,6310	1,3782	88,33
5,4811	3,2697	4,3020	1,4689	88,33
5,7339	2,8586	4,4607	1,5499	90
5,7310	3,0537	4,2277	1,0176	90
				Ortalama =89,33

- b. Öğrenme faktörleri  $C1=C2=0,64$ ,  $rand_1=0,2$  ve  $rand_2=0,2$  olduğunda doğruluk oranları Tablo 5.56.'da verildiği gibidir.

**Tablo 5.56.** Parçacık sürü optimizasyonu Adım 5 b. için sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Doğruluk oranı (%)
5,3989	3,1330	4,4560	1,2606	91,67
5,5929	3,0423	4,2660	1,4052	90
6,0028	2,9313	4,2080	1,3000	88,33
5,5138	3,1944	4,1946	1,2227	93,33
5,7802	3,1683	4,0339	0,9858	88,33
5,9924	2,9449	4,2268	1,1942	90
6,1774	2,8509	4,1999	1,0685	91,67
6,0530	2,8079	4,3211	1,3089	90
5,9119	2,9199	4,0528	1,3683	90
5,7928	3,0026	4,2633	1,2995	91,67
				Ortalama = 90,50

- c. Öğrenme faktörleri  $C1=C2=0,64$ ,  $rand_1=0,4$  ve  $rand_2=0,4$  olduğunda doğruluk oranları Tablo 5.57.'de verildiği gibidir.

**Tablo 5.57.** Parçacık sürü optimizasyonu Adım 5 c. için sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Doğruluk oranı (%)
6,1254	2,8575	4,3121	1,2149	91,67
5,4690	3,2686	4,2900	1,3288	91,67
5,9036	2,9646	4,3376	1,3114	88,33
6,0154	2,9045	4,2803	1,1510	90
5,9571	2,9120	4,2971	1,2627	88,33
5,9024	2,9292	4,2964	1,2544	90
5,5762	3,0044	4,2342	1,6839	88,33
5,6008	3,1496	4,3488	1,3050	90
5,9146	2,9375	4,4234	1,1961	88,33
5,9108	2,8202	4,3122	1,5057	90
				Ortalama = 89,67

- d. Öğrenme faktörleri  $C1=C2=0,64$ ,  $rand_1=0,618$  ve  $rand_2=0,618$  olduğunda doğruluk oranları Tablo 5.58.'de verildiği gibidir.

**Tablo 5.58.** Parçacık sürü optimizasyonu Adım 5 d. için sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Doğruluk oranı (%)
5,8556	3,0017	4,1593	1,4545	90
5,8189	3,0040	4,2619	1,1238	91,67
5,5580	3,1043	4,2024	1,3958	91,67
6,0343	2,8993	4,2372	1,2674	88,33
5,4868	3,3567	4,1132	1,2293	91,67
5,5901	3,1028	4,2007	1,3563	91,67
5,5940	3,0943	4,2373	1,3032	91,67
5,8121	3,0308	4,1514	1,1138	93,33
5,8818	3,0001	4,1825	1,3088	90
5,5987	3,0021	4,2095	1,2990	91,67
				Ortalama = 91,17

e. Öğrenme faktörleri  $C1=C2=0,64$ ,  $rand_1=0,85$  ve  $rand_2=0,85$  olduğunda doğruluk oranları Tablo 5.59.'da verildiği gibidir.

**Tablo 5.59.** Parçacık sürü optimizasyonu Adım 5 e. için sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Doğruluk oranı (%)
5,7890	3,0123	4,0246	1,2821	93,33
5,7098	3,0001	4,3185	1,2323	90
5,9626	2,8986	4,3161	1,2798	88,33
5,8006	2,9981	4,4318	1,3173	88,33
5,9307	3,0007	4,2870	1,2015	90
5,6361	2,9992	4,3927	1,3305	90
5,8713	3,0094	3,9617	1,3042	91,67
5,9185	2,9095	4,2877	1,2758	90
5,5379	3,1005	4,2130	1,3680	91,67
5,6979	2,9895	4,4240	1,4426	88,33
				Ortalama = 90,17

**Adım 6:**

a. Öğrenme faktörleri  $C1=C2=1$ ,  $rand_1=rand$  ve  $rand_2=rand$  olduğunda doğruluk oranları Tablo 5.60.'da verildiği gibidir.

**Tablo 5.60.** Parçacık sürü optimizasyonu Adım 6 a. için sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Doğruluk oranı (%)
5,5832	2,8149	4,4400	1,0860	91,67
5,9278	2,8820	4,4563	1,3360	90
5,8870	2,9309	4,5788	1,4268	86,67
5,7676	2,8483	4,2049	1,3570	90
5,9474	3,1441	4,4742	1,2495	86,67
5,7187	2,8083	4,3749	1,3052	90
5,8416	3,1532	4,0481	1,1287	91,67
5,6664	2,8023	4,3043	1,4923	90
5,6010	2,8443	3,8953	1,1386	85
5,7453	2,9791	4,3325	1,3763	90
				Ortalama = 89,17

- b. Öğrenme faktörleri  $C1=C2=1$ ,  $rand_1=0,2$  ve  $rand_2=0,2$  olduğunda doğruluk oranları Tablo 5.61.'de verildiği gibidir.

**Tablo 5.61.** Parçacık sürü optimizasyonu Adım 6 b. için sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Doğruluk oranı (%)
5,9373	2,8452	4,2086	1,3899	88,33
6,0052	2,8945	4,1892	1,2979	90
5,8968	2,9180	4,2969	1,3013	90
5,8959	2,9813	4,2767	1,3143	90
5,5822	3,1862	4,2903	1,2850	91,67
5,5930	3,3053	3,9746	1,1141	83,33
6,0806	2,8964	4,3721	1,0803	90
6,0756	2,7498	4,3026	1,4056	88,33
5,8252	2,9916	4,1530	1,3571	90
5,8134	2,9910	4,2870	1,2144	90
				Ortalama = 89,17

- c. Öğrenme faktörleri  $C1=C2=1$ ,  $rand_1=0,4$  ve  $rand_2=0,4$  olduğunda doğruluk oranları Tablo 5.62.'de verildiği gibidir.

**Tablo 5.62.** Parçacık sürü optimizasyonu Adım 6 c. için sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Doğruluk oranı (%)
5,7961	3,0113	4,1572	1,1979	90
5,8213	2,9593	4,2621	1,4231	88,33
5,9721	2,9393	4,2885	1,1469	90
6,0409	2,8064	4,3106	1,3952	90
5,9060	2,8050	4,3139	1,4167	91,67
5,7220	3,0109	4,3217	1,3006	90
5,6955	3,0515	4,0108	1,2193	91,67
6,0022	2,9945	4,0231	1,0753	91,67
6,0272	2,7940	4,3125	1,2446	90
5,7117	3,0083	4,2004	1,4513	90
				Ortalama = 90,33

- d. Öğrenme faktörleri  $C1=C2=1$ ,  $rand_1=0,618$  ve  $rand_2=0,618$  olduğunda doğruluk oranları Tablo 5.63.'de verildiği gibidir.

**Tablo 5.63.** Parçacık sürü optimizasyonu Adım 6 d. için sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Doğruluk oranı (%)
5,7037	3,0001	4,0330	1,5603	90
5,9824	2,9967	4,2978	1,0620	91,67
5,8087	2,9928	4,2062	1,3487	90
5,8501	2,9985	4,3255	1,2978	90
5,8397	2,9892	4,2438	1,2981	90
5,7704	3,0050	4,2311	1,3615	90
5,7847	2,9959	4,2618	1,3021	90
5,4757	3,2036	4,2188	1,2984	91,67
5,6355	3,0937	4,2065	1,2000	91,67
5,9659	2,9039	4,2184	1,3012	90
				Ortalama = 90,50



e. Öğrenme faktörleri  $C1=C2=1$ ,  $rand_1=0,85$  ve  $rand_2=0,85$  olduğunda doğruluk oranları Tablo 5.64.'de verildiği gibidir.

**Tablo 5.64.** Parçacık sürü optimizasyonu Adım 6 e. için sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Doğruluk oranı (%)
5,7985	3,0001	4,2544	1,3061	90
5,7278	3,0008	4,3999	1,2995	90
5,8075	2,9995	4,2575	1,2218	90
5,7914	3,0013	4,2328	1,2317	90
5,8003	3,0030	4,0969	1,3150	91,67
5,7647	2,9996	4,3187	1,4941	88,33
5,8000	2,9997	4,3540	1,2876	90
5,8538	3,0066	4,1961	1,1985	91,67
5,8200	2,9980	4,2497	1,3040	90
5,6386	2,9999	4,3611	1,3761	90
				Ortalama = 90,17

**Adım 7:**

a. Öğrenme faktörleri  $C1=C2=1,4$ ,  $rand_1=rand$  ve  $rand_2=rand$  olduğunda doğruluk oranları Tablo 5.65.'de verildiği gibidir.

**Tablo 5.65.** Parçacık sürü optimizasyonu Adım 7 a. için sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Doğruluk oranı (%)
5,6968	2,9039	3,9982	1,4357	93,33
5,9526	3,0759	4,4710	1,1969	86,67
5,8203	2,9592	4,2487	1,4550	88,33
5,8678	3,0307	4,2320	1,4733	88,33
5,8193	3,0177	4,1203	1,3817	90
5,5693	3,0324	3,9707	1,3841	91,67
5,6338	2,8370	4,0657	1,3296	91,67
5,9669	3,0454	4,1473	1,2248	91,67
5,8668	3,2249	4,3138	1,3071	88,33
5,7694	3,0269	4,4010	1,5549	85
				Ortalama = 89,50

b. Öğrenme faktörleri  $C1=C2=1,4$  ,  $rand_1=0,2$  ve  $rand_2=0,2$  olduğunda doğruluk oranları Tablo 5.66.'da verildiği gibidir.

**Tablo 5.66.** Parçacık sürü optimizasyonu Adım 7 b. için sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Doğruluk oranı (%)
5,9631	2,9907	4,3178	1,1138	91,67
5,7542	3,0012	4,3479	1,4330	88,33
5,9142	2,9106	4,2672	1,2094	91,67
5,5208	3,1323	4,1978	1,3281	91,67
5,7651	3,1551	4,0025	0,8736	86,67
5,9864	2,8719	4,2791	1,2329	90
5,9230	2,9971	4,3801	0,8383	91,67
6,0857	2,8012	4,2791	1,3936	88,33
5,9852	2,8906	4,1886	1,3736	88,33
5,7817	3,0037	4,1913	1,4637	90
				Ortalama = 89,83

c. Öğrenme faktörleri  $C1=C2=1,4$  ,  $rand_1=0,4$  ve  $rand_2=0,4$  olduğunda doğruluk oranları Tablo 5.67.'de verildiği gibidir.

**Tablo 5.67.** Parçacık sürü optimizasyonu Adım 7 c. için sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Doğruluk oranı (%)
5,8370	2,9897	4,1927	1,3701	90
5,8808	2,9322	4,2983	1,3164	90
5,6165	3,0384	4,1899	1,3468	90
5,7124	3,0411	4,2078	1,2094	90
5,7288	3,0101	4,2020	1,3423	90
5,9051	2,9518	4,2979	1,2862	90
5,8552	3,0048	4,0741	1,3116	93,33
5,7062	3,0076	4,3312	1,2212	90
5,5859	3,0075	4,2001	1,4714	90
5,7981	3,0076	4,3051	1,2613	90
				Ortalama = 90,33

d. Öğrenme faktörleri  $C1=C2=1,4$  ,  $rand_1=0,618$  ve  $rand_2=0,618$  olduğunda doğruluk oranları Tablo 5.68.'de verildiği gibidir.

**Tablo 5.68.** Parçacık sürü optimizasyonu Adım 7 d. için sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Doğruluk oranı (%)
5,7447	3,0011	4,2603	1,2090	91,67
5,7353	3,0001	4,2116	1,2999	91,67
5,9243	2,9951	4,3968	1,1296	90
5,9024	2,9999	4,2870	1,1996	91,67
5,8075	3,0000	4,2841	1,1502	91,67
5,6056	3,0000	4,3995	1,3025	90
5,8710	3,0002	4,0997	1,4909	90
5,7166	3,0000	4,2621	1,3797	90
5,7731	2,9995	4,2270	1,3065	90
5,8133	2,9983	4,2080	1,4235	90
				Ortalama = 90,67

e. Öğrenme faktörleri  $C1=C2=1,4$  ,  $rand_1=0,85$  ve  $rand_2=0,85$  olduğunda doğruluk oranları Tablo 5.69.'da verildiği gibidir.

**Tablo 5.69.** Parçacık sürü optimizasyonu Adım 7 e. için sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Doğruluk oranı (%)
5,9782	3,1380	4,3811	1,2812	86,67
5,5369	3,0117	4,4013	1,1388	93,33
5,8975	2,9512	4,3995	1,2800	88,33
5,7931	3,0485	4,1683	1,2814	90
5,7757	2,9903	3,9694	1,2642	90
6,0615	2,9421	4,0663	1,3000	91,67
5,8725	2,8085	4,2763	1,3350	90
5,3813	2,8114	4,5006	1,1066	91,67
5,7445	3,4154	4,2912	1,2738	90
5,6688	3,1285	4,1923	1,2594	91,67
				Ortalama = 90,33

**Adım 8:**

a. Öğrenme faktörleri  $C1=C2=1,618$ ,  $rand_1=rand$  ve  $rand_2=rand$  olduğunda doğruluk oranları Tablo 5.70.'de verildiği gibidir.

**Tablo 5.70.** Parçacık sürü optimizasyonu Adım 8 a. için sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Doğruluk oranı (%)
5,6506	2,9493	4,3687	1,2846	90
5,8280	3,1024	4,1834	1,2685	90
5,7696	2,9588	4,1760	1,1971	90
5,9140	2,9887	4,2084	1,4911	90
5,9215	3,0422	4,5171	1,2433	86,67
5,8468	2,8297	4,3893	1,2923	88,33
5,8402	2,9850	4,1251	1,5174	90
5,9919	2,9321	4,0569	1,2870	90
5,9407	2,9428	4,4772	1,3106	88,33
5,9373	3,0683	4,2500	1,3606	88,33
				Ortalama = 89,17

b. Öğrenme faktörleri  $C1=C2=1,618$ ,  $rand_1=0,2$  ve  $rand_2=0,2$  olduğunda doğruluk oranları Tablo 5.71.'de verildiği gibidir.

**Tablo 5.71.** Parçacık sürü optimizasyonu Adım 8 b. için sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Doğruluk oranı (%)
6,0034	2,9238	4,1680	1,3050	90
5,8637	2,9825	4,2191	1,3366	90
5,7953	3,0032	4,1956	1,2591	90
5,9001	2,9672	4,1943	1,3032	90
5,6381	3,0092	4,3711	1,2985	90
6,0230	2,9326	4,4549	1,0888	90
5,7142	2,9999	4,3917	1,2364	90
5,9238	2,9929	4,1815	1,0618	91,67
5,8934	2,9590	4,2719	1,2793	90
5,5811	3,1273	4,1782	1,3148	93,33
				Ortalama = 90,50

c. Öğrenme faktörleri  $C1=C2=1,618$ ,  $rand_1=0,4$  ve  $rand_2=0,4$  olduğunda doğruluk oranları Tablo 5.72.'de verildiği gibidir.

**Tablo 5.72.** Parçacık sürü optimizasyonu Adım 8 c. için sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Doğruluk oranı (%)
5,8865	2,9979	4,2017	1,3866	90
5,8039	2,9721	4,2535	1,3985	90
5,7586	3,0137	4,1880	1,2094	90
5,6170	2,9973	4,3781	1,4332	90
5,8062	2,9664	4,2053	1,3437	90
5,8757	2,9718	4,3049	1,3133	90
5,8072	2,8844	4,3587	1,4500	90
5,6075	3,1008	4,2804	1,3400	91,67
5,7790	3,0063	4,2030	1,3105	90
5,8340	2,9988	4,3719	1,2809	90
				Ortalama = 90,17

d. Öğrenme faktörleri  $C1=C2=1,618$ ,  $rand_1=0,618$  ve  $rand_2=0,618$  olduğunda doğruluk oranları Tablo 5.73.'de verildiği gibidir.

**Tablo 5.73.** Parçacık sürü optimizasyonu Adım 8 d. için sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Doğruluk oranı (%)
6,1691	3,2672	3,6734	0,9998	88,33
5,6807	2,9970	4,5001	1,3368	88,33
6,1046	3,0000	4,0741	1,1189	91,67
5,7084	3,3868	4,4006	0,6474	80
5,8392	2,9053	4,9809	0,6086	83,33
5,7002	2,9998	4,4981	1,2122	90
5,6010	3,3313	4,3995	1,0707	88,33
6,0284	2,8283	4,8529	0,7003	88,33
5,8390	2,9009	4,6251	1,2608	90
5,4343	2,9905	4,5807	1,7991	81,67
				Ortalama = 87

- e. Öğrenme faktörleri  $C1=C2=1,618$ ,  $rand_1=0,85$  ve  $rand_2=0,85$  olduğunda doğruluk oranlar Tablo 5.74.'de verildiği gibidir.

**Tablo 5.74.** Parçacık sürü optimizasyonu Adım 8 e. için sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Doğruluk oranı (%)
5,8227	2,9063	4,0944	1,2480	91,67
5,7988	2,9929	4,1759	1,2693	90
5,8797	3,0047	4,3598	1,2930	88,33
5,7978	3,0005	4,2221	1,3650	90
5,7007	3,0121	4,2004	1,2623	90
5,7428	2,9997	4,0721	1,4410	90
5,5012	2,9052	4,2930	1,3617	90
5,7396	3,0048	4,1585	1,2688	90
5,6717	3,0450	4,2936	1,2996	90
5,7543	3,0084	4,0888	1,2285	93,33
				Ortalama = 90,33

**Adım 9:**

- a. Öğrenme faktörleri  $C1=C2=2$ ,  $rand_1=rand$  ve  $rand_2=rand$  olduğunda doğruluk oranları Tablo 5.75.'de verildiği gibidir.

**Tablo 5.75.** Parçacık sürü optimizasyonu Adım 9 a. için sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Doğruluk oranı (%)
5,7852	3,0963	4,0464	1,4583	88,33
5,8745	2,9828	4,5157	1,3310	86,67
5,8191	2,9790	3,7642	1,1355	86,67
5,8618	2,9685	4,3055	1,4809	90
5,6018	3,1143	4,3252	1,1155	93,33
5,8108	3,0957	4,2560	1,3449	90
5,7948	3,0734	4,1374	1,3901	91,67
5,6966	3,0388	4,3376	1,2560	90
5,4753	3,0427	4,4106	1,3823	90
5,4870	2,9685	4,3042	1,3848	93,33
				Ortalama = 90

- b. Öğrenme faktörleri  $C1=C2=2$ ,  $rand_1=0,2$  ve  $rand_2=0,2$  olduğunda doğruluk oranları Tablo 5.76.'da verildiği gibidir.

**Tablo 5.76.** Parçacık sürü optimizasyonu Adım 9 b. için sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Doğruluk oranı (%)
5,7961	3,0113	4,1572	1,1979	90
5,8213	2,9593	4,2621	1,4231	88,33
5,7921	2,9393	4,2885	1,1469	91,67
6,0409	2,8064	4,3106	1,3952	90
5,9060	2,8050	4,3139	1,4167	91,67
5,7220	3,0109	4,3217	1,3006	90
5,6955	3,0515	4,0108	1,2193	91,67
6,0022	2,9945	4,0231	1,0753	91,67
6,0272	2,7940	4,3125	1,2446	90
5,7117	3,0083	4,2004	1,4513	90
				Ortalama = 90,50

- c. Öğrenme faktörleri  $C1=C2=2$ ,  $rand_1=0,4$  ve  $rand_2=0,4$  olduğunda doğruluk oranları Tablo 5.77.'de verildiği gibidir.

**Tablo 5.77.** Parçacık sürü optimizasyonu Adım 9 c. için sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Doğruluk oranı (%)
5,7914	2,9999	4,4034	1,3443	88,33
5,6471	3,1272	4,3485	1,1413	91,67
5,7013	3,0103	4,2996	1,1466	91,67
5,7996	2,8776	4,4350	1,2794	88,33
5,8002	2,9954	4,3848	1,3225	88,33
5,8835	3,0000	4,1970	1,2856	90
5,7897	2,9996	4,1982	1,5130	88,33
5,7207	2,9998	4,2963	1,2041	91,67
6,1088	2,9521	4,0793	1,1799	91,67
5,7720	3,0006	4,1155	1,4794	90
				Ortalama = 90

- d. Öğrenme faktörleri  $C1=C2=2$ ,  $rand_1=0,618$  ve  $rand_2=0,618$  olduğunda doğruluk oranları Tablo 5.78.'de verildiği gibidir.

**Tablo 5.78.** Parçacık sürü optimizasyonu Adım 9 d. için sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Doğruluk oranı (%)
5,6809	3,1040	4,3639	1,0695	91,67
5,8310	3,0818	4,3229	1,2957	90
5,9016	2,9895	4,3933	1,2435	88,33
5,7300	3,0020	4,2874	1,3831	90
5,6995	2,8708	4,0013	1,3544	93,33
5,7922	2,9999	4,2058	1,5020	88,33
5,5797	3,0035	4,2003	1,4122	90
5,6107	3,0114	4,3755	1,2811	90
5,8017	2,9996	4,2881	1,2572	90
5,7532	3,0058	4,2830	1,3344	90
				Ortalama = 90,17

e. Öğrenme faktörleri  $C1=C2=2$ ,  $rand_1=0,85$  ve  $rand_2=0,85$  olduğunda doğruluk oranları Tablo 5.79.'da verildiği gibidir.

**Tablo 5.79.** Parçacık sürü optimizasyonu Adım 9 e. için sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Doğruluk oranı (%)
5,7033	2,9631	4,4050	1,1938	90
6,1129	2,7581	4,5053	0,9548	90
5,8604	2,9992	4,0022	1,2376	90
5,8558	3,0004	4,0969	1,3232	90
5,6246	2,9521	4,3905	2,2401	90
5,7872	3,0086	4,3498	1,3000	90
5,7914	3,0008	4,1775	1,2993	90
5,7662	2,9996	4,2477	1,2980	90
5,8014	3,0023	4,3757	1,3294	88,33
5,8076	2,9965	4,2141	1,3714	90
				Ortalama = 89,83

**Adım 10:**

a. Öğrenme faktörleri  $C1=C2=3$ ,  $rand_1=rand$  ve  $rand_2=rand$  olduğunda doğruluk oranları Tablo 5.80.'de verildiği gibidir.

**Tablo 5.80.** Parçacık sürü optimizasyonu Adım 10 a. için sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Doğruluk oranı (%)
5,5452	3,0914	4,1985	1,3681	91,67
5,8377	3,0063	4,0355	1,3473	91,67
5,6914	3,1005	4,3580	1,5632	85
5,7256	2,9908	4,0487	1,1822	91,67
5,8163	2,9729	4,4294	1,3741	90
5,8725	2,8487	4,3648	1,3088	88,33
5,6280	3,0078	4,0216	1,1461	88,33
5,8173	2,9534	4,2135	0,9732	91,67
5,7676	3,1145	4,5542	1,4170	86,67
5,3242	2,9761	4,2172	1,4060	91,67
				Ortalama = 89,67

- b. Öğrenme faktörleri  $C1=C2=3$ ,  $rand_1=0,2$  ve  $rand_2=0,2$  olduğunda doğruluk oranları Tablo 5.81.'de verildiği gibidir.

**Tablo 5.81.** Parçacık sürü optimizasyonu Adım 10 b. için sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Doğruluk oranı (%)
5,8486	2,9666	4,2013	1,3308	90
5,8304	2,9998	4,0524	1,4076	90
5,6554	3,0006	4,3138	1,3537	90
5,9917	2,7981	4,3481	1,4047	90
5,6936	3,0435	4,3126	1,2011	90
5,6888	3,0015	4,3212	1,4091	90
5,8137	2,9887	4,3259	1,3013	90
5,7094	3,0076	4,2255	1,1907	90
5,6000	3,0029	4,2126	1,3596	90
5,7641	3,0049	4,2018	1,2256	90
				Ortalama = 90

- c. Öğrenme faktörleri  $C1=C2=3$ ,  $rand_1=0,4$  ve  $rand_2=0,4$  olduğunda doğruluk oranları Tablo 5.82.'de verildiği gibidir.

**Tablo 5.82.** Parçacık sürü optimizasyonu Adım 10 c. için sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Doğruluk oranı (%)
5,8628	2,9027	4,0325	1,2808	91,67
5,6782	2,9931	4,3021	1,3292	90
5,6173	3,0313	3,9993	1,2369	90
5,7790	2,9418	4,3718	1,2136	90
5,9936	3,1061	4,2253	1,1293	93,33
5,7069	3,1652	4,5340	1,2988	86,67
6,1941	2,9951	4,2166	1,2764	88,33
5,7997	3,0251	4,3324	1,2892	90
5,7390	2,9787	4,1828	1,2295	91,67
5,7380	3,0522	4,3016	1,2282	91,67
				Ortalama = 90,33

- d. Öğrenme faktörleri  $C1=C2=3$ ,  $rand_1=0,618$  ve  $rand_2=0,618$  olduğunda doğruluk oranları Tablo 5.83.'de verildiği gibidir.

**Tablo 5.83.** Parçacık sürü optimizasyonu Adım 10 d. için sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Doğruluk oranı (%)
5,8881	3,1072	4,3545	1,5460	85
5,8296	2,9990	4,2006	0,9827	91,67
6,1102	3,0889	3,9933	1,0264	93,33
5,6949	2,9920	4,2547	1,4114	90
5,7692	3,1073	4,2537	1,3707	90
5,9758	2,9970	4,3646	1,6999	86,67
6,0931	3,0697	4,5876	1,1894	88,33
5,4928	2,8877	3,8667	1,4803	90
5,7739	3,1000	3,6715	1,4098	91,67
5,9602	3,1854	4,2774	0,8798	93,33
				Ortalama = 90



e. Öğrenme faktörleri  $C1=C2=3$ ,  $rand_1=0,85$  ve  $rand_2=0,85$  olduğunda doğruluk oranları Tablo 5.84.'de verildiği gibidir.

**Tablo 5.84.** Parçacık sürü optimizasyonu Adım 10 e. için sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Doğruluk oranı (%)
6,2301	3,1749	3,2344	1,6976	86,67
5,0676	3,1176	3,6560	0,8941	85
4,9433	2,4920	4,4601	1,1159	85
6,0592	3,6194	4,1776	0,9481	88,33
5,6780	3,6151	3,1829	1,4805	73,33
5,6912	3,1202	3,2297	1,1629	76,67
6,3407	3,4096	3,6325	1,0885	90
5,5347	3,3877	2,9165	0,7308	73,33
6,0638	3,5910	3,7202	1,3044	86,67
6,2113	2,9547	3,7549	2,1650	73,33
				Ortalama = 81,83

**Adım 11:**

a. Öğrenme faktörleri  $C1=C2=4$ ,  $rand_1=rand$  ve  $rand_2=rand$  olduğunda doğruluk oranları Tablo 5.85.'de verildiği gibidir.

**Tablo 5.85.** Parçacık sürü optimizasyonu Adım 11 a. için sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Doğruluk oranı (%)
5,8027	3,1104	4,0978	1,5134	88,33
5,8568	2,7976	3,9862	1,2256	91,67
5,6198	3,1714	4,2532	1,5179	88,33
5,8663	3,0949	4,0910	1,3135	91,67
5,4912	2,9119	3,7191	1,1823	86,67
5,3839	2,9779	4,3935	1,0175	88,33
5,7082	3,1389	4,3177	1,4818	88,33
5,8161	2,8620	4,1689	0,9676	90
5,8905	3,1378	4,0940	1,1656	93,33
5,9646	3,0407	4,5932	1,3190	86,67
				Ortalama = 89,33

- b. Öğrenme faktörleri  $C1=C2=4$ ,  $rand_1=0,2$  ve  $rand_2=0,2$  olduğunda doğruluk oranları Tablo 5.86.'da verildiği gibidir.

**Tablo 5.86.** Parçacık sürü optimizasyonu Adım 11 b. için sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Doğruluk oranı (%)
5,8754	2,8905	4,2252	1,4202	88,33
5,6066	3,0216	4,1019	1,4297	91,67
5,6116	2,9998	4,3141	1,5053	90
5,8823	3,0000	3,9998	1,2208	91,67
5,6486	3,0856	4,2931	1,2591	91,67
5,8811	3,0004	4,1408	1,1684	91,67
5,6143	3,1018	4,4380	0,4481	81,67
6,0327	2,9954	4,2138	1,2384	93,33
5,5470	3,0056	41997	1,4101	90
5,7203	3,0000	4,2101	1,4570	90
				Ortalama = 90

- c. Öğrenme faktörleri  $C1=C2=4$ ,  $rand_1=0,4$  ve  $rand_2=0,4$  olduğunda doğruluk oranları Tablo 5.87.'de verildiği gibidir.

**Tablo 5.87.** Parçacık sürü optimizasyonu Adım 11 c. için sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Doğruluk oranı (%)
5,7114	2,8306	4,1444	1,5172	90
5,7938	3,0011	4,2309	1,2520	90
5,7650	2,9997	4,3183	1,2111	90
5,8673	3,0014	4,3521	1,2863	88,33
5,4446	2,9570	4,4230	1,2999	91,67
5,6525	3,0000	4,1626	1,2171	90
5,7395	3,1418	4,1999	1,2538	91,67
5,7740	3,0024	4,2845	1,2514	90
5,7679	2,8758	4,1661	1,2047	91,67
5,8007	2,8443	4,2590	1,2571	90
				Ortalama = 90,33

- d. Öğrenme faktörleri  $C1=C2=4$ ,  $rand_1=0,618$  ve  $rand_2=0,618$  olduğunda doğruluk oranları Tablo 5.88.'de verildiği gibidir.

**Tablo 5.88.** Parçacık sürü optimizasyonu Adım 11 d. için sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Doğruluk oranı (%)
5,5084	2,5728	3,3938	0,9869	83,33
6,0646	3,4227	4,4339	1,1793	86,67
6,0189	4,0524	4,2559	1,2388	81,67
5,3948	3,2532	4,0639	1,5802	88,33
6,0720	3,8676	4,1476	1,5296	81,67
5,9819	3,1954	3,2548	1,7454	85
5,5980	3,1624	4,0517	2,4579	73,33
6,1372	2,8163	3,0959	1,1778	88,33
5,9040	3,0992	3,5861	0,7712	81,67
5,8802	2,5846	4,1048	1,0384	90
				Ortalama = 84

e. Öğrenme faktörleri  $C1=C2=4$ ,  $rand_1=0,85$  ve  $rand_2=0,85$  olduğunda doğruluk oranları Tablo 5.89.'da verildiği gibidir.

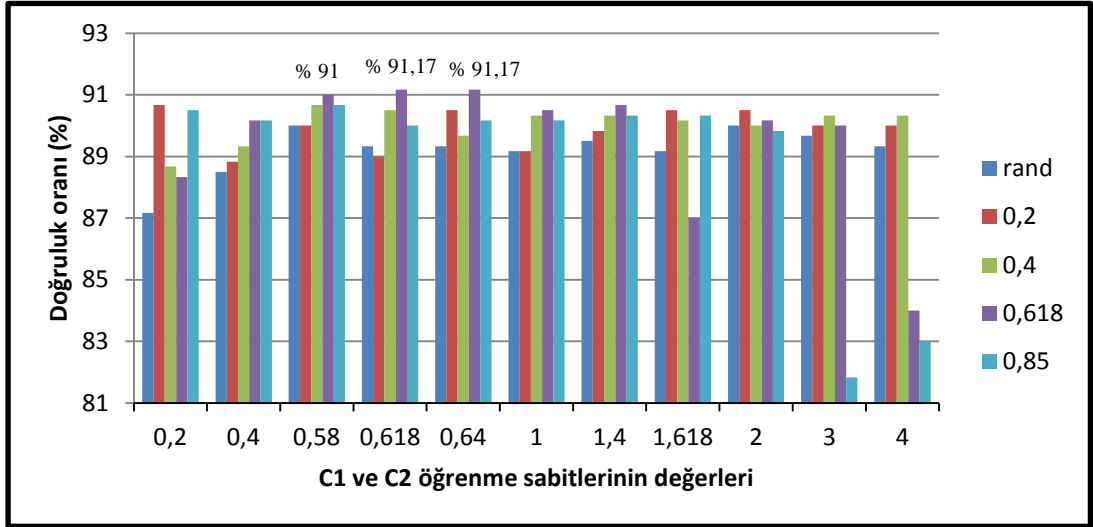
**Tablo 5.89.** Parçacık sürü optimizasyonu Adım 11 e. için sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Doğruluk oranı (%)
6,1382	2,7943	3,7745	1,4152	90
5,3908	3,1406	4,1321	1,4068	91,67
6,4589	2,7014	4,3102	1,9654	75
5,2475	2,8010	3,6176	0,4511	83,33
6,5027	2,8305	4,1203	2,1887	73,33
5,6355	2,3029	3,7300	1,3272	86,67
5,3122	3,2133	3,6599	0,9339	86,67
5,5003	3,3685	4,3371	1,7947	76,67
5,5355	2,6407	3,6022	1,7208	86,67
6,2677	3,2099	4,4046	1,7777	80
				Ortalama = 83

Tablo 5.90. buraya kadar olan ilk 11 adımın tüm durumlarının doğruluk oranlarını göstermektedir. Sütun değerleri  $c1=c2$  değerlerini satır değerleri ise  $rand_1$  ve  $rand_2$  değerlerini göstermektedir.

**Tablo 5.90.** Parçacık Sürü Optimizasyon algoritması uygulama yazılımı ilk 11 adımın tüm durumlarının doğruluk tabloları

rand <sub>1</sub> rand <sub>2</sub> için tüm durumlar ↓	←C1=C2 için tüm durumlar→										
	0,2	0,4	0,58	0,618	0,64	1	1,4	1,618	2	3	4
rand	87,17	88,50	90	89,33	89,33	89,17	89,5	89,17	90	89,67	89,33
0,2	90,67	88,83	90	89	90,50	89,17	89,83	90,50	90,50	90	90
0,4	88,67	89,33	90,67	90,50	89,67	90,33	90,33	90,17	90	90,33	90,33
0,618	88,33	90,17	91	91,17	91,17	90,50	90,67	87	90,17	90	84
0,85	90,50	90,17	90,67	90	90,17	90,17	90,33	90,33	89,83	81,83	83



Şekil 5.3. Parçacık sürü optimizasyonu algoritması uygulama yazılımı, ilk 11 adımın tüm durumlarının doğruluk oranlarının grafik temsili

Şekil 5.3. dikkatlice incelenecek olursa en yüksek doğruluk oranları C1 ve C2 sabitlerinin [0,1] kapalı aralığındaki 0.58, 0.618 ve 0.64 değerlerini aldığı yerlerde ve  $rand_1$  ve  $rand_2$  değerlerinin 0.618 olduğu durumlarda gerçekleşmiştir. Bu da göstermektedir ki; yine en yüksek sonuçlar altın oran değeri etrafında gerçekleşmektedir. C1 ve C2 değerleri, yöntemin öğrenme sabitleridir ve örnek verilerin dağılım uzayının boyutunu direkt etkilemektedir.  $rand_1$  ve  $rand_2$  değerleri ise pbest ve gbest değerlerinin etkilerini yeni çözüme taşıyan sabitlerdir. Bunların altın oran etrafında en iyi doğruluk oranlarını vermesi yeni çözümleri üretmede altın oranın etkisinin iyi olduğunu göstermektedir.

Literatürde Parçacık sürü optimizasyonu algoritması için C1 ve C2 değişkenlerinin optimal değerlerinin genelde 2 değeri civarında olduğu ifade edilmektedir. Fakat bu çalışmamız göstermektedir ki bu her çalışma için geçerli olmamaktadır. Bu çalışmada C1 ve C2 değerlerinin en iyi sonuçlarını [0,1] aralığında ve bu aralıkta da altın oran değerine yakın olan noktalarda verdiği görülmüştür.

**Adım 12:** Bu adımda fidan gelişim algoritmasının başlangıç popülasyonu oluşturma tekniği uygulanarak ilk 11 adımda elde edilen bazı sonuçlarla karşılaştırması yapılmıştır.

a. Öğrenme faktörleri  $C1=C2=0,618$ ,  $rand_1=0,618$  ve  $rand_2=0,618$  olduğunda doğruluk oranları Tablo 5.91.'de verildiği gibidir.

**Tablo 5.91.** Parçacık sürü optimizasyonu Adım 12 a. için sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Doğruluk oranı (%)
5,7388	3,0028	4,2207	1,3315	90
5,7910	3,0005	4,3216	1,3405	90
5,8027	3,0005	4,1697	1,4302	90
5,8307	2,9945	4,2487	1,3092	90
5,6794	3,0420	4,3951	1,0004	93,33
5,7802	3,0001	4,3426	1,3470	90
5,7220	3,0005	4,2820	1,2001	91,67
5,7187	3,0015	4,2886	1,3914	90
5,8085	3,0221	4,2036	0,9913	91,67
5,9323	2,9910	3,9719	1,2515	91,67
				Ortalama = 90,83

b. Öğrenme faktörleri  $C1=C2=3$ ,  $rand_1=0,85$  ve  $rand_2=0,85$  olduğunda doğruluk oranları Tablo 5.92.'de verildiği gibidir.

**Tablo 5.92.** Parçacık sürü optimizasyonu Adım 12 b. için sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Doğruluk oranı (%)
5,7405	2,8262	4,8689	0,9262	90
6,0416	3,1673	5,1515	0,4514	73,33
5,7344	3,8025	3,7803	2,1491	71,67
4,8723	2,4040	3,3985	0,1000	83,33
5,3923	2,7710	4,1091	0,9634	88,33
5,1237	3,3549	3,9734	1,6123	80
6,0374	2,6165	4,8690	1,3971	88,33
5,6336	2,9413	4,5859	1,0413	90
5,4714	2,8269	4,8673	0,9269	90
5,0817	3,3033	4,8859	1,1769	88,33
				Ortalama = 84,33

c. Öğrenme faktörleri  $C1=C2=0,64$ ,  $rand_1=0,618$  ve  $rand_2=0,618$  olduğunda doğruluk oranları Tablo 5.2.93.'de verildiği gibidir.

**Tablo 5.93.** Parçacık sürü optimizasyonu Adım 12 c. için sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Doğruluk oranı (%)
5,7653	3,0002	4,1915	1,3510	90
5,9074	2,9975	4,2637	1,1226	90
5,8807	2,9965	4,2932	1,2160	90
5,7876	3,0040	4,1076	1,2941	90
5,7602	3,0074	4,1733	1,4603	90
5,6768	3,0024	4,2478	1,3115	91,67
5,7164	2,9998	4,3921	1,1039	91,67
5,8053	2,9978	4,2330	1,3074	90
5,7074	3,0055	4,1972	1,5583	90
5,8954	2,9265	4,3188	1,3016	88,33
				Ortalama = 90,17

d. Öğrenme faktörleri  $C1=C2=4$ ,  $rand_1=0,85$  ve  $rand_2=0,85$  olduğunda doğruluk oranları Tablo 5.94.'de verildiği gibidir.

**Tablo 5.94.** Parçacık sürü optimizasyonu Adım 12 d. için sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Doğruluk oranı (%)
5,2994	2,7055	5,1657	0,8055	88,33
5,2567	3,4338	3,4201	0,5573	85
5,2415	2,6646	5,2662	0,7646	83,33
5,6069	3,1443	1,8234	1,7676	83,33
5,2367	3,5655	4,4292	1,6567	83,33
6,7212	2,5496	4,0832	1,5538	91,67
5,3376	2,7324	5,0995	0,8324	83,33
5,3781	2,7610	5,0291	0,8610	86,67
5,2396	2,3460	3,5924	0,8128	88,33
5,8406	3,0875	4,2266	1,4125	88,33
				Ortalama = 85,67

e. Öğrenme faktörleri  $C1=C2=1$ ,  $rand_1=0,4$  ve  $rand_2=0,4$  olduğunda doğruluk oranları Tablo 5.95.'de verildiği gibidir.

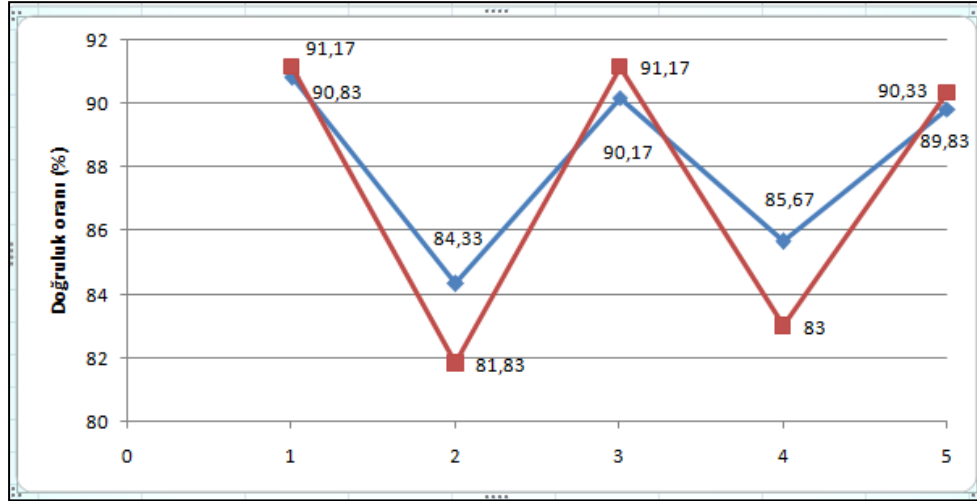
**Tablo 5.95.** Parçacık sürü optimizasyonu Adım 12 e. için sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Doğruluk oranı (%)
5,7156	2,9992	4,2986	1,1627	90
5,7204	3,0001	4,3769	1,0326	91,67
5,7367	2,9990	4,3076	1,4710	88,33
5,7563	2,9257	4,3655	1,4891	90
5,7660	3,0002	4,2018	1,3293	90
5,7296	3,0091	4,0813	1,3198	90
5,7180	3,0009	4,3304	1,1895	91,67
5,7165	2,9999	4,5140	0,9781	91,67
5,7385	2,9607	4,2692	1,6854	86,67
5,8118	2,9996	4,2192	1,4956	88,33
				Ortalama = 89,83

**Tablo 5.96.** Parçacık sürü optimizasyon algoritması için fidan gelişim algoritması öncesi ve sonrası durum mukayesesi

	Fidan gelişimden algoritmasından önceki doğruluk oranı (%)	Fidan gelişim algoritması ile doğruluk oranı (%)
$C1=C2=0,618,$ $rand_1=rand_2=0,618$	91,17	90,83
$C1=C2=3,$ $rand_1=rand_2=0,85$	81,83	84,33
$C1=C2=0,64,$ $rand_1=rand_2=0,618$	91,17	90,17
$C1=C2=4,$ $rand_1=rand_2=0,85$	83	85,67
$C1=C2=1,$ $rand_1=rand_2=0,4$	90,33	89,83

Tablo 5.96.'da, ilk 11 adımda elde edilen doğruluk oranlarının bazı durumlardaki değerlerinin, Fidan gelişim algoritmasının başlangıç popülasyonunu oluşturma tekniğinden faydalanılarak elde edilmiş halleri ile karşılaştırması sunulmuştur.



**Şekil 5.4.** Parçacık sürü optimizasyon algoritmasında fidan gelişim algoritmasının etkilerinin grafik temsili

Şekil 5.4. dikkatlice incelenecek olursa son 5 adımda, C1 ve C2 değerlerinin [0,1] aralığına yakın olduğu yerlerde fidan gelişim algoritmasını uygulamak doğruluk oranını daha iyi bir yere taşımamıştır. Çünkü Parçacık sürü optimizasyon algoritmasının yapısında hız ve buna bağlı olarak çözüm değerleri oluşturulurken maksimum ve minimum değerlere bağlı kalmaktadır. ( Denklem (2.1), (2.2), (2.3) ve (2.4) )

Ayrıca uygulama yazılımında; hız ve çözüm (pozisyon = x) değerlerinin başlangıç değerleri oluşturulurken eğitim veri setindeki maksimum ve minimum değerleri aşmayacak şekilde popülasyon oluşturulduğundan bir nevi fidan gelişim algoritmasına benzer bir yaklaşım denenmiştir. Dolayısıyla fidan gelişim algoritmasının başlangıç popülasyonu ile yapılan denemeler, zaten kararlı olan [0,1] aralığında daha iyi sonuçlar vermemiştir. Fakat [0,1] aralığından uzak olan yerlerde fidan gelişim algoritmasının sonuçları, doğruluk oranını daha iyi değerlere taşımıştır.

### 5.3. Ateş Böceği Algoritması Uygulama Yazılımı ve Sonuçları

Tez çalışmasının bu bölümünde Wisconsin Diagnosis Breast Cancer Database ile çalışılmıştır [74].

Veri tabanından alınan veriler Tablo 5.97.'deki formattadır.

**Tablo 5.97.** Wisconsin Diagnosis Breast Cancer veri tabanından alınan ham bilgilerin temsili

1000025	5	1	1	1	2	1	3	1	1	2
1002945	5	4	4	5	7	10	3	2	1	2
1015425	3	1	1	1	2	2	3	1	1	2
1016277	6	8	8	1	3	4	3	7	1	2
1017023	4	1	1	3	2	1	3	1	1	2
1017122	2	1	2	1	2	1	3	1	1	4
1018099	4	2	1	1	2	1	2	1	1	2

Yukarıdaki tabloda ilk sütun kod numaralarını, son sütun ise tümörün iyi veya kötü huylu olduğunu belirlemektedir. “2” değerine sahip olanlar “benign” yani iyi huylu kitle sınıfını; “4” değerine sahip olanlar ise “malignant” yani kötü huylu kitle sınıfını temsil etmektedir. Veri tabanında bu şekilde 700 adet örnek mevcuttur. Fakat elde edilen bilgilerin bazılarının içerisinde “ ? “ şeklinde eksik bilgiler bulunmaktadır. Bu kısımlar çıkarılarak çalışılmıştır. Bu şekilde olan veriler çıkarıldıktan sonra 100 adet örnek test için ayrılmıştır. Toplamda eğitim amaçlı 577 örnek ile çalışılmıştır. Kod numaraları kısmı, örneklere verilen bir numara olup, çözüm yöntemi ile alakası olmadığından çözüm yöntemi için sunulan tabloda işin içinden çıkarılmıştır. Son sütun olan sınıf özelliği de, sınıflandırma (classification) değil de , kümeleme (clustering) yapılacağı için yazılımın tahmini ile belirlenmeye çalışılacaktır. Bu nedenle çalışılacak olan veri aşağıdaki hale gelmiştir.



Sütunlar sırasıyla, kitle kalınlığı, hücre boyutunun birbirine benzerliği, hücre şeklinin birbirine benzerliği, marjinal yapışkanlık, tek epitelyal hücre boyutu, saf çekirdek, donuk kromatin ,normal çekirdek, mitoz bölünmelerdir.

**Tablo 5.98.** Ateş böceği algoritması uygulama yazılımında işlenecek olan verinin yeni hali

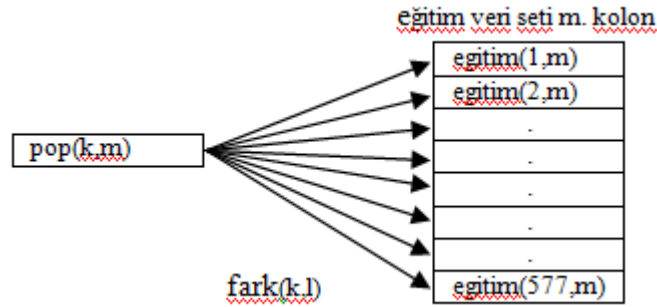
5	1	1	1	2	1	3	1	1
5	4	4	5	7	10	3	2	1
3	1	1	1	2	2	3	1	1
6	8	8	1	3	4	3	7	1
4	1	1	3	2	1	3	1	1
2	1	2	1	2	1	3	1	1
4	2	1	1	2	1	2	1	1

Çözüm için 20 elemanlı bir ateşböceği popülasyonu oluşturulmuştur. Aşağıdaki işlemler ile eğitim kümesi ile popülasyonun her değerinin tek tek farkları alınır. Fark matrisi oluşturulur. Burada uzaklık hesaplamaları manhattan uzaklık formülüne göre yapılmıştır.

$$\|x_i - x_j\| = \sum_{k=1}^d |(x_{i,k} - x_{j,k})| \quad (5.23)$$

```

for k=1,2,...20
  for m=1,2,...9
    fark(k,m)=sum(|pop(k,m)-eğitim(:,m)|)
  
```



**Şekil 5.5.** Popülasyon değerleri ile eğitim verilerinin farklarının temsili

Bu işlem adımı ile de, farkların toplamı alınır ve farktop matrisine atılır. Böylece popülasyondaki her 20 ateş böceğinin de eğitim kümesi ile farklarının toplamı hesaplanmış olur.

```

for k=1,2,...20
  farktop(1,k)=sum(fark(k,:))

```

**Tablo 5.99.** Ateş böceği algoritması uygulama yazılımı farklar matrisinin temsili

fark(1,1)	fark (1,2)	.	.	.	fark (1,9)
fark (2,1)	fark (2,2)	.	.	.	fark (2,9)
.	.	.	.	.	.
.	.	.	.	.	.
fark (20,1)	fark (20,2)	.	.	.	fark(20,9)

$\sum (\text{fark}(k,:))$

Her çözüm için bir farklar toplamı hesaplanacak olursa toplam 20 tane farklar toplamı ortaya çıkar.

**Tablo 5.100.** Ateş böceği algoritması uygulama yazılımı farklar matrisinin toplamının temsili

farktop(1,1)	farktop(1,2)	.	.	.	farktop(1,20)
--------------	--------------	---	---	---	---------------

r değeri, öklit uzaklığı olarak hesaplanmakta olup, ateş böceklerinin birbirine olan uzaklığı olarak kabul edilmektedir. Aşağıdaki adımdaki işlemde tüm ateş böceklerinin birbirine olan uzaklıkları, r matrisine atılmaktadır.

```
for k=1,2,...,20
  for m=1,2,...,20
    r(k,m)=sqrt((farktop(1,k)-farktop(1,m)).^2)
```

Sıradaki adımda ise tüm ateş böceklerinin, öteki ateş böceklerine olan uzaklıkları toplamı hesaplanır. rtop matrisine atılır. Işık şiddeti ve çekicilik hesaplamaları rtop matrisinin değerleri üzerinden yapılmıştır.

```
for k=1,2,...,20
  rtop(1,k)=sum(r(k,:))
```

i<sub>1</sub> matrisi ışık şiddetlerinin matrisidir. Hesaplama 1000000 sayısına bölme işlemi normalizasyonu sağlamak için yapılmıştır. Çünkü ateş böceği algoritmasının teorisinde, exponansiyel olarak değişen üslü ifade kısmı çok büyük değerlerde sıfır değerini vermektedir. Bu nedenle belirli bir aralıkta normalizasyon yapmak için bu işleme tabi tutulmuştur.

```
for k=1,2,...,20
  i(1,k)=i0*(exp(-gamma*rtop(1,k)/1000000))
```

Çekicilik değerleri de hesaplanırken aynı şekilde normalizasyon işlemi gerçekleştirilmiştir.

```
for k=1,2,...20
beta(1,k)=beta0*(exp(-gamma*((rtop(1,k)/1000000).^2)))
```

Daha sonra yeni çözüm değerleri aşağıdaki şekilde hesaplanır.

```
for k=1:20
    pop(k,:)= (abs(10-((pop(b,:)+rand*beta(1,b)+alpha*rand))));
end
```

Toplamda 20 adet ateş böceğini temsilen 20 tane matris ile çalışılmıştır. Her ateş böceğini temsil eden matrisin 9 sütunu vardır. Çünkü üzerinde çalışılan veri setinde 9 farklı özellik mevcuttur. Uygulama yazılımımız, sonuçta BEST olarak isimlendirilen ve 1X9 boyutunda bir matrisi çıktı olarak vermektedir. Bu matrisin her elemanı test kümemizin sütun elemanlarının her birindeki elemanlarla teker teker karşılaştırılıp, bir toplam hata değeri elde edilmektedir. Amaç bu hata değerini minimize etmektir. BEST matrisinin her bir elemanı, nümerik optimum çözüm değerlerini vermektedir.

**Tablo 5.101.** Ateş böceği algoritması uygulama yazılımı BEST matris temsili

BEST(1,1)	BEST(1,2)	BEST(1,3)	...	BEST(1,8)	BEST(1,9)
-----------	-----------	-----------	-----	-----------	-----------

Uygulama yazılımında parametrelerin etkilerini görmek üzere denenmiş olan 23 adım sırasıyla bu bölümün sonuna kadar verilmiştir.

**Adım 1:** gamma=1, beta<sub>0</sub>=1, alpha=1 ve i<sub>0</sub>=1, başlangıç popülasyonu rastgele olursa sonuçlar Tablo 5.102.'de olduğu gibidir.

**Tablo 5.102.** Ateş böceği algoritması Adım 1 için sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	BEST (1,5)	BEST (1,6)	BEST (1,7)	BEST (1,8)	BEST (1,9)	Doğruluk oranı(%)
8	7	7	8	7	8	7	8	7	95
1	2	1	1	2	1	2	2	1	93
8	8	8	8	8	8	8	8	8	95
1	2	3	1	3	2	1	3	1	91
3	2	2	2	2	2	3	3	2	91
6	6	8	8	6	8	6	7	6	94
1	1	1	1	3	1	3	1	3	94
1	3	3	1	3	1	3	3	1	94
8	8	8	9	9	8	8	9	9	94
6	6	6	8	6	8	6	8	6	93
									Ortalama= 93,4

**Adım 2:**  $\gamma=0,618$ ,  $\beta_0=1$ ,  $\alpha=1$  ve  $i_0=1$ , başlangıç popülasyonu rastgele olursa sonuçlar Tablo 5.103.'de olduğu gibidir.

**Tablo 5.103.** Ateş böceği algoritması Adım 2 için sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	BEST (1,5)	BEST (1,6)	BEST (1,7)	BEST (1,8)	BEST (1,9)	Doğruluk oranı (%)
8	8	8	8	8	8	8	8	8	95
3	1	1	1	3	3	3	3	1	92
2	2	2	2	2	2	1	1	1	93
7	7	7	8	7	7	8	8	8	95
7	7	7	8	9	9	7	7	9	95
2	2	1	1	1	2	2	2	1	93
9	8	9	8	8	9	8	9	9	95
2	2	2	2	2	2	2	2	2	92
6	7	6	7	6	7	7	7	6	95
7	7	8	7	7	8	8	8	7	95
									Ortalama=94

**Adım 3:**  $\gamma=0,2$ ,  $\beta_0=1$ ,  $\alpha=1$  ve  $i_0=1$ , başlangıç popülasyonu rastgele olursa sonuçlar Tablo 5.104.'de olduğu gibidir.

**Tablo 5.104.** Ateş böceği algoritması Adım 3 için sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	BEST (1,5)	BEST (1,6)	BEST (1,7)	BEST (1,8)	BEST (1,9)	Doğruluk oranı (%)
7	8	7	7	8	9	9	9	9	95
9	9	9	9	9	9	9	9	9	95
3	3	3	2	3	2	2	3	2	85
3	2	1	3	1	2	3	2	2	91
3	3	3	2	3	3	2	2	2	85
1	3	3	2	3	2	3	1	1	89
4	4	2	3	3	2	3	4	2	85
7	8	8	7	7	7	7	8	7	95
1	3	3	1	3	1	3	1	1	95
6	9	8	9	7	9	6	6	8	95
									Ortalama =91

**Adım 4:**  $\gamma=0,95$ ,  $\beta_0=1$ ,  $\alpha=1$  ve  $i_0=1$ , başlangıç popülasyonu rastgele olursa sonuçlar Tablo 5.105.'de olduğu gibidir.

**Tablo 5.105.** Ateş böceği algoritması Adım 4 için sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	BEST (1,5)	BEST (1,6)	BEST (1,7)	BEST (1,8)	BEST (1,9)	Doğruluk oranı (%)
7	7	7	8	9	9	7	7	9	95
4	2	2	2	3	3	4	3	2	88
2	2	1	1	1	2	2	2	1	93
1	3	1	3	3	2	3	1	1	93
2	2	2	2	2	2	2	2	2	92
6	7	6	7	6	7	7	7	6	94
7	7	8	7	7	8	8	8	7	95
7	8	7	7	8	9	9	9	9	95
3	3	3	2	3	2	2	3	2	89
3	2	2	3	2	2	3	2	2	91
									Ortalama =92,5

**Adım 5:**  $\gamma=1,618$ ,  $\beta_0=1$ ,  $\alpha=1$  ve  $i_0=1$ , başlangıç popülasyonu rastgele olursa sonuçlar Tablo 5.106.'da olduğu gibidir.

**Tablo 5.106.** Ateş böceği algoritması Adım 5 için sonuçlar

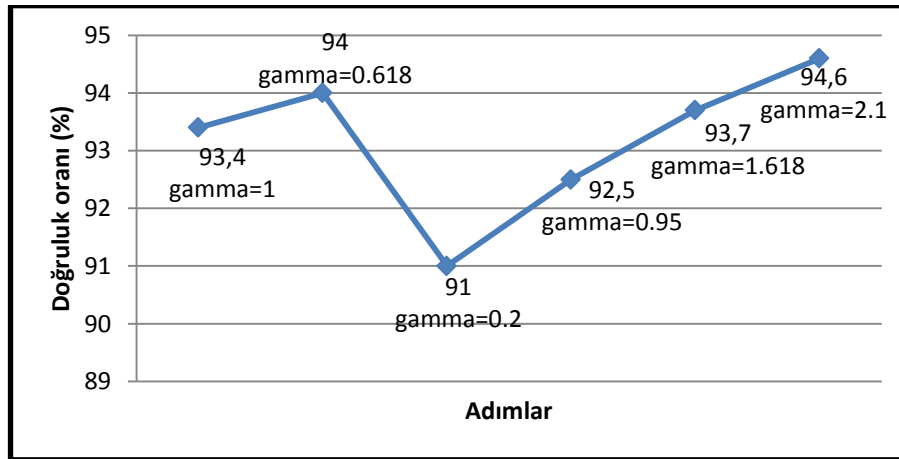
BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	BEST (1,5)	BEST (1,6)	BEST (1,7)	BEST (1,8)	BEST (1,9)	Doğruluk oranı (%)
2	1	1	1	2	1	1	2	2	92
9	9	9	7	9	9	7	9	7	95
6	8	7	6	7	6	6	6	7	95
3	3	1	1	1	1	3	3	1	93
4	2	2	2	4	2	4	4	2	90
7	7	8	8	7	8	7	7	8	96
2	2	1	2	2	1	1	1	2	93
1	1	1	1	1	1	1	1	1	95
1	3	3	3	3	1	3	1	3	92
9	7	7	8	7	8	8	7	8	96
									Ortalama =93,7

**Adım 6:**  $\gamma=2,1$ ,  $\beta_0=1$ ,  $\alpha=1$  ve  $i_0=1$ , başlangıç popülasyonu rastgele olursa sonuçlar Tablo 5.107.'de oda olduğu gibidir.

**Tablo 5.107.** Ateş böceği algoritması Adım 6 için sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	BEST (1,5)	BEST (1,6)	BEST (1,7)	BEST (1,8)	BEST (1,9)	Doğruluk oranı (%)
7	8	7	8	7	7	7	7	8	96
6	8	7	6	7	6	6	6	7	96
8	8	9	9	9	9	8	8	9	94
7	8	8	8	7	8	7	7	7	94
2	2	1	1	2	1	2	2	1	93
1	1	1	1	1	1	1	1	1	95
2	2	2	2	2	2	2	2	2	93
8	7	7	8	7	8	8	7	8	96
7	8	9	9	7	7	9	8	8	94
6	8	8	8	6	6	6	6	8	95
									Ortalama =94,6

Ateş böceği algoritmasında  $\gamma$  değeri tasarımcıya bağlı olarak  $[0,\infty)$  aralığında değişebilir. Bu nedenle 6. adımda  $\gamma=2,1$  değerinde, ortalama en yüksek değere ulaşabilmektedir. Adım 1, 2, 3, 4, 5 ve 6 dikkatlice incelendiğinde  $[0,1]$  kapalı aralığında en yüksek ortalamaya altın oran değerinin ondalık kısmı olan 0,618 değerinde ulaşmıştır. Bu değerden uzaklaşıldığında başarı oranında anında düşme gözükmektedir. Bu durum şekil 5.6.'da görülmektedir.



**Şekil 5.6.** Ateş böceği algoritması İlk 6 adımın ortalama doğruluk oranlarının grafiği

**Adım 7:**  $\gamma=0,618$ ,  $\beta_0=0,618$ ,  $\alpha=1$  ve  $i_0=1$ , başlangıç popülasyonu rastgele olursa sonuçlar Tablo 5.108.'de olduğu gibidir.

**Tablo 5.108.** Ateş böceği algoritması Adım 7 için sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	BEST (1,5)	BEST (1,6)	BEST (1,7)	BEST (1,8)	BEST (1,9)	Doğruluk oranı (%)
1	1	1	2	2	1	2	1	2	95
7	7	7	7	7	7	7	7	7	95
2	1	2	1	2	2	2	1	1	94
2	2	2	2	2	2	2	2	2	92
3	1	1	1	3	1	3	3	2	93
3	3	1	1	3	1	3	3	1	94
9	9	9	9	9	9	9	9	9	95
3	1	1	1	3	2	3	2	1	94
2	2	2	1	1	1	2	2	1	93
1	1	1	1	1	1	1	1	1	95
									Ortalama = 94

**Adım 8:**  $\gamma=0,618$ ,  $\beta_0=0,2$ ,  $\alpha=1$  ve  $i_0=1$ , başlangıç popülasyonu rastgele olursa sonuçlar Tablo 5.109.'da olduğu gibidir.

**Tablo 5.109.** Ateş böceği algoritması Adım 8 için sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	BEST (1,5)	BEST (1,6)	BEST (1,7)	BEST (1,8)	BEST (1,9)	Doğruluk oranı (%)
2	1	1	2	2	2	1	2	1	94
7	8	7	8	7	8	8	8	7	94
3	2	1	3	2	1	1	1	2	94
8	8	8	8	8	8	8	8	8	95
2	1	2	2	2	1	2	1	2	93
2	2	2	1	2	1	1	2	1	92
3	3	2	3	3	2	1	1	1	93
1	3	1	2	3	2	3	1	3	94
2	2	2	2	2	2	2	2	2	92
3	1	2	1	3	1	3	3	2	93
									Ortalama =93,4

**Adım 9:**  $\gamma=0,618$ ,  $\beta_0=1,618$ ,  $\alpha=1$  ve  $i_0=1$ , başlangıç popülasyonu rastgele olursa sonuçlar Tablo 5.110.'da olduğu gibidir.

**Tablo 5.110.** Ateş böceği algoritması Adım 9 için sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	BEST (1,5)	BEST (1,6)	BEST (1,7)	BEST (1,8)	BEST (1,9)	Doğruluk oranı (%)
2	1	2	1	1	2	1	2	2	93
9	8	9	7	9	9	7	7	9	94
2	1	2	2	1	2	2	2	2	91
3	1	2	1	1	3	3	3	1	92
3	1	1	3	3	1	3	1	1	96
2	2	2	3	4	3	2	2	2	90
1	1	1	1	1	1	1	1	1	95
3	2	3	1	3	1	3	3	1	91
6	8	8	8	6	6	6	8	6	94
2	1	2	1	1	1	2	1	2	94
									Ortalama =93

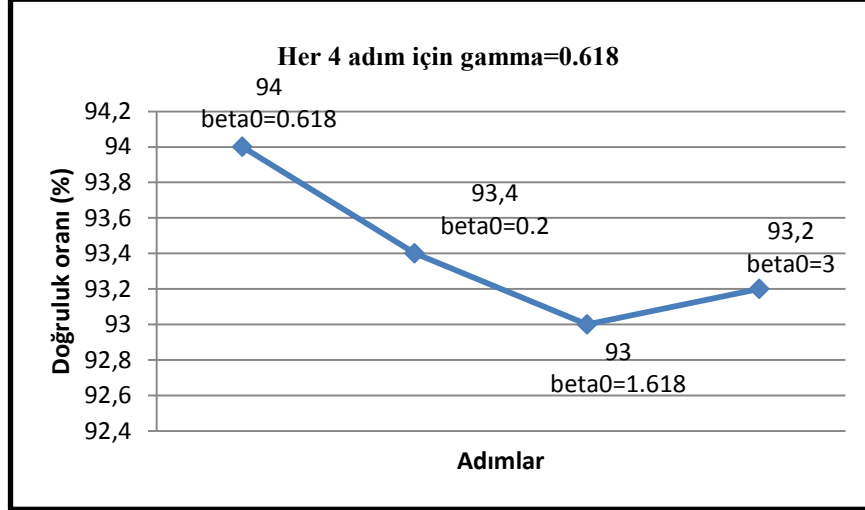
**Adım 10:**  $\gamma=0,618$ ,  $\beta_0=3$ ,  $\alpha=1$  ve  $i_0=1$ , başlangıç popülasyonu rastgele olursa sonuçlar Tablo 5.111.'de olduğu gibidir.

**Tablo 5.111.** Ateş böceği algoritması Adım 10 için sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	BEST (1,5)	BEST (1,6)	BEST (1,7)	BEST (1,8)	BEST (1,9)	Doğruluk oranı (%)
1	1	2	1	1	1	1	3	1	93
8	8	8	8	8	8	8	8	8	95
2	3	2	3	1	3	3	2	2	90
1	1	2	1	3	2	1	1	1	95
2	2	1	1	2	2	1	1	2	93
2	3	2	2	3	3	3	2	2	90
3	2	1	1	3	1	3	3	1	95
8	8	6	8	5	8	7	5	7	95
2	3	3	2	1	2	1	2	1	91
1	1	1	1	1	1	1	1	1	95
									Ortalama =93,2

Adım 7, 8, 9 ve 10 dikkatlice incelendiğinde başarı oranının en yüksek, 7. adımda  $\beta_0$  parametresinin 0,618 olduğu değerde olduğu görülecektir. Şekil 5.7. bu durumu göstermektedir.





Şekil 5.7. Ateş böceği algoritması Adım 7, 8, 9 ve 10 için ortalama doğruluk oranlarının grafiği

**Adım 11:** gamma=0,618, beta<sub>0</sub>=0,618, alpha=0,618 ve i<sub>0</sub>=1, başlangıç popülasyonu rastgele olursa sonuçlar Tablo 5.112.'de olduğu gibidir.

**Tablo 5.112.** Ateş böceği algoritması Adım 11 için sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	BEST (1,5)	BEST (1,6)	BEST (1,7)	BEST (1,8)	BEST (1,9)	Doğruluk oranı (%)
8	8	8	8	8	8	8	8	8	95
4	2	3	1	4	1	1	1	1	93
3	1	1	2	3	3	1	1	1	95
2	2	2	1	2	1	3	1	1	94
2	3	3	1	3	3	1	2	1	92
3	1	1	3	3	3	1	1	3	94
1	4	1	2	4	1	3	3	2	94
3	1	2	1	1	3	2	1	1	94
2	3	1	1	3	1	3	1	3	95
1	1	1	1	2	1	2	1	2	95
Ortalama =94,1									

**Adım 12:**  $\gamma=0,618$ ,  $\beta_0=0,618$ ,  $\alpha=0,25$  ve  $i_0=1$ ; başlangıç popülasyonu rastgele olursa sonuçlar Tablo 5.113.'de olduğu gibidir.

**Tablo 5.113.** Ateş böceği algoritması Adım 12 için sonuçlar

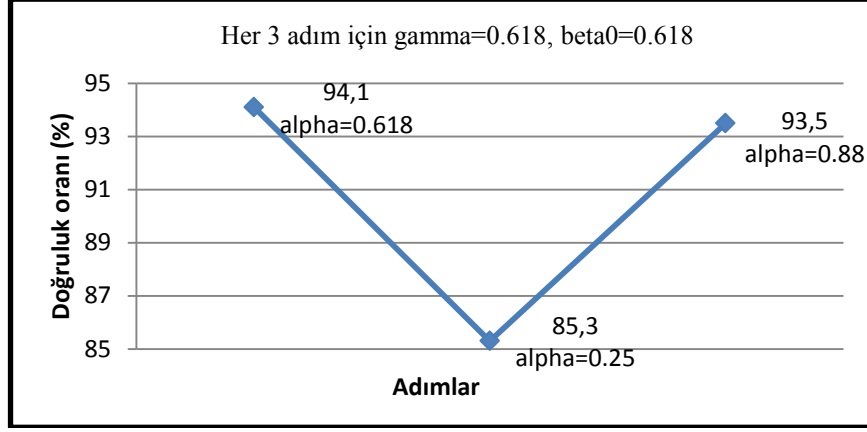
BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	BEST (1,5)	BEST (1,6)	BEST (1,7)	BEST (1,8)	BEST (1,9)	Doğruluk oranı (%)
1	2	6	1	8	1	7	2	9	90
9	8	8	2	6	8	3	4	5	92
6	9	6	5	1	7	3	5	6	88
5	7	8	4	4	4	3	5	7	80
2	8	4	6	9	7	6	1	1	82
1	5	7	7	4	8	1	9	5	84
1	2	7	6	2	8	9	8	4	84
1	3	5	3	9	7	8	5	7	83
3	4	1	7	8	8	7	7	4	84
9	8	3	2	3	7	6	6	7	86
									Ortalama =85,3

**Adım 13:**  $\gamma=0,618$ ,  $\beta_0=0,618$ ,  $\alpha=0,88$  ve  $i_0=1$ , başlangıç popülasyonu rastgele olursa sonuçlar Tablo 5.114.'de loda olduğu gibidir.

**Tablo 5.114.** Ateş böceği algoritması Adım 13 için sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	BEST (1,5)	BEST (1,6)	BEST (1,7)	BEST (1,8)	BEST (1,9)	Doğruluk oranı (%)
2	1	1	1	2	1	2	2	1	92
2	1	1	2	2	1	2	1	2	94
7	7	7	7	7	7	7	8	8	95
1	1	1	1	2	1	2	1	1	95
8	8	8	8	8	8	8	8	8	95
4	1	3	3	1	2	4	2	3	90
1	1	1	1	3	3	1	1	1	95
6	8	8	8	6	8	6	8	6	94
2	2	2	3	3	3	2	2	2	91
3	1	1	1	2	1	1	3	1	94
									Ortalama =93,5

Adım 11,12 ve 13 dikkatlice incelendiğinde başarı oranının en yüksek, 11. adımda  $\alpha$  parametresinin 0,618 olduğu değerde olduğu görülecektir. Şekil 5.8. bu durumu göstermektedir.



Şekil 5.8. Ateş böceği algoritması Adım 11, 12 ve 13 için ortalama doğruluk oranlarının grafiği

**Adım 14:**  $\gamma=0,618$ ,  $\beta_0=0,618$ ,  $\alpha=0,618$  ve  $i_0=0,618$ , başlangıç popülasyonu rastgele olursa sonuçlar Tablo 5.115.'de olduğu gibidir.

**Tablo 5.115.** Ateş böceği algoritması Adım 14 için sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	BEST (1,5)	BEST (1,6)	BEST (1,7)	BEST (1,8)	BEST (1,9)	Doğruluk oranı (%)
1	2	2	1	2	1	1	2	1	93
8	8	8	8	8	8	8	8	8	95
3	1	1	3	1	1	3	3	1	92
1	1	1	1	1	1	1	1	1	95
1	1	1	1	2	2	2	1	1	93
3	3	1	1	3	3	1	1	3	94
2	1	1	2	2	1	1	2	1	94
7	7	7	7	7	7	7	7	7	96
1	1	1	3	3	3	1	1	1	94
1	1	1	1	2	1	2	2	2	94
									Ortalama =94

**Adım 15:**  $\gamma=0,618$ ,  $\beta_0=0,618$ ,  $\alpha=0,618$  ve  $i_0=0,2$ , başlangıç popülasyonu rastgele olursa sonuçlar Tablo 5.116.'da olduğu gibidir.

**Tablo 5.116.** Ateş böceği algoritması Adım 15 için sonuçlar

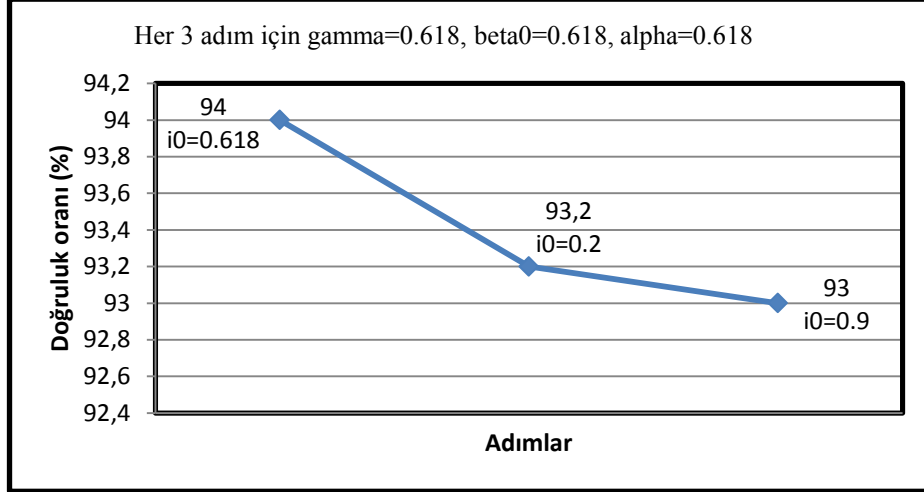
BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	BEST (1,5)	BEST (1,6)	BEST (1,7)	BEST (1,8)	BEST (1,9)	Doğruluk oranı (%)
6	8	7	8	6	8	8	6	6	94
3	2	2	3	2	3	3	3	2	90
3	3	1	1	1	1	3	3	1	93
3	1	1	1	3	1	3	3	1	95
2	2	1	1	2	1	2	2	1	94
3	3	2	2	3	2	2	2	2	91
8	8	8	8	8	8	8	8	8	95
3	3	1	3	3	3	2	3	1	91
3	2	1	1	3	3	1	2	2	94
5	1	1	1	4	2	3	2	1	95
									Ortalama =93,2

**Adım 16:**  $\gamma=0,618$ ,  $\beta_0=0,618$ ,  $\alpha=0,618$  ve  $i_0=0,9$ , başlangıç popülasyonu rastgele olursa sonuçlar Tablo 5.117.'de olduğu gibidir.

**Tablo 5.117.** Ateş böceği algoritması Adım 16 için sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	BEST (1,5)	BEST (1,6)	BEST (1,7)	BEST (1,8)	BEST (1,9)	Doğruluk oranı (%)
1	3	1	2	3	3	1	1	1	94
2	2	2	2	2	2	2	2	2	92
3	3	1	2	1	3	3	3	1	91
4	1	1	2	3	3	4	1	3	94
1	2	2	2	2	2	2	2	1	93
7	9	9	7	7	7	9	7	9	96
8	8	8	8	8	8	8	8	8	95
4	3	1	4	3	1	1	1	3	93
3	3	3	3	3	1	1	1	1	91
3	2	1	1	3	3	3	3	2	91
									Ortalama =93

Adım 14, 15 ve 16 dikkatlice incelendiğinde başarı oranının en yüksek, 14. adımda  $i_0$  parametresinin 0,618 olduğu değerde olduğu görülecektir. Bu durum şekil 5.9.'da gösterilmektedir.



Şekil 5.9. Ateş böceği algoritması Adım 14,15 ve 16 için ortalama doğruluk oranlarının grafiği

Buradan sonraki adımlarda başlangıç popülasyonunun oluşturulmasında, Fidan gelişim algoritmasının başlangıç popülasyonunu oluşturmadaki mantığı uygulanmıştır.

Bu uygulamada veri tabanında 9 özellik bulunduğundan fidan gelişim algoritması ile başlangıç popülasyonu oluşturulduğunda  $2^9 = 512$  adet çözüm oluşacaktır. Bunu aşabilmek için, özellikleri gruplama yöntemine gidilebilir. Bu çalışmada, 9 özellik 2-2-2-3 şeklinde gruplandırılmıştır. Şekil 5.10. bu gruplandırmayı temsil etmektedir.

$L_1+$	$L_1+$	$L_1+$	$L_1+$	$L_1+$	$L_1+$	$L_1+$	$L_1+$	$L_1+$
$(u_1-l_1)*r$	$(u_1-l_1)*r$	$(u_1-l_1)*(1-r)$	$(u_1-l_1)*(1-r)$	$(u_1-l_1)*r$	$(u_1-l_1)*r$	$(u_1-l_1)*r$	$(u_1-l_1)*r$	$(u_1-l_1)*r$
Grup 1		Grup 2		Grup 3		Grup 4		

Şekil 5.10. Fidan gelişim algoritması gruplanmış başlangıç popülasyonu temsili

Her grup kendi içinde  $r$  veya  $(1-r)$  ile çarpılıp sonuç elde edilecek şekilde uygulanmıştır. Bu şekilde 20 adet ateş böceği için 20 satır 9 sütunluk bir başlangıç popülasyonu oluşturma yoluna gidilmiştir.

Burada  $L$  değerleri eğitim değerleri içerisindeki sütunlardaki minimum değerlere,  $u$  değerleri ise eğitim değerleri içerisindeki maksimum değerlere sahip olan matris satırlarını (çözüm değerlerini) temsil etmektedir. Oluşturulan popülasyon maksimum ve minimum değerleri kullanarak yapıldığı için sonuca yakınsama daha iyi olabilmektedir. Bu mantıkla oluşturulan başlangıç popülasyonu yardımıyla yeni matrisler elde edilmiştir. Algoritmanın elde ettiği sonuçlar aşağıdaki adımlarda sunulmuştur. 17. adımdan son adıma kadar elde edilen sonuçlarda başlangıç popülasyonu Fidan gelişim algoritmasının başlangıç popülasyonunu oluşturmadaki mantığı ile oluşturulmuştur.

**Adım 17:** Bu adımda başlangıç popülasyonu Fidan gelişim algoritması ile oluşturulurken; minimum ve maksimum değerler, eğitim datası içerisinde, her özelliğin kendi sütununa ait en küçük ve en büyük değerler olacak şekilde seçilmiştir.  $\gamma=0,618$ ,  $\beta_0=0,618$ ,  $\alpha=0,618$  ve  $i_0=0,618$ , başlangıç popülasyonu Fidan gelişim algoritması yöntemi ile olursa sonuçlar Tablo 5.118.'de olduğu gibidir.

**Tablo 5.118.** Ateş böceği algoritması Adım 17 için sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	BEST (1,5)	BEST (1,6)	BEST (1,7)	BEST (1,8)	BEST (1,9)	Doğruluk oranı (%)
1	1	1	1	1	1	1	1	1	95
8	8	6	6	6	6	8	8	8	95
1	1	2	2	2	2	1	1	1	94
7	7	7	7	7	7	9	9	9	95
8	8	7	7	7	7	8	8	8	95
8	8	8	8	8	8	8	8	8	95
7	7	7	7	7	7	7	7	7	96
9	9	9	9	9	9	9	9	9	95
7	7	7	7	7	7	8	8	8	95
1	1	2	2	2	2	1	1	1	94
									Ortalama =94,9

**Adım 18:** Bu adımda başlangıç popülasyonu Fidan Gelişim Algoritması ile oluşturulurken; minimum ve maksimum değerler, eğitim datası içerisinde, her özelliğin kendi sütununa ait en küçük ve en büyük değerler olacak şekilde seçilmiştir.  $\gamma=0,1$ ,  $\beta_0=0,1$ ,  $\alpha=0,1$  ve  $i_0=0,1$ , başlangıç popülasyonu Fidan gelişim algoritması yöntemi ile olursa sonuçlar Tablo 5.119.'da olduğu gibidir.

**Tablo 5.119.** Ateş böceği algoritması Adım 18 için sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	BEST (1,5)	BEST (1,6)	BEST (1,7)	BEST (1,8)	BEST (1,9)	Doğruluk oranı (%)
5	5	5	5	5	5	5	5	5	84
4	4	6	6	6	6	4	4	4	82
9	9	9	9	9	9	1	1	1	89
8	8	8	8	8	8	2	2	2	91
2	2	2	2	2	2	8	8	8	85
9	9	9	9	9	9	1	1	1	89
8	8	8	8	8	8	1	1	1	89
2	2	2	2	2	2	7	7	7	84
9	9	9	9	9	9	2	2	2	89
4	4	5	5	5	5	4	4	4	77
									Ortalama =85,9

**Adım 19:** Bu adımda başlangıç popülasyonu Fidan gelişim algoritması ile oluşturulurken; minimum ve maksimum değerler, eğitim datası içerisinde, her özelliğin kendi sütununa ait en küçük ve en büyük değerler olacak şekilde seçilmiştir.  $\gamma=0,4$ ,  $\beta_0=0,4$ ,  $\alpha=0,4$  ve  $i_0=0,4$ , başlangıç popülasyonu Fidan gelişim algoritması yöntemi ile olursa sonuçlar Tablo 5.120.'de olduğu gibidir.

**Tablo 5.120.** Ateş böceği algoritması Adım 19 için sonuçlar

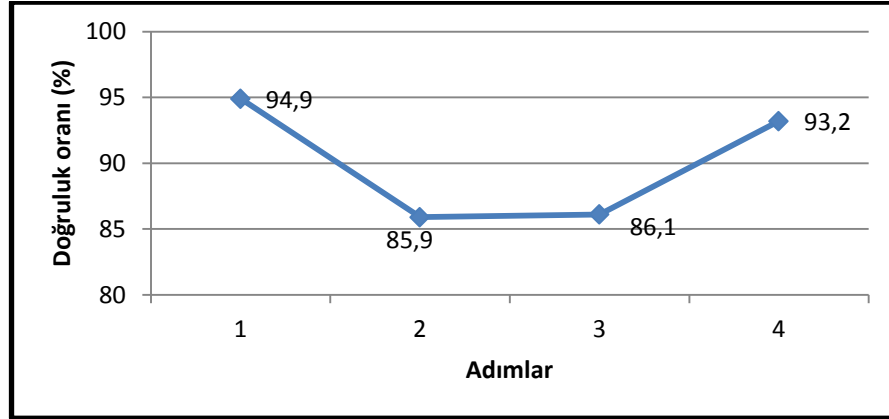
BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	BEST (1,5)	BEST (1,6)	BEST (1,7)	BEST (1,8)	BEST (1,9)	Doğruluk oranı (%)
8	8	8	8	8	8	2	2	2	91
9	9	9	9	9	9	1	1	1	89
8	8	8	8	8	8	1	1	1	89
5	5	5	5	5	5	5	5	5	84
1	1	1	1	1	1	8	8	8	88
2	2	2	2	2	2	7	7	7	84
7	7	7	7	7	7	2	2	2	86
7	7	7	7	7	7	1	1	1	84
4	4	5	5	5	5	4	4	4	77
9	9	9	9	9	9	2	2	2	89
									Ortalama =86,1

**Adım 20:** Bu adımda başlangıç popülasyonu Fidan gelişim algoritması ile oluşturulurken; minimum ve maksimum değerler, eğitim datası içerisinde, her özelliğin kendi sütununa ait en küçük ve en büyük değerler olacak şekilde seçilmiştir.  $\gamma=0,85$ ,  $\beta_0=0,85$ ,  $\alpha=0,85$  ve  $i_0=0,85$ , başlangıç popülasyonu Fidan gelişim algoritması yöntemi ile olursa sonuçlar Tablo 5.121.'de olduğu gibidir.

**Tablo 5.121.** Ateş böceği algoritması Adım 20 için sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	BEST (1,5)	BEST (1,6)	BEST (1,7)	BEST (1,8)	BEST (1,9)	Doğruluk oranı (%)
2	2	2	2	2	2	1	1	1	93
2	2	3	3	3	3	2	2	2	90
8	8	8	8	8	8	8	8	8	95
1	1	2	2	2	2	1	1	1	94
3	3	3	3	3	3	1	1	1	90
2	2	2	2	2	2	2	2	2	92
1	1	1	1	1	1	1	1	1	95
9	9	7	7	7	7	9	9	9	95
9	9	9	9	9	9	9	9	9	95
1	1	3	3	3	3	1	1	1	93
									Ortalama =93,2

Adım 17, 18, 19 ve 20 dikkatlice incelendiğinde başarı oranının en yüksek, 17. adımda bütün parametre değerlerinin 0,618 olduğu durumda olduğu görülecektir.



Şekil 5.11. Ateş böceği algoritması Adım 17,18, 19 ve 20 için ortalama doğruluk oranlarının grafiği

Şekil 5.11.'den görüldüğü üzere ortalama başarı oranı %94,9 ile, tüm denemelerin en yüksek oranına ulaşmıştır. Ayrıca tek tek elde edilen sonuçlara bakıldığında da %96 ile daha önce elde edilmiş olan oran burada da tekrarlanmıştır. Dolayısıyla hem tekil başarı oranlarında hem de ortalama başarı oranında bu adım en yüksek sonuçlara ulaşılan adım olmuştur. Buradan hem Fidan gelişim algoritmasının etkisini hem de altın oranın etkisini bir kez daha görebilmekteyiz.

**Adım 21:** Bu adımda başlangıç popülasyonu Fidan gelişim algoritması ile oluşturulurken; minimum ve maksimum değerler, eğitim datası içerisinde, tüm datanın en küçük ve en büyük değerleri olacak şekilde seçilmiştir.  $\gamma=0,618$ ,  $\beta_0=0,618$ ,  $\alpha=0,618$  ve  $i_0=0,618$ , başlangıç popülasyonu Fidan gelişim algoritması yöntemi ile olursa sonuçlar Tablo 5.122.'de olduğu gibidir.

Tablo 5.122. Ateş böceği algoritması Adım 21 için sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	BEST (1,5)	BEST (1,6)	BEST (1,7)	BEST (1,8)	BEST (1,9)	Doğruluk oranı (%)
1	1	3	3	3	3	1	1	1	93
2	2	2	2	2	2	1	1	1	93
8	8	8	8	8	8	8	8	8	95
1	1	1	1	1	1	1	1	1	95
8	8	6	6	6	6	8	8	8	95
1	1	3	3	3	3	1	1	1	93
1	1	2	2	2	2	1	1	1	94
7	7	7	7	7	7	9	9	9	95
2	2	3	3	3	3	2	2	2	90
2	2	2	2	2	2	2	2	2	92
									Ortalama =93,5



**Adım 22:** Bu adımda başlangıç popülasyonu Fidan gelişim algoritması ile oluşturulurken; minimum ve maksimum değerler, eğitim datası içerisinde, tüm datanın en küçük ve en büyük değerleri olacak şekilde seçilmiştir.  $\gamma=0,3$ ,  $\beta_0=0,3$ ,  $\alpha=0,3$  ve  $i_0=0,3$ , başlangıç popülasyonu Fidan gelişim algoritması yöntemi ile olursa sonuçlar Tablo 5.123.'de olduğu gibidir.

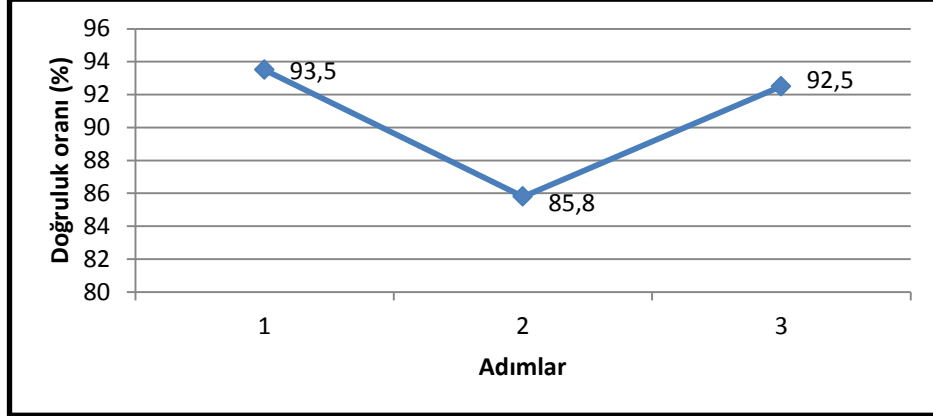
**Tablo 5.123.** Ateş böceği algoritması Adım 22 için sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	BEST (1,5)	BEST (1,6)	BEST (1,7)	BEST (1,8)	BEST (1,9)	Doğruluk oranı (%)
8	8	8	8	8	8	1	1	1	89
6	6	4	4	4	4	6	6	6	84
3	3	7	7	7	7	3	3	3	81
9	9	9	9	9	9	1	1	1	89
5	5	5	5	5	5	5	5	5	84
4	4	6	6	6	6	4	4	4	82
9	9	9	9	9	9	2	2	2	89
1	1	1	1	1	1	8	8	8	89
2	2	2	2	2	2	8	8	8	85
7	7	7	7	7	7	2	2	2	86
									Ortalama =85,8

**Adım 23:** Bu adımda başlangıç popülasyonu Fidan gelişim algoritması ile oluşturulurken; minimum ve maksimum değerler, eğitim datası içerisinde, tüm datanın en küçük ve en büyük değerleri olacak şekilde seçilmiştir.  $\gamma=0,9$ ,  $\beta_0=0,9$ ,  $\alpha=0,9$  ve  $i_0=0,9$ , başlangıç popülasyonu Fidan gelişim algoritması yöntemi ile olursa sonuçlar Tablo 5.124.'de olduğu gibidir.

**Tablo 5.124.** Ateş böceği algoritması Adım 23 için sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	BEST (1,5)	BEST (1,6)	BEST (1,7)	BEST (1,8)	BEST (1,9)	Doğruluk oranı (%)
2	2	2	2	2	2	2	2	2	92
2	2	2	2	2	2	1	1	1	93
1	1	3	3	3	3	1	1	1	93
2	2	2	2	2	2	1	1	1	93
3	3	3	3	3	3	2	2	2	89
1	1	1	1	1	1	1	1	1	95
1	1	2	2	2	2	1	1	1	94
2	2	3	3	3	3	2	2	2	87
9	9	7	7	7	7	9	9	9	95
1	1	2	2	2	2	1	1	1	94
									Ortalama =92,5



Şekil 5.12. Ateş böceği algoritması Adım 21, 22 ve 23 için ortalama doğruluk oranlarının grafiği

Şekil 5.11. ve Şekil 5.12. göstermektedir ki; Fidan gelişim algoritması ile başlangıç popülasyonu oluştururken minimum ve maksimum değerleri tüm örneklerin içerisinde seçmektense, her sütunun kendine ait olan minimum ve maksimum değerlerini kullanmak daha mantıklı sonuçlar vermektedir. Elde edilen en iyi başarı oranı 17. Adımda %94,9 ile Fidan gelişim algoritmasının her sütun için kendi minimum ve maksimum değerleri ile başlandığında elde edilmiştir.

Bu bölümde, ateş böceği algoritması veri kümeleme için başarı ile uygulanmıştır. Altın oranın etkileri, bu uygulamada da açıkça görülmüştür. Gerek ışık şiddeti hesaplanırken, gerek çekicilik değeri hesaplanırken ve gerekse, popülasyonun yeni değerleri hesaplanırken kullanılan parametreleri altın oran değerinde seçmek doğruluk oranları her zaman iyiye götürmüştür.

Işık şiddeti ve çekiciliği hesaplanırken kullanılan denklemlerdeki başlangıç değerleri ve eksponansiyel değişen hesabın üstel parametresi, uygun seçilmediğinde ve iyi bir oranda değiştirilmediğinde sonuçları kötü etkilemektedir. Altın oran değerinde seçilen bu parametrelerin doğruluk oranlarını iyileştirdiği görülmüştür. Bir önceki çözüm değerini kullanarak hesaplanan yeni çözümlerde, göz önüne alınan parametrenin altın oran değerinde artırılması çözüme düzgün bir yakınsamayı sağlamaktadır.

#### 5.4. Yarasa Algoritması Uygulama Yazılımı ve Sonuçları

Bu bölümde gerçekleştirilen uygulamada kullanılan veriler bölüm 5.1. ve bölüm 5.2.'deki veri setindeki veriler ile aynıdır. Uygulamada aynı veri seti ile çalışmaktaki amacımız, yine Yarasa algoritmasının aynı veri gurubu üzerindeki sonuçlarını karşılaştırabilmek içindir. Aynı zamanda altın oran uygulamalarındaki etkileri gözlemleyebilmektir.

Literatürde Yarasa algoritması için öngörülen, sürekli zamanlı matematiksel fonksiyonların çözümünde ve endüstriyel uygulamalarda uygulanabilir olmasıdır [75]. Bu bölümde ayrık zamanlı bir uygulama örneği ile veri kümeleme denenmiştir. Sonuçta, ortalamada %91,67; farklı zamanlarda ayrı ayrı yapılan denemelerde ise %95 e varan başarı oranlarına ulaşılmıştır. Bu da ayrık zamanlı uygulamaların Yarasa algoritması üzerinde çok iyi bir başarı gösterebileceğini göstermiştir.

Geliştirilen yazılım başlangıçta altın oran değeri hiçbir adımda kullanılmadan geliştirilmiş ve en iyi 10 sonuç arasında, en yüksek 91,67, ortalamada % 84,83 başarı oranı elde edilmiştir. Uygulama yazılımının farklı adımlarında kullanılan altın oran değeri sonucunda en yüksek başarı oranı %95, ortalama başarı oranı %91 değerine ulaşmıştır.

Veri setinde 150 farklı örnek mevcuttur. Her bir sınıftan 50 adet örnek vardır. Geliştirilen uygulama yazılımında, her sınıf için 30 örnek algoritmanın eğitim verileri için kullanılmıştır. Her sınıftan 20 örnekte test için bırakılmıştır.

Uygulama yazılımımız sonuçta BEST olarak isimlendirilen ve 1X4 boyutunda bir matrisi çıktı olarak vermektedir. Bu matrisin her elemanı test kümemizin sütun elemanlarının her birindeki elemanlarla teker teker karşılaştırılıp, bir toplam hata değeri elde edilmektedir. Amaç bu hata değerini minimize etmektir. BEST matrisinin her bir elemanı, nümerik optimum çözüm değerlerini vermektedir.

**Tablo 5.125.** Yarasa algoritması uygulama yazılımı BEST matris temsili

BEST(1,2)	BEST(1,3)	BEST(1,4)
-----------	-----------	-----------

Aşağıdaki algoritmik hesaplama uygulama yazılımının temel çalışma prensibini vermektedir.

```

for i=1:16
    farklartoplami(i,1) =  $\sum_{i=1}^4 \sum_{j=1}^{90} (|\text{popülasyon}(i, 1) - \text{egitim}(1, j)|)$ 
    farklartoplami (i,2) =  $\sum_{i=1}^4 \sum_{j=1}^{90} (|\text{popülasyon}(i, 2) - \text{egitim}(2, j)|)$ 
    farklartoplami(i,3) =  $\sum_{i=1}^4 \sum_{j=1}^{90} (|\text{popülasyon}(i, 3) - \text{egitim}(3, j)|)$ 
    farklartoplami(i,4) =  $\sum_{i=1}^4 \sum_{j=1}^{90} (|\text{popülasyon}(i, 4) - \text{egitim}(4, j)|)$ 
end
uygunluk(1,k) =  $\sum$  farkların_toplamı(k, :)
[minimum,indeks]=min(fitness)
BEST=x(indeks,:)

```

Popülasyonun her değerinin eğitim kümesinin ilgili sütunuyla farkları tek tek alınıp, farklartoplami matrisine atılıyor. Aslında burada popülasyon kümesinin eğitim değerleri ile manhattan uzaklık farkları alınıyor. Buradaki 16 değeri Yarasa algoritmasında kullanılan popülasyondaki yarasa değerlerini temsil eden çözüm sayısıdır. Bu uygulama için 16 seçilmiştir.

**Tablo 5.126.** Yarasa algoritması uygulama yazılımı farklar matrisinin temsili

fark(1,1)	fark (1,2)	fark (1,3)	fark (1,4)
fark (2,1)	fark (2,2)	fark (2,3)	fark (2,4)
.	.	.	.
.	.	.	.
fark (16,1)	fark (16,2)	fark (16,3)	fark(16,4)

$\underbrace{\hspace{15em}}_{\sum (\text{diffrence}(k,:))}$

**Tablo 5.127.** Yarasa algoritması uygulama yazılımı farklar matrisinin toplamının temsili

farklartoplami(1,1)	farklartoplami(1,2)	.	.	.	farklartoplami(1,16)
---------------------	---------------------	---	---	---	----------------------

Daha sonra farklartoplami matrisinin her satırının toplamı alınıp fitness matrisine atılıyor. Aslında fitness matrisi hata matrisi oluyor. Bu matris küçükten büyüğe doğru dizildiğinde en küçük değer, aslında eğitim kümesinden uzaklıklar açısından en küçük değer yani “BEST matris” oluyor.

Aynı mantıkla algoritmayı durdurma adımında ENK değeri kullanılmıştır. Aslında hata tolerans değeridir. Uygulamada altın oran denemelerinden önce 300 değerinin altına nadir olarak düşmüştür. Altın oran denemelerinden sonra 285 civarına kadar düşmüştür. Bu nedenle algoritmayı durdurma adımı olarak, bir yere kadar ENK = 300; daha sonrada 290 ve

285 olarak denenmiştir. 22 farklı adımda gerçekleştirilen uygulama yazılımının adımları aşağıda sıra ile verilmiştir.

**Adım 1:** Minimum ve maksimum frekans değerlerimiz  $f_{\min} = 0$  ,  $f_{\max} = 1$  olarak seçilmiştir. Bu değerler algoritmanın tüm adımlarında aynıdır. Bu tasarlayıcıya bağlı olarak değişebilecek bir değer olup probleme göre uygulayıcı tarafından farklı seçilebilir. Tablo 5.128. adım 1 başlangıç koşullarını, tablo 5.129. adım 1 in sonuçlarını, şekil 5.13. ise grafik temsilini göstermektedir.

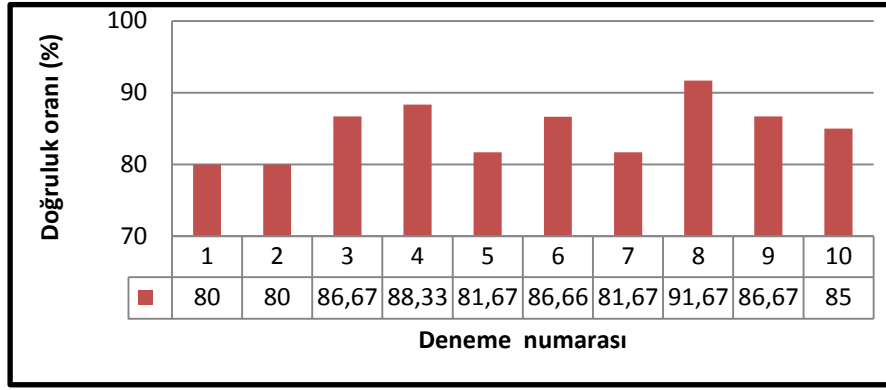
**Tablo 5.128.** Yarasa algoritması Adım 1 için başlangıç koşulları

$f_{\min} = 0, f_{\max} = 1$
$A = 0,5 \quad r = 0,4$
Başlangıç popülasyon değerleri rastgele
$r = r + 0,1 * \text{rand}$
$A = A - 0,1 * \text{rand}$
$xyeni = \text{BEST} + (0,01 * \text{rand}_1 * \text{rand}_2)$
$300 < \text{ENK}$

Burada x matrisi dediğimiz, çözüm olabilecek değerlerin her birinin bir yarasa olarak temsilidir. Başlangıçta rastgele olarak seçilmiştir. İlerleyen adımlarda başlangıç popülasyonunun düzgün dağılımını sağlamak amacıyla fidan gelişim algoritması yardımıyla oluşturulmuş durumdaki sonuçları da verilecektir. r ve A değerlerindeki değişim bu adımda rastgele denenmiştir. 0,1 ile çarpma işlemi r ve A değerlerinin istenilen aralıkta kalmalarını sağlamak içindir. Bu adım için elde edilen sonuçlar aşağıdaki gibidir.

**Tablo 5.129.** Yarasa algoritması Adım 1 uygulama sonuçları

	BEST(1,1)	BEST(1,2)	BEST(1,3)	BEST(1,4)	Doğruluk oranı (%)
1	5,7982	3,2083	3,4683	1,4755	80
2	5,559	3,2691	4,0029	0,9788	80
3	5,8942	2,6904	4,6722	1,5015	86,67
4	5,5585	3,0271	3,719	1,5671	88,33
5	5,4695	3,1935	4,2918	1,7086	81,67
6	5,8797	2,9816	3,9443	1,6883	86,66
7	5,4918	2,9724	3,965	0,8412	81,67
8	5,3913	3,292	4,2675	1,2591	91,67
9	5,9356	2,9228	4,8722	1,2761	86,67
10	5,8321	2,9068	4,6811	1,6743	85
					Ortalama= 84,83%



Şekil 5.13. Yarasa algoritması Adım 1 için sonuç grafiği

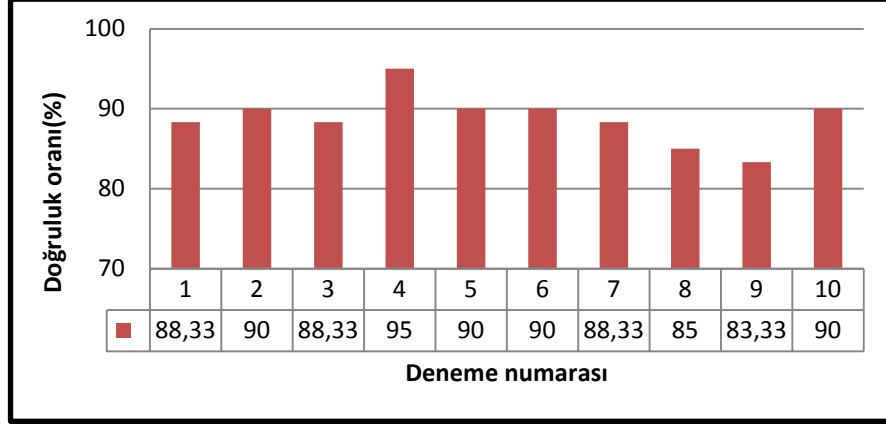
**Adım 2:** Bu adımda uygulanan durumlar ve sonuçları tablo 5.130. , tablo 5.131. ve şekil 5.14.'de verilmiştir.

**Tablo 5.130.** Yarasa algoritması Adım 2 için başlangıç koşulları

$f_{min}=0$ $f_{max}=1$
$A=0,5$ $r=0,4$
Başlangıç popülasyon değerleri rastgele
$r=r+0,1*1,618$
$A=A-0,1*1,618$
$xyeni=BEST+(0,01*rand_1*rand_2)$
$300<ENK$

**Tablo 5.131.** Yarasa algoritması Adım 2 uygulama sonuçları

	BEST(1,1)	BEST(1,2)	BEST(1,3)	BEST(1,4)	Doğruluk oranı (%)
1	5,6482	2,8145	3,9875	1,6569	88,33
2	5,7973	3,2814	3,8221	1,4799	90
3	6,3556	3,0654	4,2564	1,4534	88,33
4	5,9455	3,0972	4,2112	0,9022	95
5	5,3701	2,7445	4,561	1,396	90
6	5,8109	3,2283	4,0788	1,1112	90
7	5,7937	3,0596	3,8478	1,6747	88,33
8	5,9675	3,2612	4,5413	1,316	85
9	5,5986	2,9782	4,3335	1,7602	83,33
10	5,8959	2,9037	3,7798	1,1478	90
					Ortalama= 88,83%



Şekil 5.14. Yarasa algoritması Adım 2 için sonuç grafiği

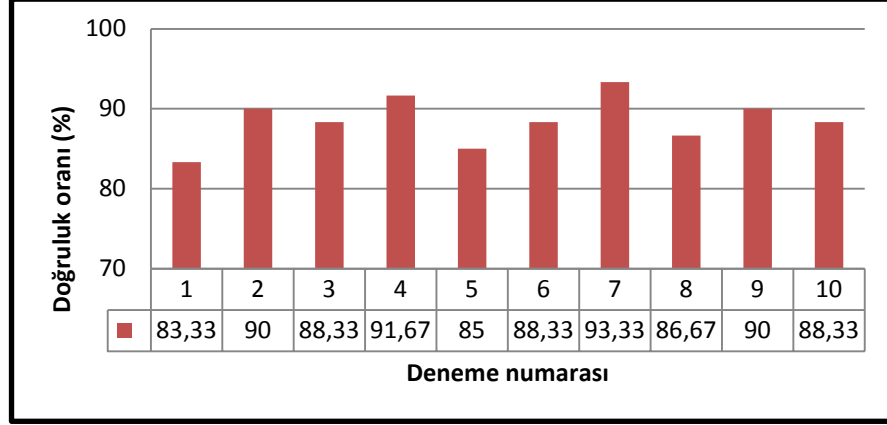
**Adım 3:** Bu adımda uygulanan durumlar ve sonuçları tablo 5.132. , tablo 5.133. ve şekil 5.15.'de verilmiştir.

**Tablo 5.132.** Yarasa algoritması Adım 3 için başlangıç koşulları

$f_{min}=0$ $f_{max}=1$
$A=0,5$ $r=0,4$
Başlangıç popülasyon değerleri rastgele
$r=r+0,1*1,3$
$A=A-0,1*1,3$
$xyeni=BEST+(0,01*rand_1*rand_2)$
$300<ENK$

**Tablo 5.133.** Yarasa algoritması Adım 3 uygulama sonuçları

	BEST(1,1)	BEST(1,2)	BEST(1,3)	BEST(1,4)	Doğruluk oranı (%)
1	6,1656	3,1929	3,9378	1,6052	83,33
2	6,0538	3,3265	4,1167	1,0058	90
3	6,0293	3,1498	3,8641	1,5872	88,33
4	5,783	2,9997	4,3698	0,9387	91,67
5	5,6441	3,2038	4,6926	1,3829	85
6	5,7248	2,8225	4,3473	1,5801	88,33
7	5,2945	3,1151	4,2592	1,4326	93,33
8	5,9023	2,8759	4,1428	0,6565	86,67
9	6,2477	2,9077	4,5855	1,4957	90
10	6,0935	2,9037	4,2133	1,5174	88,33
					Ortalama= 88.50%



Şekil 5.15. Yaras algoritması Adım 3 için sonuç grafiği

**Adım 4:** Bu adımda uygulanan durumlar ve sonuçları tablo 5.134. , tablo 5.135. ve şekil 5.16.'da verilmiştir.

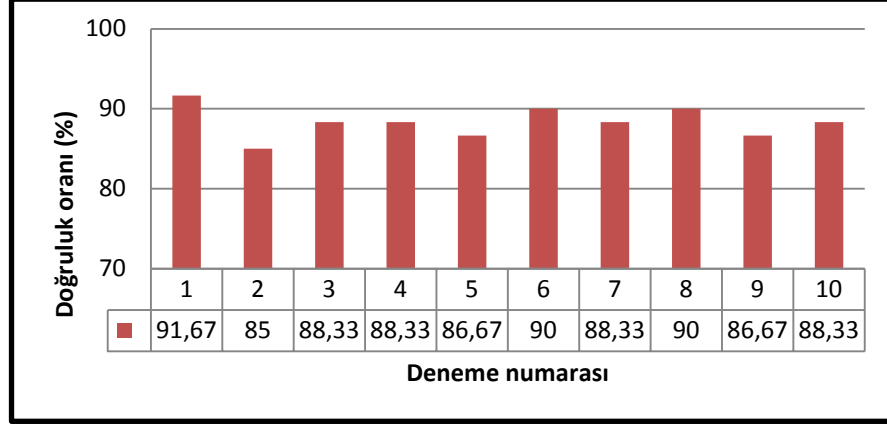
**Tablo 5.134.** Yaras algoritması Adım 4 için başlangıç koşulları

$f_{min}=0$ $f_{max}=1$
$A=0,5$ $r=0,4$
Başlangıç popülasyon değerleri rastgele
$r=r+0,1*0,9$
$A=A-0,1*0,9$
$xyeni=BEST+(0,01*rand_1*rand_2)$
$300<ENK$

**Tablo 5.135.** Yaras algoritması Adım 4 uygulama sonuçları

	BEST(1,1)	BEST(1,2)	BEST(1,3)	BEST(1,4)	Doğruluk oranı (%)
1	5,7238	3,2825	4,4061	1,0699	91,67
2	5,9202	3,2512	4,6391	1,0311	85
3	5,8109	3,2283	4,0788	1,1112	88,33
4	5,7937	3,0596	3,8478	1,6747	88,33
5	6,135	2,7908	4,254	1,591	86,67
6	5,4014	2,7925	4,0342	1,5028	90
7	5,9522	2,7202	3,6413	1,3098	88,33
8	5,6901	3,0998	4,2091	1,4868	90
9	6,2218	3,0829	4,3012	1,645	86,67
10	5,5624	3,2373	4,1617	1,5037	88,33
					Ortalama=88,33%





Şekil 5.16. Yaras algoritması Adım 4 için sonuç grafiği

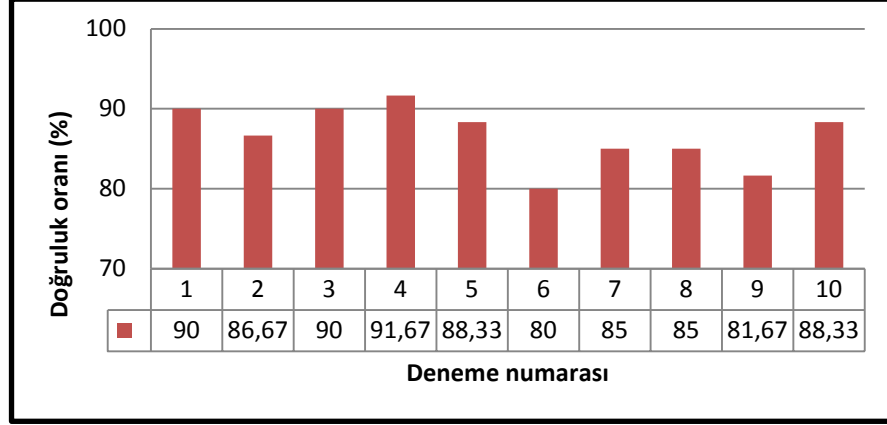
**Adım 5:** Bu adımda uygulanan durumlar ve sonuçları tablo 5.136. , tablo 5.137. ve şekil 5.17.'de verilmiştir.

**Tablo 5.136.** Yaras algoritması Adım 5 için başlangıç koşulları

$f_{min}=0$ $f_{max}=1$
$A=0,5$ $r=0,4$
Başlangıç popülasyon değerleri rastgele
$r=r+0,1*0,618$
$A=A-0,1*0,618$
$xyeni=BEST+(0,01*rand_1*rand_2)$
$300<ENK$

**Tablo 5.137.** Yaras algoritması Adım 5 uygulama sonuçları

	BEST(1,1)	BEST(1,2)	BEST(1,3)	BEST(1,4)	Doğruluk oranı (%)
1	6,0285	2,8706	4,4934	0,8848	90
2	5,5097	2,9976	4,2023	1,6959	86,67
3	5,7065	3,251	4,3851	0,8411	90
4	5,1868	3,0434	4,6449	1,3591	91,67
5	6,0033	3,1368	4,6224	1,3039	88,33
6	5,89	3,0877	4,8443	1,6262	80
7	5,8606	2,773	4,2734	1,7748	85
8	5,8811	3,0787	4,4013	1,3705	85
9	5,9346	3,1817	4,3561	1,6238	81,67
10	5,539	3,0904	4,1963	1,0408	88,33
					Ortalama= 86.67%



Şekil 5.17. Yaras algoritması Adım 5 için sonuç grafiği

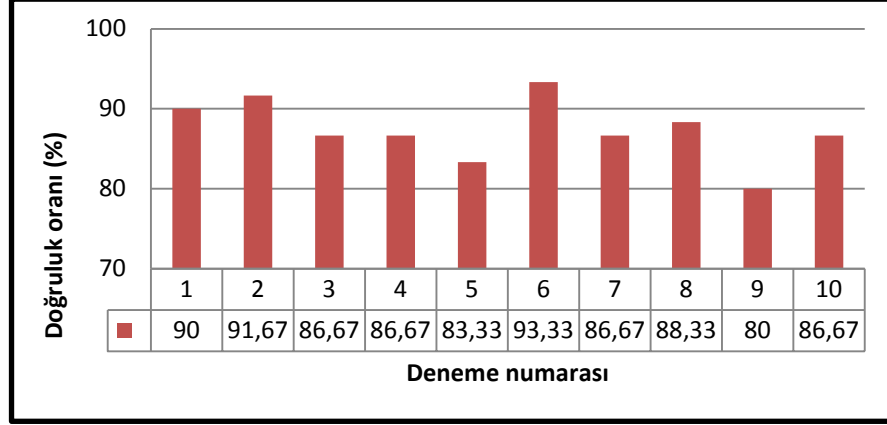
**Adım 6:** Bu adımda uygulanan durumlar ve sonuçları tablo 5.138. , tablo 5.139. ve şekil 5.18.'de verilmiştir.

**Tablo 5.138.** Yaras algoritması Adım 6 için başlangıç koşulları

$f_{min}=0$ $f_{max}=1$
$A=0,5$ $r=0,4$
Başlangıç popülasyon değerleri rastgele
$r=r+0,1*1,9$
$A=A-0,1*1,9$
$xyeni=BEST+(0,01*rand_1*rand_2)$
$300<ENK$

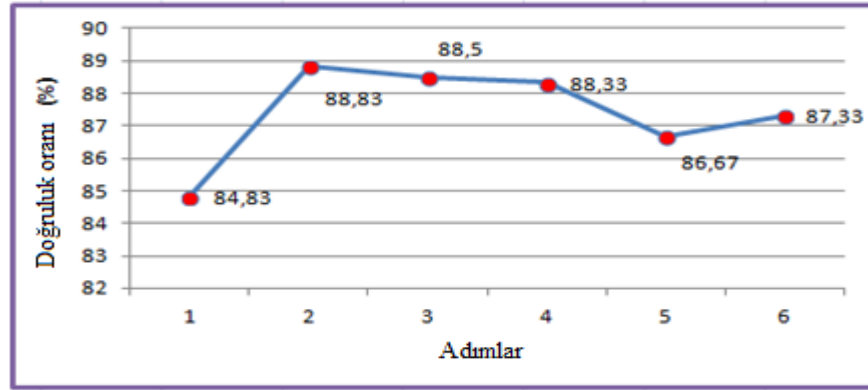
**Tablo 5.139.** Yaras algoritması Adım 6 uygulama sonuçları

	BEST(1,1)	BEST(1,2)	BEST(1,3)	BEST(1,4)	Doğruluk oranı (%)
1	5,5428	2,9706	3,9167	1,4039	90
2	6,0735	3,1871	4,0694	1,3041	91,67
3	6,0567	3,0359	4,7392	1,5554	86,67
4	6,256	3,1103	4,4894	1,1251	86,67
5	5,6777	2,7417	3,8074	1,0747	83,33
6	5,1949	3,2005	4,2563	1,4572	93,33
7	5,4044	2,9377	3,6044	1,5291	86,67
8	5,939	2,928	4,7471	0,9119	88,33
9	5,4503	3,0502	3,8075	0,8862	80
10	6,0824	3,1374	4,3671	1,5674	86,67
					Ortalama=% 87,33



Şekil 5.18. Yarasa algoritması Adım 6 için sonuç grafiği

Buraya kadar olan 6 adımda  $r$  ve  $A$  değerlerinin değişim oranı test edilmiştir. Yapılan farklı denemeler göstermiştir ki, en başarılı sonuçlar  $r$  ve  $A$  değerlerinin değişim oranı için 1,618 değeri olan altın oran değerinin kullanıldığı durumda görülmüştür. Bu durum şekil 5.19.'da gösterilmektedir.



Şekil 5.19. Yarasa algoritması Adım 1-Adım 6 arası ortalama doğruluk oranları grafik temsili

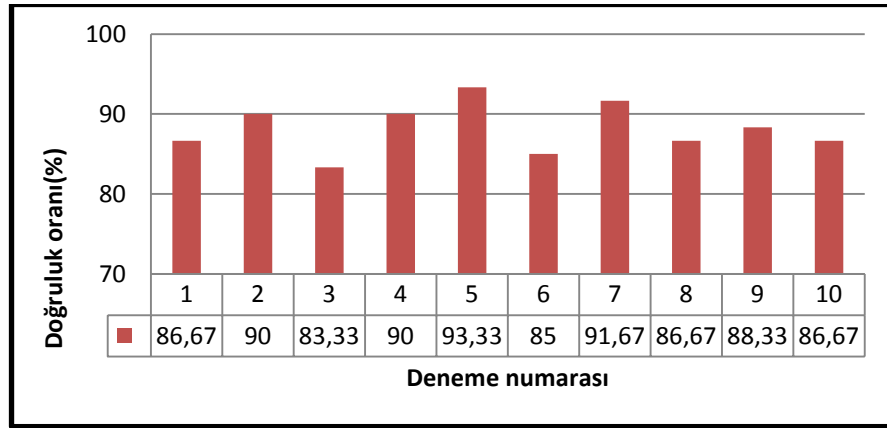
**Adım 7:** Bu adımda uygulanan durumlar ve sonuçları tablo 5.140. , tablo 5.141. ve şekil 5.20.'de verilmiştir.

**Tablo 5.140.** Yarasa algoritması Adım 7 için başlangıç koşulları

$f_{min}=0$ $f_{max}=1$
$A=0,5$ $r=0,4$
Başlangıç popülasyon değerleri rastgele
$r=r+0,1*rand$
$A=A-0,1*rand$
$xyeni=BEST+(0,01*1,618*rand)$
$300<ENK$

**Tablo 5.141.** Yarasa algoritması Adım 7 uygulama sonuçları

	BEST(1,1)	BEST(1,2)	BEST(1,3)	BEST(1,4)	Doğruluk oranı (%)
1	5,3585	3,0626	4,7666	1,4001	86,67
2	6,1057	2,9831	4,105	1,3089	90
3	5,5838	3,0337	3,4514	1,5289	83,33
4	6,0034	2,8129	3,9372	1,7039	90
5	6,0037	3,1419	3,7506	1,1172	93,33
6	5,3564	3,0917	4,6436	1,3727	85
7	5,8887	3,3057	4,1519	1,0917	91,67
8	5,6322	2,7455	4,2372	1,6611	86,67
9	5,9339	3,3949	4,3009	1,255	88,33
10	6,2626	3,1003	4,6407	1,3778	86,67
					Ortalama=% 88,17



**Şekil 5.20.** Yarasa algoritması Adım 7 için sonuç grafiği

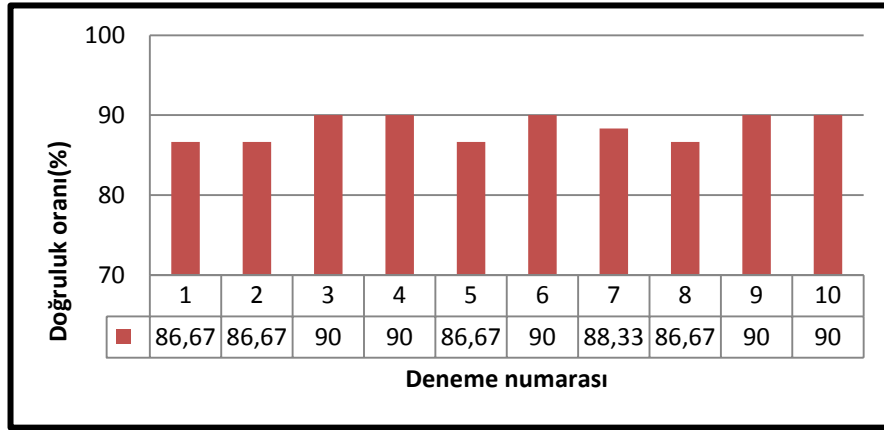
**Adım 8:** Bu adımda uygulanan durumlar ve sonuçları tablo 5.142. , tablo 5.143. ve şekil 5.21.'de verilmiştir.

**Tablo 5.142.** Yarasa algoritması Adım 8 için başlangıç koşulları

$f_{min}=0$ $f_{max}=1$
$A= 0,5$ $r=0,4$
Başlangıç popülasyon değerleri rastgele
$r=r+0,1*1,618$
$A=A-0,1*1,618$
$xyeni=BEST+(0,01*1,618*rand)$
$300<ENK$

**Tablo 5.143.** Yarasa algoritması Adım 8 uygulama sonuçları

	BEST(1,1)	BEST(1,2)	BEST(1,3)	BEST(1,4)	Doğruluk oranı (%)
1	5,8963	2,8683	4,0819	0,7816	86,67
2	5,5882	3,2667	3,8675	1,5558	86,67
3	5,8093	2,9812	4,5411	0,7939	90
4	5,9766	3,0247	4,2081	0,7738	90
5	5,6343	2,8375	4,1127	0,9775	86,67
6	5,6901	3,0998	4,2091	1,4868	90
7	5,7392	2,703	4,2503	1,4629	88,33
8	5,8074	2,8232	3,4124	1,326	86,67
9	5,6454	2,7784	3,9442	1,3135	90
10	6,1057	2,9831	4,105	1,3089	90
					Ortalama=%88,50



**Şekil 5.21.** Yarasa algoritması Adım 8 için sonuç grafiği

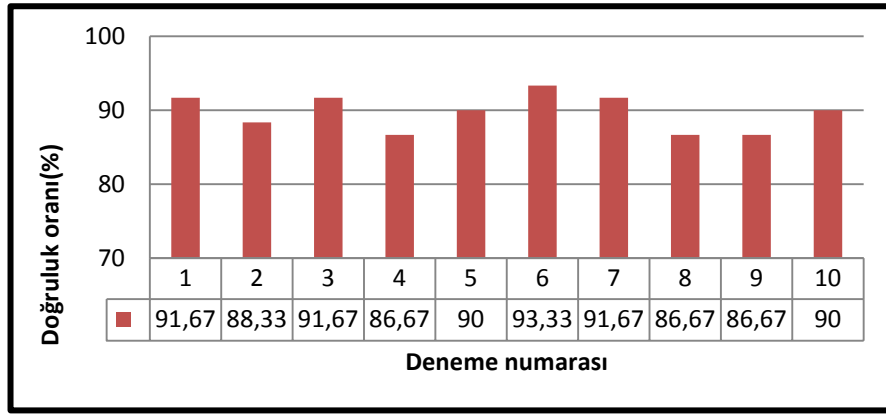
**Adım 9:** Bu adımda uygulanan durumlar ve sonuçları tablo 5.144. , tablo 5.145. ve şekil 5.22.'de verilmiştir.

**Tablo 5.144.** Yarasa algoritması Adım 9 için başlangıç koşulları

$f_{min}=0$ $f_{max}=1$
$A= 0,5$ $r=0,618$
Başlangıç popülasyon değerleri rastgele
$r=r+0,1*1,618$
$A=A-0,1*1,618$
$xyeni=BEST+(0,01*1,618*rand)$
$300<ENK$

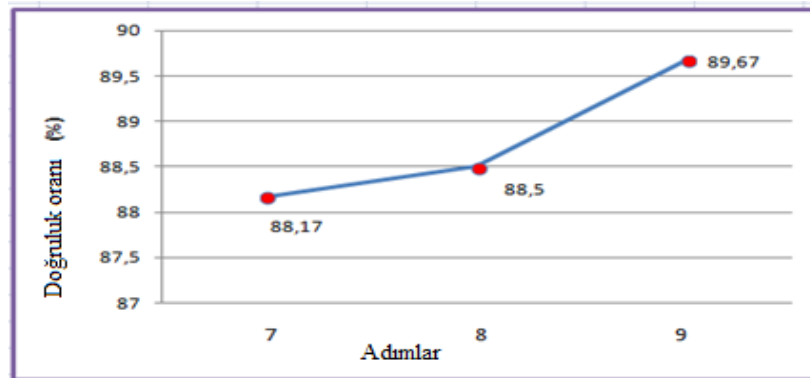
**Tablo 5.145.** Yarasa algoritması Adım 9 uygulama sonuçları

	BEST(1,1)	BEST(1,2)	BEST(1,3)	BEST(1,4)	Doğruluk oranı (%)
1	6,1178	2,9841	3,7094	1,5972	91,67
2	5,6766	2,9372	4,8764	1,211	88,33
3	5,5301	2,9364	4,0421	1,4371	91,67
4	5,7833	3,0628	3,6402	1,7356	86,67
5	5,6042	3,267	4,5551	0,9515	90
6	6,1883	3,1306	3,83	1,0106	93,33
7	5,7335	3,1221	3,69	1,4351	91,67
8	6,1196	2,7813	4,3136	1,5008	86,67
9	5,4975	2,9888	4,1189	1,0323	86,67
10	5,3562	3,1673	4,4422	0,9679	90
					Ortalama= % 89,67



**Şekil 5.22.** Yarasa algoritması Adım 9 için sonuç grafiği

Son üç adımdan görüldüğü üzere; BEST matris etrafında oluşturulacak yeni çözümlerde, altın oran değeri ile çarpıp bir artış sağlamak başarı oranını daha da artırmaktadır. Bu durum şekil 5.23.'de görülmektedir.



**Şekil 5.23.** Yarasa algoritması Adım 7-Adım 9 arası ortalama doğruluk oranları grafik temsili

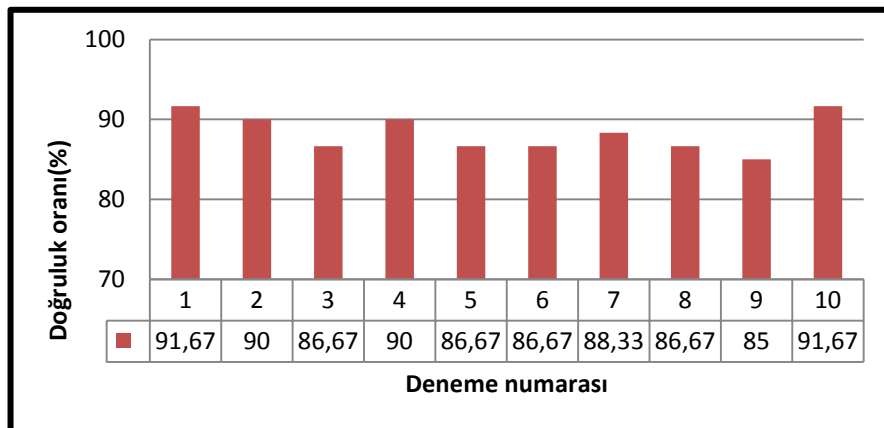
**Adım 10:** Bu adımda uygulanan durumlar ve sonuçları tablo 5.146. , tablo 5.147. ve şekil 5.24.'de verilmiştir. Bu adımdan itibaren başlangıç popülasyonunun değerleri fidan gelişim algoritmasına göre oluşturulmuştur.

**Tablo 5.146.** Yarasa algoritması Adım 10 için başlangıç koşulları

$f_{min}=0$ $f_{max}=1$
$A=0,5$ $r=0,4$
<u>Başlangıç popülasyonu fidan gelişim algoritması ile oluşturulmuştur</u>
$r=r+0,1*rand$
$A=A-0,1*rand$
$xyeni=BEST+(0,01*rand_1*rand_2)$
$300<ENK$

**Tablo 5.147.** Yarasa algoritması Adım 10 uygulama sonuçları

	BEST(1,1)	BEST(1,2)	BEST(1,3)	BEST(1,4)	Doğruluk oranı (%)
1	5,3593	2,8569	4,603	1,207	91,67
2	6,0451	3,0209	3,9363	1,571	90
3	5,8344	3,4499	4,067	1,2743	86,67
4	6,0034	2,8129	3,9372	1,7039	90
5	5,5475	2,981	4,3611	0,6833	86,67
6	5,2187	3,0411	3,9851	1,1852	86,67
7	5,7248	2,8225	4,3473	1,5801	88,33
8	5,8693	2,8683	4,0819	0,7816	86,67
9	5,5641	3,1198	3,9945	1,0364	85
10	6,2856	3,0212	3,8859	1,1074	91,67
					Ortalama= % 88,34



**Şekil 5.24.** Yarasa algoritması Adım 10 için sonuç grafiği

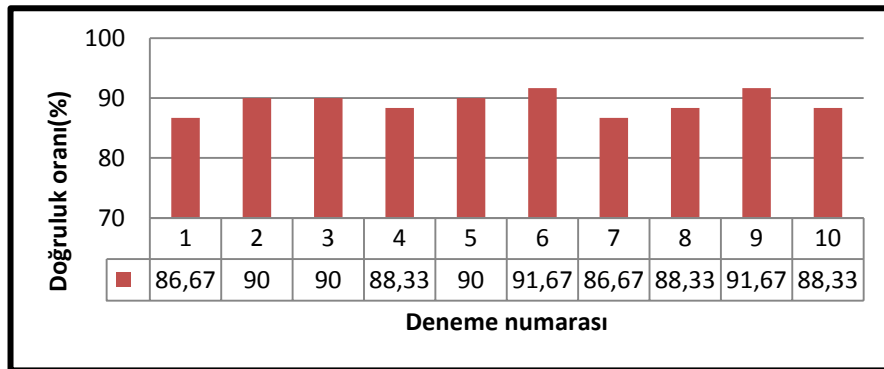
**Adım 11:** Bu adımda uygulanan durumlar ve sonuçları tablo 5.148. , tablo 5.149. ve şekil 5.25.'de verilmiştir.

**Tablo 5.148.** Yarasa algoritması Adım 11 için başlangıç koşulları

f_min=0 f_max=1
A= 0,5 r=0,4
<u>Başlangıç popülasyonu fidan gelişim algoritması ile oluşturulmuştur</u>
r=r+0,1*1,618
A=A-0,1*1,618
xyeni=BEST+(0,01*rand <sub>1</sub> *rand <sub>2</sub> )
300<ENK

**Tablo 5.149.** Yarasa algoritması Adım 11 uygulama sonuçları

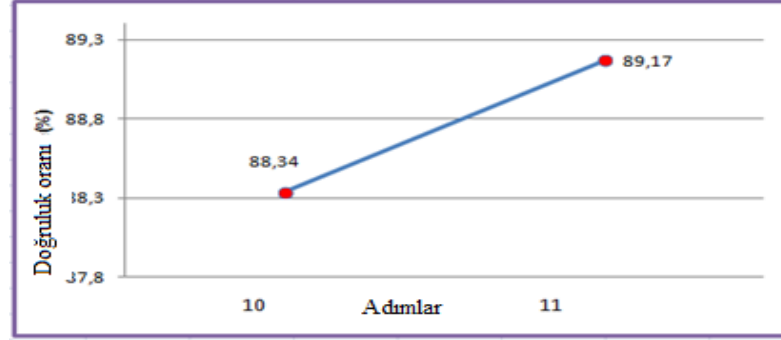
	BEST(1,1)	BEST(1,2)	BEST(1,3)	BEST(1,4)	Doğruluk oranı (%)
1	5,8344	3,4499	4,067	1,2743	86,67
2	6,0034	2,8129	3,9372	1,7039	90
3	5,4683	2,9698	4,6015	1,2289	90
4	5,7308	2,7114	3,9148	1,319	88,33
5	5,8609	3,254	4,3969	1,1264	90
6	5,3924	3,1675	4,5597	1,2845	91,67
7	5,9061	2,5642	4,2733	1,3513	86,67
8	6,363	2,993	4,1435	1,0705	88,33
9	5,7189	3,2236	4,05	1,1561	91,67
10	6,1597	2,8027	4,4453	1,5053	88,33
					Ortalama= % 89,17



**Şekil 5.25.** Yarasa algoritması Adım 11 için sonuç grafiği



Adım 10 ve Adım 11 dikkatli incelenecek olursa, Fidan gelişim algoritmasının başlangıç popülasyonunu oluşturmadaki mantığı kullanıldığı zaman, başarı oranı aynı parametreler ile oluşturulan fakat başlangıç popülasyonunun rastgele oluşturulduğu Adım 1 e göre daha iyidir. Adım 1 in ortalama başarı oranı %84,83, Adım 10 un ise %88,34 tür. 11. Adımda r ve A parametrelerindeki değişim oranı altın orana bağlı olarak değiştiğinde ortalama başarı oranı %%89,17 olmaktadır. Bu durum şekil 5.26.'da görülmektedir.



Şekil 5.26. Yaras algoritması Adım 10 ve Adım 11 ortalama doğruluk oranları grafik temsili

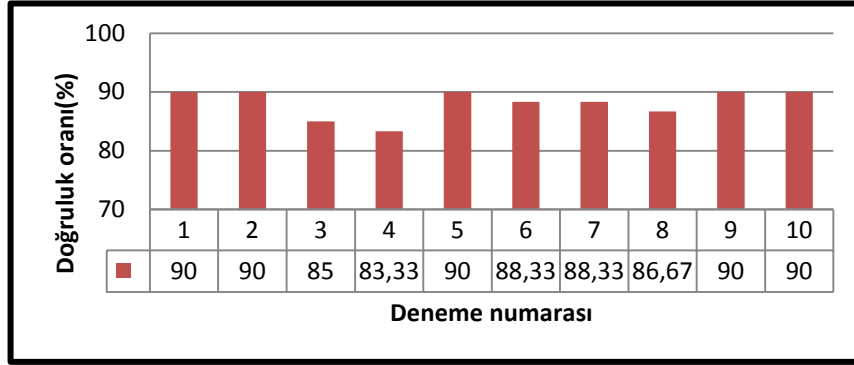
**Adım 12:** Bu adımda uygulanan durumlar ve sonuçları tablo 5.150. , tablo 5.151. ve şekil 5.27.'de verilmiştir.

**Tablo 5.150.** Yaras algoritması Adım 12 için başlangıç koşulları

$f_{min}=0$ $f_{max}=1$
$A= 0,5$ $r=0,4$
<u>Başlangıç popülasyonu fidan gelişim algoritması ile oluşturulmuştur</u>
$r=r+0,1*rand$
$A=A-0,1*rand$
$xyeni=BEST+(0,01*rand_1*rand_2)$
$300<ENK$

**Tablo 5.151.** Yarasa algoritması Adım 12 uygulama sonuçları

	BEST(1,1)	BEST(1,2)	BEST(1,3)	BEST(1,4)	Doğruluk oranı (%)
1	5,9663	2,8327	3,7904	1,4504	90
2	5,9766	3,0247	4,2081	0,7738	90
3	5,6343	2,8375	4,1127	0,9775	85
4	5,4833	3,0792	3,7252	1,1697	83,33
5	5,6901	3,0998	4,2091	1,4868	90
6	5,4966	2,7602	4,0995	1,2517	88,33
7	5,7392	2,703	4,2503	1,4629	88,33
8	5,8074	2,8232	3,4124	1,326	86,67
9	5,4851	3,0161	4,2949	1,0725	90
10	6,1849	2,8911	4,5958	1,3778	90
					Ortalama= % 88,17



**Şekil 5.27.** Yarasa algoritması Adım 12 için sonuç grafiği

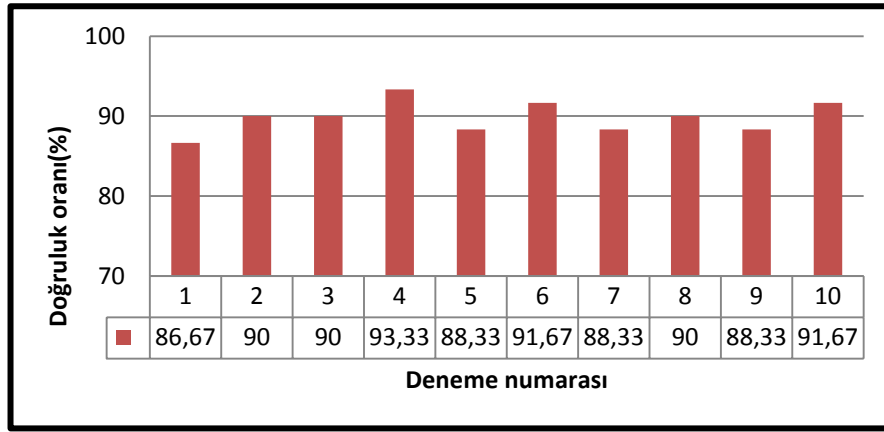
**Adım 13:** Bu adımda uygulanan durumlar ve sonuçları tablo 5.152. , tablo 5.153. ve şekil 5.28.'de verilmiştir.

**Tablo 5.152.** Yarasa algoritması Adım 13 için başlangıç koşulları

$f_{min}=0$ $f_{max}=1$
$A=0,5$ $r=0,4$
<u>Başlangıç popülasyonu fidan gelişim algoritması ile oluşturulmuştur</u>
$r=r+0,1*1,618$
$A=A-0,1*1,618$
$xyeni=BEST+(0,01*0,9*rand)$
$300<ENK$

**Tablo 5.153.** Yarasa algoritması Adım 13 uygulama sonuçları

	BEST(1,1)	BEST(1,2)	BEST(1,3)	BEST(1,4)	Doğruluk oranı (%)
1	5,7833	3,0628	3,6402	1,7356	86,67
2	6,2227	2,9649	4,338	1,2	90
3	5,9477	2,7774	4,2817	1,3176	90
4	5,2708	3,0154	4,305	1,4314	93,33
5	5,8301	2,7134	4,6399	1,1886	88,33
6	5,3789	3,2552	4,4333	1,2774	91,67
7	6,041	2,7742	4,6546	1,3835	88,33
8	5,8974	2,7121	4,3234	1,4182	90
9	5,7075	3,1914	3,4293	1,2067	88,33
10	6,33	2,8564	4,0628	1,2975	91,67
					Ortalama= % 89,83



**Şekil 5.28.** Yarasa algoritması Adım 13 için sonuç grafiği

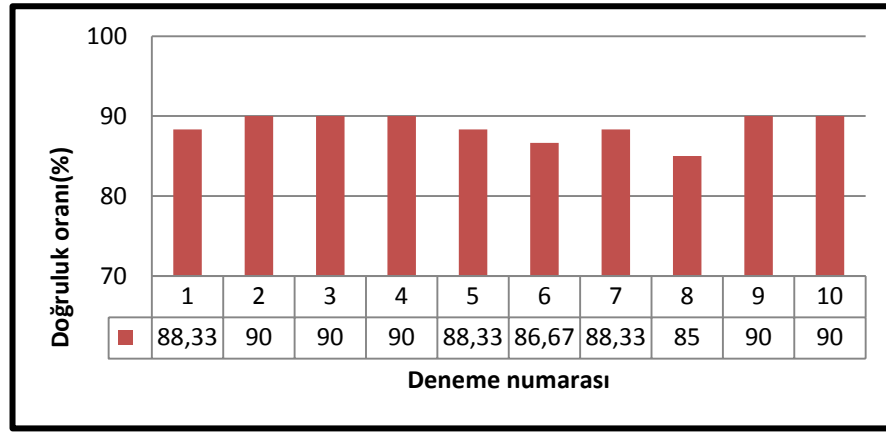
**Adım 14:** Bu adımda uygulanan durumlar ve sonuçları tablo 5.154. , tablo 5.155. ve şekil 5.29.'da verilmiştir.

**Tablo 5.154.** Yarasa algoritması Adım 14 için başlangıç koşulları

$f_{min}=0$ $f_{max}=1$
$A= 0,5$ $r=0,4$
<u>Başlangıç popülasyonu fidan gelişim algoritması ile oluşturulmuştur</u>
$r=r+0,1*rand$
$A=A-0,1*rand$
$xyeni=BEST+(0,01*1,618*rand)$
$300<ENK$

**Tablo 5.155.** Yarasa algoritması Adım 14 uygulama sonuçları

	BEST(1,1)	BEST(1,2)	BEST(1,3)	BEST(1,4)	Doğruluk oranı (%)
1	6,0251	2,7485	4,4641	1,216	88,33
2	5,4683	2,9698	4,6015	1,2289	90
3	5,8004	3,0261	4,0207	1,0525	90
4	6,2125	3,1414	3,9025	1,306	90
5	5,555	3,2388	4,4079	0,8061	88,33
6	5,8693	2,8683	4,0819	0,7816	86,67
7	5,5641	3,1198	3,9945	1,0364	88,33
8	5,8496	3,1345	3,7397	0,969	85
9	5,8093	2,9812	4,5411	0,7939	90
10	5,9766	3,0247	4,2081	0,7738	90
					Ortalama= % 88,67



**Şekil 5.29.** Yarasa algoritması Adım 14 için sonuç grafiği

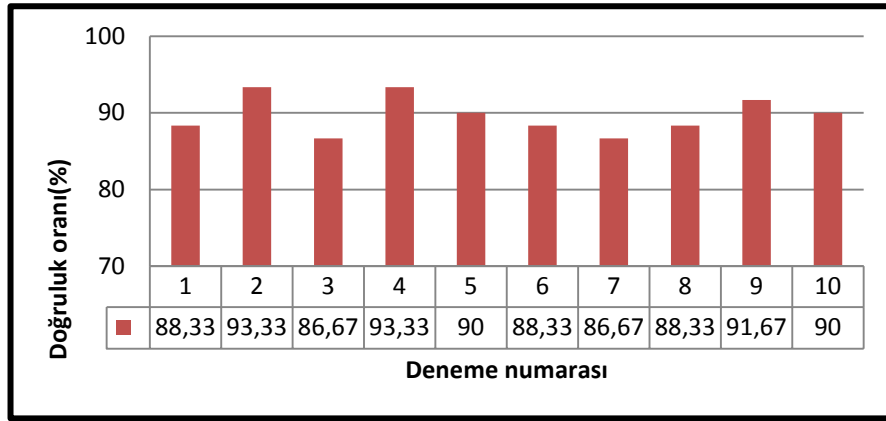
**Adım 15:** Bu adımda uygulanan durumlar ve sonuçları tablo 5.156. , tablo 5.157. ve şekil 5.30.'da verilmiştir.

**Tablo 5.156.** Yarasa algoritması Adım 15 için başlangıç koşulları

$f_{min}=0$ $f_{max}=1$
$A=0,5$ $r=0,4$
<u>Başlangıç popülasyonu fidan</u> <u>gelişim algoritması ile</u> <u>oluşturulmuştur</u>
$r=r+0,1*1,618$
$A=A-0,1*1,618$
$xyeni=BEST+(0,01*1,618*rand)$
$300<ENK$

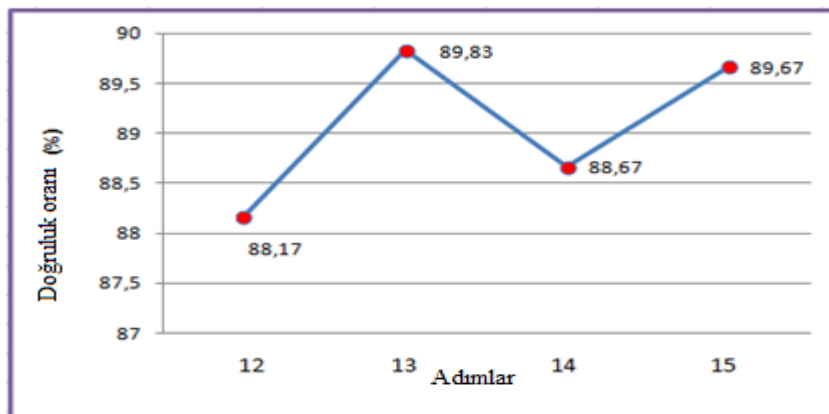
**Tablo 5.157.** Yarasa algoritması Adım 15 uygulama sonuçları

	BEST(1,1)	BEST(1,2)	BEST(1,3)	BEST(1,4)	Doğruluk oranı(%)
1	6,1071	3,0936	4,7568	1,1618	88,33
2	5,9147	3,137	4,3478	0,9746	93,33
3	5,7283	2,8048	3,8847	0,9734	86,67
4	5,9759	2,9944	4,0539	1,0284	93,33
5	6,0349	2,9784	3,8672	0,9262	90
6	5,4177	2,8416	4,4924	1,7225	88,33
7	5,7779	2,9697	4,1418	0,6531	86,67
8	5,5707	2,9576	4,0647	1,6624	88,33
9	6,1023	2,9906	3,8201	1,2324	91,67
10	5,8057	2,8052	4,2286	1,3073	90
					Ortalama= %89,67



**Şekil 5.30.** Yarasa algoritması Adım 15 için sonuç grafiği

12-15. adımlardan görüldüğü üzere BEST matrisinin değerleri etrafında oluşturulacak yeni bir çözüm matrisi için; r, A ve xyeni değerlerindeki değişme işleminde, altın oran değerini de göz önüne alınca, başarı oranının 13. adımda başarı ortalamalarında en yüksek değeri olan %89,83 ulaşılmıştır. Bu durum şekil 5.31.'de görülmektedir.



**Şekil 5.31.** Yarasa algoritması Adım 12-Adım 15 arası ortalama doğruluk oranları grafik temsili

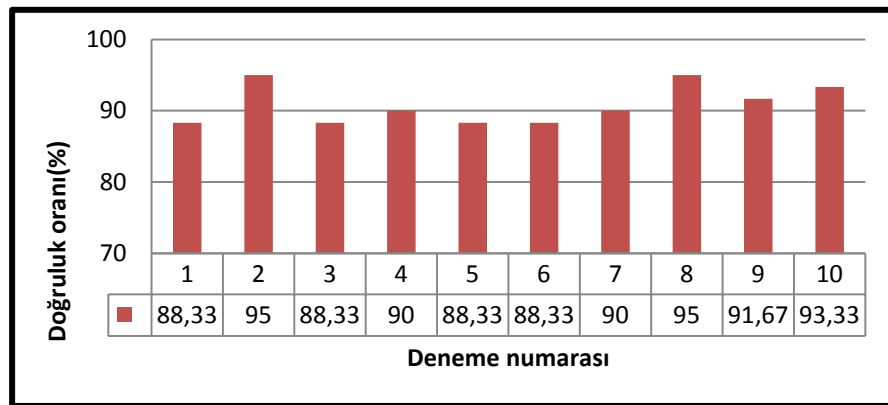
**Adım 16:** Bu adımda uygulanan durumlar ve sonuçları tablo 5.158. , tablo 5.159. ve şekil 5.32.'de verilmiştir.

**Tablo 5.158.** Yarasa algoritması Adım 16 için başlangıç koşulları

f_min=0 f_max=1
A= 0,5 r=0,618
<u>Başlangıç popülasyonu fidan gelişim algoritması ile oluşturulmuştur</u>
r=r+0,1*1,618
A=A-0,1*1,618
xyeni=BEST+(0,01*1,618*rand)
300<ENK

**Tablo 5.159.** Yarasa algoritması Adım 16 uygulama sonuçları

	BEST(1,1)	BEST(1,2)	BEST(1,3)	BEST(1,4)	Doğruluk oranı (%)
1	5,2829	3,0667	4,363	1,6953	88,33
2	5,8749	3,1651	3,9184	1,2643	95
3	5,9084	2,9719	4,5819	1,0401	88,33
4	5,7647	3,0994	4,4407	1,2353	90
5	5,3884	2,7975	4,5274	1,6292	88,33
6	5,6341	2,9616	4,4894	0,6855	88,33
7	6,2731	2,8858	4,1293	1,14	90
8	5,5428	2,9706	3,9167	1,4039	95
9	6,2124	2,6906	4,154	1,2768	91,67
10	5,4211	2,9926	4,3488	1,283	93,33
					Ortalama= 90,83



**Şekil 5.32.** Yarasa algoritması Adım 16 için sonuç grafiği

r parametresinin başlangıç değerini altın oran ile orantılı değiştirince, bu adımda ortalama başarı oranı bu adıma kadar olan en yüksek değerine çıkmıştır. Ayrıca tekil denemelerde % 95 başarı oranı 2 kez yakalanmıştır. r değeri [0,1] aralığında olduğundan altın oran değerinin ondalık kısmı seçilmiştir.

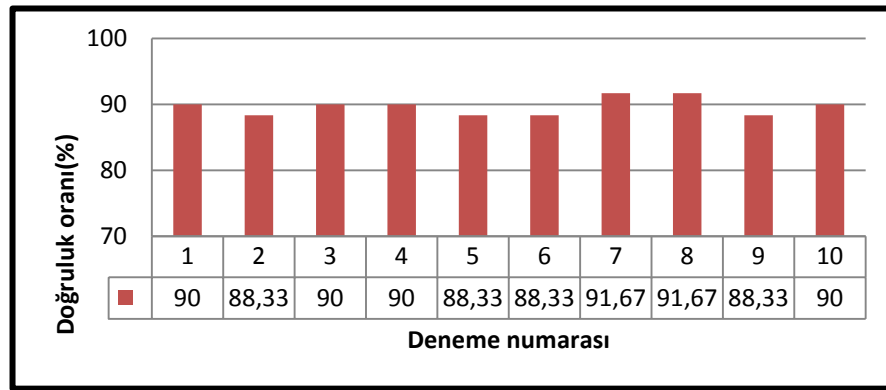
**Adım 17:** Bu adımda uygulanan durumlar ve sonuçları tablo 5.160. , tablo 5.161. ve şekil 5.33.'de verilmiştir.

**Tablo 5.160.** Yarasa algoritması Adım 17 için başlangıç koşulları

f_min=0 f_max=1
A= 0,5 r=0,618
<u>Başlangıç popülasyonu fidan gelişim algoritması ile oluşturulmuştur</u>
$r=r+abs(1-r*(1-exp(1,618)*rand))$
$A=A-0,1*1,618$
$xyeni=BEST+(0,01*1,618*rand)$
300<ENK

**Tablo 5.161.** Yarasa algoritması Adım 17 uygulama sonuçları

	BEST(1,1)	BEST(1,2)	BEST(1,3)	BEST(1,4)	Doğruluk oranı (%)
1	5,841	2,9865	4,3648	0,9383	90
2	5,8609	3,254	4,3969	1,1264	88,33
3	5,697	2,8358	4,2187	1,5552	90
4	5,4927	3,1416	4,6045	0,8552	90
5	5,555	3,2388	4,4079	0,8061	88,33
6	6,3556	3,0654	4,2564	1,4534	88,33
7	5,9791	2,9089	3,6522	1,3033	91,67
8	5,792	3,1715	4,007	1,4075	91,67
9	5,912	2,9649	3,7496	1,6866	88,33
10	5,9253	3,2869	4,278	1,1044	90
					Ortalama=89,67



**Şekil 5.33.** Yarasa algoritması Adım 17 için sonuç grafiği

Denklem (2.15)'de ifade edildiği üzere, r değerinin Yarasa algoritmasının orijinal halindeki eksponansiyel hali ile sonuçlar en iyiye yakın çıkmıştır. Ancak ayrı zamanlı bir uygulama olduğu için sürekli bir zaman denkleminin sonucu daha iyiye götürmediğini düşünebiliriz.

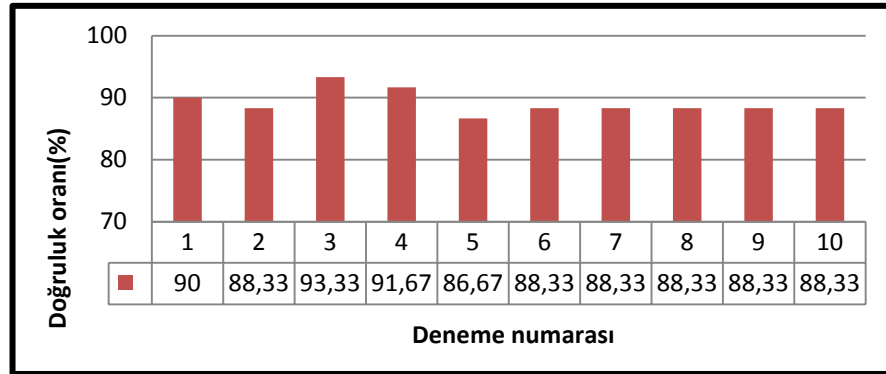
**Adım 18:** Bu adımda uygulanan durumlar ve sonuçları tablo 5.162. , tablo 5.163. ve şekil 5.34.'de verilmiştir.

**Tablo 5.162.** Yarasa algoritması Adım 18 için başlangıç koşulları

f_min=0 f_max=1
A= 0,618 r=0,618
<u>Başlangıç popülasyonu fidan gelişim algoritması ile oluşturulmuştur</u>
$r=r+abs(1-r*(1-exp(1,618)*rand))$
$A=A-0,1*1,618$
$xyeni=BEST+(0,01*1,618*rand)$
300<ENK

**Tablo 5.163.** Yarasa algoritması Adım 18 uygulama sonuçları

	BEST(1,1)	BEST(1,2)	BEST(1,3)	BEST(1,4)	Doğruluk oranı (%)
1	5,596	2,9694	4,3874	1,2439	90
2	5,9986	2,634	4,2814	1,1248	88,33
3	6,2612	2,7975	4,2278	1,1518	93,33
4	5,3369	2,8358	4,7165	1,4434	91,67
5	5,6028	3,0094	4,9073	1,2534	86,67
6	5,8484	3,3403	4,1067	1,4696	88,33
7	5,7902	2,8697	4,0407	0,9552	88,33
8	5,9101	2,9672	4,0816	1,6256	88,33
9	5,9273	2,8058	4,2598	1,0825	88,33
10	5,782	2,7925	3,7114	1,6781	88,33
					Ortalama=89,17

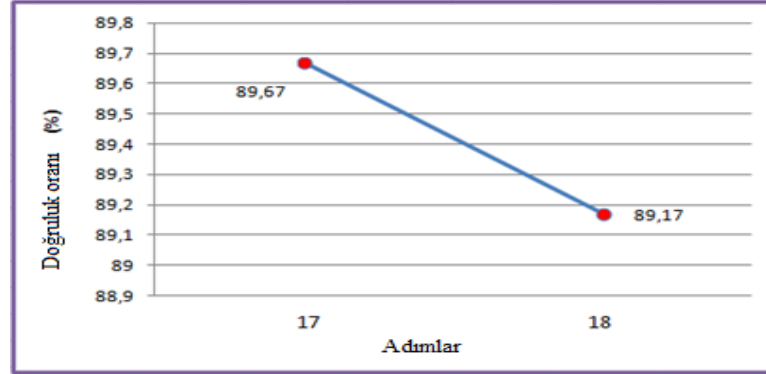


**Şekil 5.34.** Yarasa algoritması Adım 18 için sonuç grafiği

A değerinin başlangıç değerine altın oranla bağlantılı seçmek sonucu çok kötü bir aralığa çekmemiş, fakat yine de daha iyi bir değere doğru da götürmemiştir. Bu durum şekil 5.35.'de görülmektedir. A değerinin değişim oranının altın oranla iyi sonuçlar vermesi daha



anlamli bir sonuçtur. Yeni jenerasyonlara aktarilan yeni A deęerlerinin oluřurken deęerindeki azalıřı altın orana gre olunca daha anlamli olmaktadır.



řekil 5.35. Yarasa algoritması Adım 17 ve Adım 18 ortalama doęruluk oranları grafik temsili

**Adım 19:** Bu adımda uygulanan durumlar ve sonuçları tablo 5.164. , tablo 5.165. ve řekil 5.36.'da verilmiřtir.

**Tablo 5.164.** Yarasa algoritması Adım 19 iin bařlangı kořulları

f_min=0 f_max=1
A= 0,618 r=0,618
<u>Bařlangı poplasyonu fidan</u> <u>geliřim algoritması ile</u> <u>oluřturulmuřtur</u>
$r=r+abs(1-r*(1-exp(1,618)*rand))$
$A=A-0,1*1,618$
$xyeni=BEST+(0,01*1,618*rand)$
$f(i)=f\_min+(f\_max-f\_min)*0,618$
300<ENK

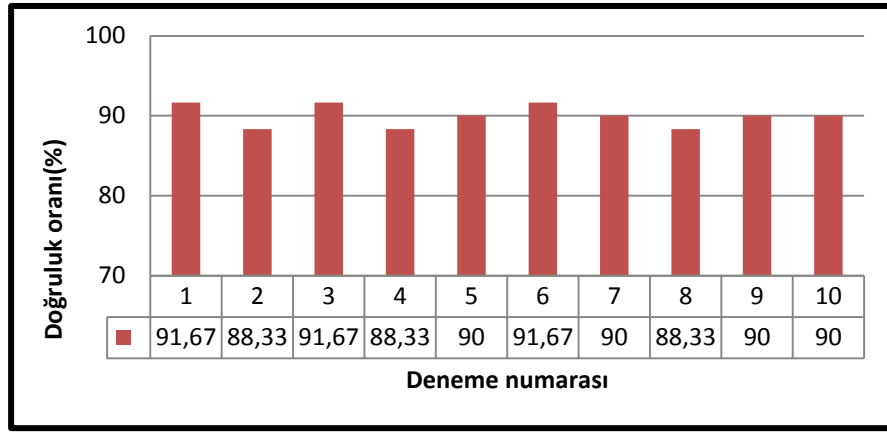
Bu adımda algoritmaya yeni bir altın oran denemesi daha eklenmiřtir. Yarasaların ıkarmıř oldukları seslerin frekanslarını temsil eden  $f_i$  deęerleri vardır.

Denklem (2.11)'de ifade edildięi gibi  $f_i = f_{min} + (f_{max} - f_{min}) * \beta$  forml ile hesaplanır.

Buradaki  $\beta$  parametresini řimdiye kadar rastgele olarak alırken bu adımdan sonra bu deęeri altın oran deęeri ile baęlantılı seeceęiz. Yine  $\beta$  deęeri [0,1] aralıęında deęiřtięinden; altın oranın ondalık kısmı olan 0,618 deęeri kullanılacaktır.

**Tablo 5.165.** Yarasa algoritması Adım 19 uygulama sonuçları

	BEST(1,1)	BEST(1,2)	BEST(1,3)	BEST(1,4)	Doğruluk oranı (%)
1	5,426	3,0577	4,3695	1,3408	91,67
2	5,6766	2,9372	4,8764	1,211	88,33
3	6,174	2,8975	4,092	1,0484	91,67
4	6,0473	2,6837	4,487	1,5554	88,33
5	5,548	3,2946	4,5733	1,1108	90
6	5,2889	3,0176	4,5654	1,3377	91,67
7	5,6102	3,2821	4,4739	1,2901	90
8	6,074	2,9328	3,6576	1,293	88,33
9	5,3256	2,9222	4,5487	1,3681	90
10	5,8335	3,1033	3,8981	1,5758	90
					Ortalama= 90



**Şekil 5.36.** Yarasa algoritması Adım 19 için sonuç grafiği

Şekil 5.36.'dan da görüldüğü üzere frekansları altın oran ile bağlantılı kullanmak başarı oranını arttırmıştır. Aynı zamanda her örnek deneme adımı için, başarı oranını % 90 civarına yaklaştırmaya başlamıştır. Diğer örneklerde ortalama başarı oranı %90 civarında olmasına rağmen grafikte sapmalar olmaktadır. Bu adımdan sonra tüm örnekler %90 civarında başarıyı yakalayabilmektedir.

Uygulama yazılımında yapılan altın oranla alakalı düzenlemeler ENK tolerans değeri 285 değerine yakın bir değere gerilemiştir. Bu adımdan sonraki 3 adımda tolerans değeri aşağı çekilerek bu değer etrafında denenmiştir.

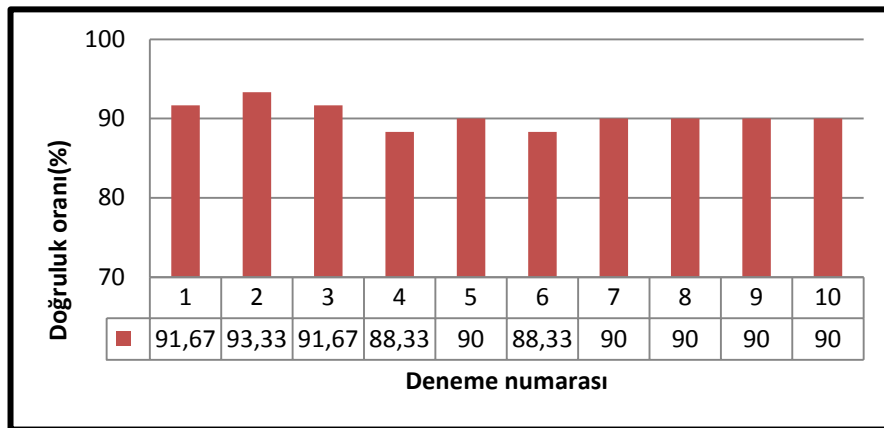
**Adım 20:** Bu adımda uygulanan durumlar ve sonuçları tablo 5.166. , tablo 5.167. ve şekil 5.37.'de verilmiştir.

**Tablo 5.166.** Yarasa algoritması Adım 20 için başlangıç koşulları

f_min=0 f_max=1
A= 0,618 r=0,618
<u>Başlangıç popülasyonu fidan gelişim algoritması ile oluşturulmuştur</u>
$r=r+abs(1-r*(1-exp(1,618)*rand))$
$A=A-0,1*1,618$
$xyeni=BEST+(0,01*1,618*rand)$
$f(i)=f\_min+(f\_max-f\_min)*0,618$
290<ENK

**Tablo 5.167.** Yarasa algoritması Adım 20 uygulama sonuçları

	BEST(1,1)	BEST(1,2)	BEST(1,3)	BEST(1,4)	Doğruluk oranı (%)
1	6,1889	3,0039	4,1274	1,3169	91,67
2	5,5297	3,0991	4,4943	1,0608	93,33
3	5,8227	3,1444	4,1355	1,1815	91,67
4	5,7906	3,1348	4,1558	1,5518	88,33
5	5,8106	2,7733	4,2331	1,438	90
6	5,7251	2,8877	3,7956	1,2325	88,33
7	5,7607	2,8326	4,297	1,2296	90
8	5,6755	3,0295	4,3892	1,1234	90
9	5,5895	2,8324	4,5372	1,2171	90
10	5,7291	2,9362	4,3432	1,229	90
					Ortalama= 90,33



**Şekil 5.37.** Yarasa algoritması Adım 20 için sonuç grafiği

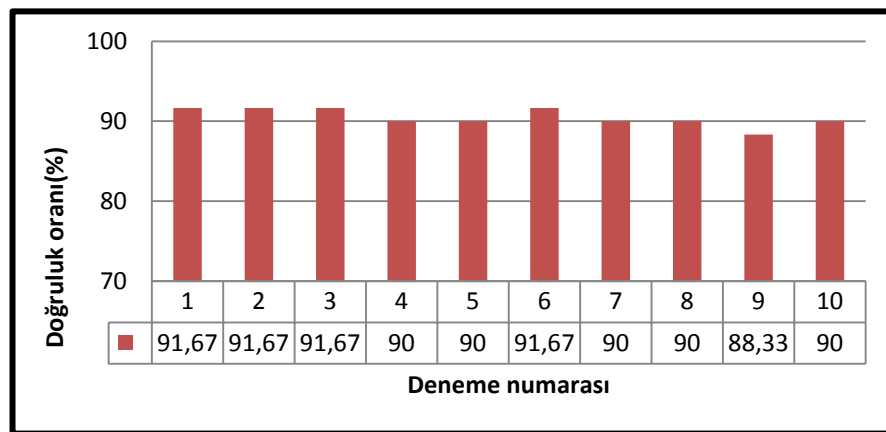
**Adım 21:** Bu adımda uygulanan durumlar ve sonuçları tablo 5.168. , tablo 5.169. ve şekil 5.38.'de verilmiştir.

**Tablo 5.168.** Yarasa algoritması Adım 21 için başlangıç koşulları

f_min=0 f_max=1
A= 0,618 r=0,618
<u>Başlangıç popülasyonu fidan gelişim algoritması ile oluşturulmuştur</u>
$r=r+abs(1-r*(1-exp(1,618)*rand))$
$A=A-0,1*1,618$
$xyeni=BEST+(0,01*1,618*rand)$
$f(i)=f\_min+(f\_max-f\_min)*0,618$
285<ENK

**Tablo 5.169.** Yarasa algoritması Adım 21 uygulama sonuçları

	BEST(1,1)	BEST(1,2)	BEST(1,3)	BEST(1,4)	Doğruluk oranı (%)
1	5,9856	2,9946	4,2025	1,2264	91,67
2	5,7827	3,1362	4,304	1,2985	91,67
3	5,9092	3,0494	4,3029	1,1467	91,67
4	5,694	2,9291	4,3471	1,4307	90
5	5,757	3,0038	4,2412	1,3525	90
6	5,839	3,0646	4,158	1,4057	91,67
7	5,6569	3,0113	4,2892	1,4506	90
8	5,9941	3,0035	4,4052	1,2266	90
9	5,7174	3,0745	4,3997	1,3741	88,33
10	5,8417	3,0017	4,1229	1,3825	90
					Ortalama= 90,50



**Şekil 5.38.** Yarasa algoritması Adım 21 için sonuç grafiği

Adım 21'deki tolerans değerinin aşağı çekilme işleminden sonra ortalama başarı oranında en yüksek değer olan % 90,50 değerine ulaşılmıştır. Aynı zamanda da hemen hemen tüm tekil denemeler %90 başarı bandına yerleşmiştir. Tek bir örnek %88,33 değerine sahiptir. Bunun dışındaki diğer denemelerin hepsinin sonucu %90 ve üzerinde değerler çıkmaktadır.

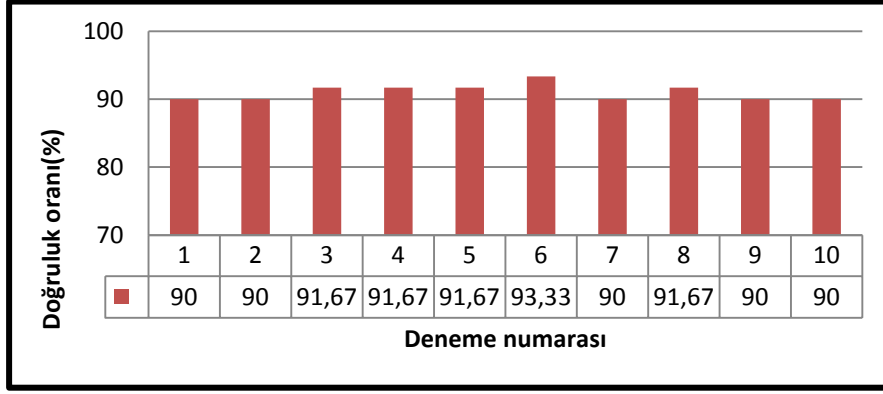
**Adım 22:** Bu adımda uygulanan durumlar ve sonuçları tablo 5.170. , tablo 5.171. ve şekil 5.39.'da verilmiştir.

**Tablo 5.170.** Yarasa algoritması Adım 22 için başlangıç koşulları

$f\_min=0$ $f\_max=1$
$A=0,5$ $r=0,618$
<u>Başlangıç popülasyonu fidan gelişim algoritması ile oluşturulmuştur</u>
$r=r+0,1*1,618$
$A=A-0,1*1,618$
$xyeni=BEST+(0,01*1,618*rand)$
$f(i)=f\_min+(f\_max-f\_min)*0,618$
$285<ENK$

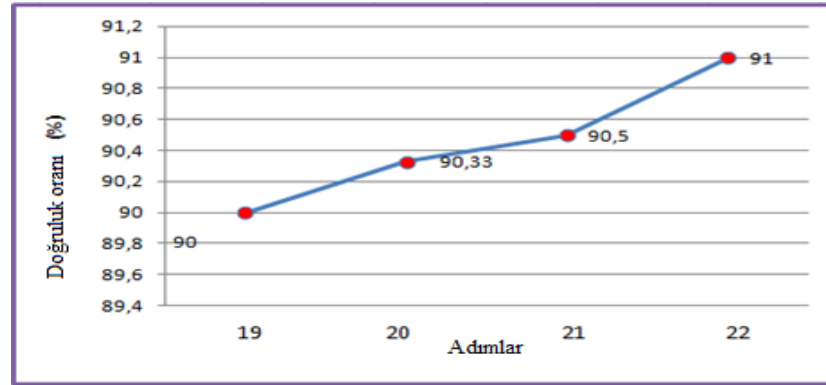
**Tablo 5.171.** Yarasa algoritması Adım 22 uygulama sonuçları

	BEST(1,1)	BEST(1,2)	BEST(1,3)	BEST(1,4)	Doğruluk oranı (%)
1	5,7456	3,0789	4,0648	1,1654	90
2	5,746	3,0249	4,1603	1,5099	90
3	5,643	2,993	4,3935	1,1911	91,67
4	5,7546	3,0762	4,3518	1,1857	91,67
5	5,6367	3,0914	4,2914	1,3184	91,67
6	5,7259	3,0183	4,2697	1,0886	93,33
7	5,7068	2,9533	4,2381	1,3655	90
8	5,5909	2,9885	4,1379	1,4198	91,67
9	5,6188	3,0078	4,5108	1,2686	90
10	5,8778	2,9626	4,2261	1,1361	90
					Ortalama= 91



Şekil 5.39. Yarasa algoritması Adım 22 için sonuç grafiği

Adım 21’de r değerinin artışı exponansiyel, Adım 22’de ise doğrusal olmuştur. Ayrık zamanlı bir uygulama olduğu için Adım 22’ de diğer şartlar aynı olmasına rağmen ortalama başarı oranı Adım 21’den daha yüksek çıkmıştır. Bu adımdaki tolerans değerinin aşağı çekilme işleminden sonra ortalama başarı oranında en yüksek değer olan % 91 değerine ulaşılmıştır. Aynı zamanda da tüm tekil denemeler %90 başarı bandının üzerine yerleşmiştir. Son 4 adımın doğruluk oranları şekil 5.40.’da görülmektedir.



Şekil 5.40. Yarasa algoritması Adım 19-Adım 22 arası ortalama doğruluk oranları grafik temsili

Uygulamanın bu bölümünde Yarasa algoritması, ayrık zamanlı veriler üzerinde kümeleme işlemini gerçekleştirmek suretiyle başarılı bir şekilde uygulanmıştır. Literatürde çoğunlukla sürekli zamanlı uygulamalar üzerinde yoğunlaşan Yarasa algoritması ile, ayrık zamanlı bir uygulama başarılı bir şekilde tamamlanmıştır.

Denemeler sonucunda Őu grlmŐtr ki; ayrık zamanlı bir uygulamada zellikle  $r$  sinyal yayma deęerinin deęiŐiminde srekli bir zamanlı ifade ile arttırma iŐlemi gerekleŐtirildięinde, baŐarı oranı dŐmektedir. Aynı  $r$  sinyal yayma deęerini basit bir cebirsel arttırım ile altın orana baęlı kalarak yaptığımızda baŐarı daha iyi deęerlere ıkmaktadır.

Ayrıca grlt oranı, sinyal yayma deęeri ve bunların gncellenmesinde kullanılan arttırım ya da azatlım deęerleri, her jenerasyonda  $BEST * f(i)$  etrafında oluŐturulup bir sonraki jenerasyona taŐınan yeni zm matrisinin arttırım deęeri ve frekans hesaplamasında kullanılan parametre deęeri gibi tm etkiler, altın oran deęerinde seildięinde doęruluk oranı her seferinde daha iyiye gitmektedir. Tm bu parametrelerde bir arttırım, azatlım ya da bir deęer ile arpma iŐlemi vardır. Bunu altın oran deęeri ile deęiŐtirmek, bu deęiŐim iŐlemini makul bir orana taŐımıŐtır. Gereęinden fazla arttırmak lokal maksimum deęerleri atlamaya sebep olabilir. Gereęinden az arttırmak ise bu lokal maksimum deęerlere ulaŐmayı ok geciktirebilir. Altın oran ile bu denemeleri yapmak doęruluk oranlarını iyi yerlere taŐımıŐtır.

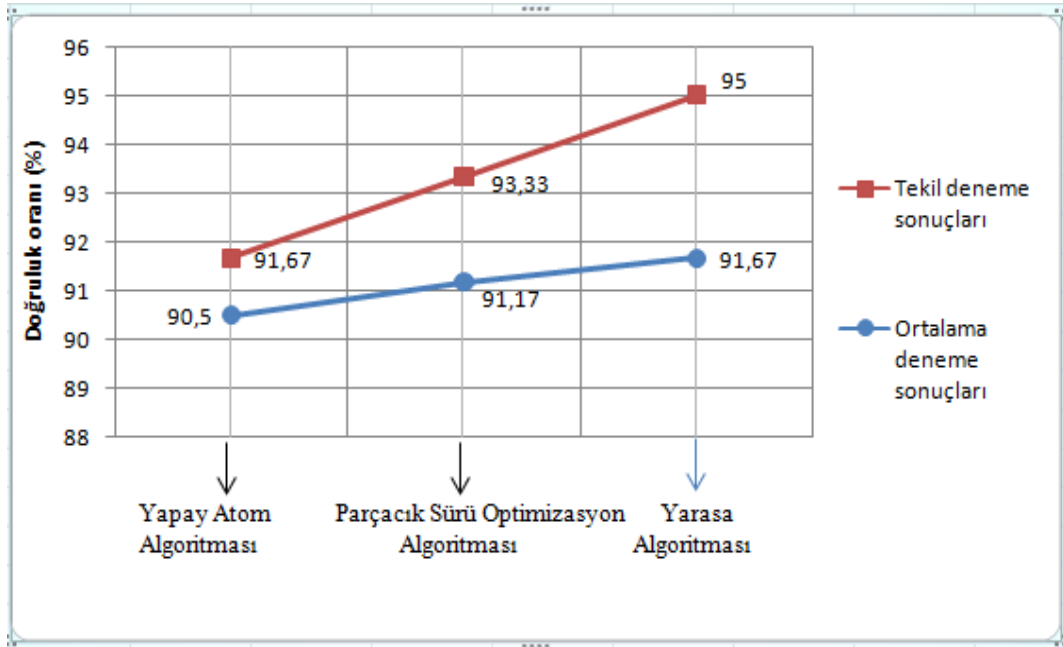
## 6. TARTIŞMA ve SONUÇ

Bu çalışmada dört farklı sezgisel yöntem olan, Yapay atom algoritması, Parçacık sürü optimizasyon algoritması, Ateş böceği algoritması ve Yarasa algoritması yöntemleri üzerinde veri kümeleme çalışılmıştır. Bu yöntemlerde altın oranının etkileri ortaya konulmuştur.

Yapay atom algoritması, Parçacık sürü optimizasyon algoritması ve Yarasa algoritması uygulama yazılımlarında ortak veri seti, Iris-Plants veri seti kullanılmıştır. Her üç yöntemin ortalama ve tekil sonuçlardaki en yüksek doğruluk oranları Tablo 6.1.'de verilmiştir.

**Tablo 6.1.** Iris-Plants veri setini kullanan uygulama yazılımlarının en yüksek sonuçlarının karşılaştırılması

	Ortalama başarı oranı (%)	Tekil denemelerde en yüksek doğruluk oranı (%)
Yapay atom algoritması	90,5	91,67
Parçacık sürü optimizasyon algoritması	91,17	93,33
Yarasa algoritması	91,67	95



**Şekil 6.1.** Iris-Plants veri setini kullanan uygulama yazılımlarının en yüksek sonuçlarının karşılaştırılma grafiği



Tablo 6.1. ve Şekil 6.1. incelendiğinde, aynı veri setini kullanan 3 farklı yöntemden, hem tekil denemeler için hem de ortalama denemeleri için; en yüksek doğruluk oranı Yarasa algoritmasına ait olduğu görülür. Daha sonraki yüksek başarı oranı Parçacık sürü optimizasyon algoritmasına aittir. Ayrıca tekil ve ortalama başarı oranlarının en yüksek olduğu bu durumlar kendi uygulama yazılımlarında altın oran denemelerinin olduğu durumlardadır.

Bu yöntemler içerisinde parametre sayısı en fazla olan yöntem Yarasa algoritması yöntemidir. Sonuçta bir çözümü etkileyen parametrelerin, çözüme olan etkilerinin ağırlıkları farklıdır. Yarasa algoritmasının parametre sayısının fazla olması ve bu parametrelerin sırayla altın orana yaklaştırılması, çok etkili olmuştur.

Yarasa algoritmasında da rastgele yeni çözüm üretilmektedir. Fakat üretilen bu yeni çözümün kabul edilmesi, uygunluk fonksiyonunun bir önceki çözümden iyi çıkmasına ve gürültü oranının rastgele üretilmiş olan bir değerden yüksek çıkmasına bağlıdır. Böylelikle çözüme dahil edilecek yeni değerler bir doğruluk kriterine göre çözüm popülasyonuna dahil edilmiş oluyorlar. Buda yeni eklenen çözümlerin tamamen bir rastgelelikten uzak olmasını ve daha tutarlı sonuçlar elde edilmesini sağlamıştır.

Literatürde özellikle Yarasa algoritması için sürekli zamanlı uygulamalar için uygulanabilirlikten bahsedilmektedir. Bu tez çalışması göstermiştir ki; ayrık zamanlı veriler içinde Yarasa algoritması uygulanabilmektedir. Doğruluk oranı da %95'e varan değerle gayet iyi çıkmaktadır.

Yarasa algoritmasında  $r_i$  değerindeki değişim bir sürekli zamanlı ifade ile sağlanmaktadır. Geliştirilen uygulama yazılımında veri kümele için ayrık zamanlı veriler üzerinde çalışıldığından uygulama yazılımında  $r$  (sinyal yayma) ve  $A$  (gürültü oranı) değerlerindeki artış ve azalış hem exponansiyel olan bu formül ile hem de cebirsel bir artış ve azalış ile denenmiştir. Aynı şartlar için cebirsel artış ve azalışın doğruluk oranlarının daha iyi olduğu görülmüştür. Buradan şöyle bir sonuca varılmıştır. Uygulama sürekli zamanlı veriler üzerinde olduğunda bu artış ve azalışlar exponansiyel olmalı; uygulama ayrık zamanlı veriler üzerinde olduğunda bu artış ve azalışlar cebirsel yapılmalıdır.

Ateş böceği algoritmasında farklı bir veri seti olan Wisconsin Diagnosis Breast Cancer Database veri seti ile çalışılmıştır. Amaç farklı bir veri seti üzerinde de altın oran ve başarı oranlarını test edebilmektir. Bu uygulama yazılımında da Adım 17'de parametrelerin altın oran değerinde olduğu durumda, ortalama %94,9 ve yine aynı durumun tekil denemelerde ise %96 ile en yüksek doğruluk oranına ulaşmıştır. Altın oran değeri uygulamadaki etkisini bu uygulama yazılımında da göstermiştir.

Sezgisel yöntemlerde BEST çözüm etrafında oluşturulan ve yeni jenerasyondaki çözümlere dahil edilen yeni çözüm adayları mevcuttur. Bunların oluşturulmasında, BEST çözüm belirli bir oranda bir katsayıyla çarpılır ya da belirli bir oranla çarpılıp tekrar BEST çözüm üzerine eklenir. Bu yolla yeni çözüm adayları elde edilir. Bu yöntemin kullandığı çözüme göre değişir. Bu tez çalışması göstermiştir ki; bu yeni çözümleri üretmede kullanılan tekniklerde kullanılan parametreleri ve altın oran etrafında tutmak doğruluk oranlarını hep iyiye taşımıştır. Parametrelerin [0,1] kapalı aralığında tanımlı olduğu durumlarda, çoğu zaman en yüksek doğruluk oranları, altın oranın ondalık kısmı olan 0,618 etrafında gerçekleşmiştir. [0,1] kapalı aralığının dışında tanımlı olan daha geniş aralıklarda ise doğruluk oranı, altın oranın 1,618 değeri ve etrafında iyi sonuçlar vermiştir.

Ayrıca sezgisel yöntemlerde, popülasyonun iyi çözüm adaylarının yeni nesillere aktarılması bilinen bir durumdur. Bu tez çalışmasının 5.1. Yapay atom algoritması bölümünde, kovalent ve iyonik çözümlerle bu aktarma işlemi yapılmaya çalışılmıştır. Kovalent çözümler aslında uygunluk değeri iyi çıkan çözümlerdir ve bir sonraki nesle aktarılmıştır. Fakat bu bölümdeki sonuçlar göstermiştir ki; sadece iyi olan çözümlerin oranını belirlemek, sonucu iyi bir yere taşımamıştır. Hatta burada altın orana göre popülasyonu oranlamakta sonucu değiştirmemiştir. Bölüm 5.2., 5.3. ve 5.4. göstermiştir ki; en iyi çözüm etrafında geliştirilen yeni çözümlerdeki hesapsal parametrelerle oynamak başarı oranını yükseltmekte daha etkili olmuştur. Zaten en iyi sonuçlarda parametrelerin altın oran veya altın orana yakın olduğu yerlerde kendini göstermiştir.

Ayrıca Fidan gelişim algoritması yönteminin başlangıç popülasyonunu oluşturmadaki düzenli yapısının etkileri, özellikle bölüm 5.3. ve 5.4.'de açıkça görülmüştür. Başlangıç popülasyonunun çözüm uzayını düzgün bir dağılımla oluşturmak, doğruluk oranını daha da yükseltmiştir. Bu yönüyle Fidan gelişim algoritmasının etkilerinin, ileride yapılacak olan çalışmalara ışık tutacağını düşünmekteyiz.

## KAYNAKLAR

- 1- İ. KAYMAZ, Optimizasyon Teknikleri, [http://194.27.49.11/makine/ikaymaz/optimizasyon/dosyalar/DERS\\_1\\_OPTIMIZASYONA\\_GIRIS.pdf](http://194.27.49.11/makine/ikaymaz/optimizasyon/dosyalar/DERS_1_OPTIMIZASYONA_GIRIS.pdf), 2014.
- 2- Murty, K. G., *Optimization Models For Decision Making*, vol. 1, Internet Edition, Chapter 1: Models for Decision Making, 2003, 1-18.
- 3- A. Akcayol, <http://w3.gazi.edu.tr/~akcayol/files/ZOL1Giris.pdf>, 2014.
- 4- S. Akyol ve B. Alataş, *Güncel Sürü Zekası Optimizasyon Algoritmaları*, Nevşehir Üniversitesi Fen Bilimleri Enstitü Dergisi 1, 2012, 36-50.
- 5- L. Lamberti, C. Pappalètere, *Weight optimization of skeletal structures with multi-point simulated annealing*, Computer Modelling in Engineering and Sciences, vol. 18, no. 3, 2007, pp. 183-221.
- 6- R.-E. Precup, R.-C. David, E. M. Petriu, S. Preitl, A. S. Paul, *Gravitational search algorithm-based tuning of fuzzy control systems with a reduced parametric sensitivity*, in Soft Computing in Industrial Applications, A. Gaspar-Cunha, R. Takahashi, G. Schaefer, and L. Costa, Eds., Springer-Verlag, Berlin, Heidelberg, Advances in Intelligent and Soft Computing, vol. 96, 2011, pp. 141-150.
- 7- S.I. Birbil, S.C. Fang, *An electromagnetism-like mechanism for global optimization*, Journal of Global Optimization, vol 25, 2003, pp. 263-282.
- 8- R. Özdağ, A. Karıcı, *The Application of Electromagnetism-like Algorithm for the Dynamic Deployment Problem in Wireless Sensor Networks*, in Proc. 2nd International Eurasain Conference on Mathematical Sciences and Applications, Sarajevo, Bosnia and Hersegovina, Aug. 26-29, 2013, pp. 199.
- 9- O.K. Erol, İ. Eksin, *A new optimization method: Big bang-big crunch*, Advances in Engineering Software, vol 37, no. 2, pp. 106-111, February 2006.
- 10- Xin-She Yang, *A New Metaheuristic Bat-Inspired Algorithm*, NISCO 2010, SCI 284, 2010, pp. 65-74.
- 11- Xin-She Yang and Amir H. Gandomi, *Bat Algorithm: A Novel Approach for Global Engineering Optimization*, Engineering Computations, Vol. 29, Issue 5, pp. 464-483, 2012.
- 12- Xin-She Yang, *Bat Algorithm for Multi-Objective Optimisation*, Int. J. Bio-Inspired Computation Vol 3 no. 5, 2011, pp. 267-274.
- 13- J.-T. Tsai, *Solving Japanese nongrams by Taguch-based genetic algorithm*, Applied Intelligence, vol 37, no. 3, 2012, pp. 405-419.
- 14- H. Xing, R. Qu, *A compact genetic algorithm for the network coding based resource minimization problem*, Applied Intelligence, vol 36, no. 4, 2011, pp. 809-823.
- 15- J. Rivero, D. Coadra, J. Calle, P. Isasi, *Using the ACO algorithm for path searches in social network*, Applied Intelligence, vol 36, no. 4, 2011, pp. 899-917.
- 16- D. Karaboğa, B. Baştürk, *A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm*, Journal of Global Optimization, vol 39, no. 3, 2007, pp. 459-471.
- 17- B. Akay, D. Karaboğa, *A Modified Artificial Bee Colony Algorithm for Real-Parameter Optimization*, Information Sciences, vol 192, no. 1, 2012, pp. 120-142.
- 18- L.N. De Castro, F.J. Von Zuben, *Learning and optimization using the clonal selection principle*, IEEE Transactions on Evolutionary Computation, vol 6, no. 3, 2002, pp. 239-251.
- 19- A. Karıcı, *Theory of saplings growing-up algorithm*, in Proc. ICANNGA-2007: Adaptive and Natural Computing Algorithms, Editors: Bartłomiej Beliczynski, Andrej Dzielinski, Marcin Iwanowski, Bernardete Ribeiro, LNCS, vol 4431, 2007., pp 450-460.
- 20- A. Karıcı, Bilal Alataş, *Thinking Capability of Saplings Growing Up Algorithm*, in Proc. IDEAL-2006: 7th International Conference on Intelligent Data Engineering and Automated Learning, LNCS, vol 4224, 2006, pp.386-393.

- 21- A. Karıcı, *Saplins Sowing and Growing up Algorithm Convergence Properties*, in Proc. INISTA-2007: International Symposium on Innovations in Intelligent Systems and Applications, Yıldız Technical University, İstanbul, 2007, pp. 322-326.
- 22- A. Karıcı, M. Yiğiter, M. Demir, *Natural Inspired Computational Intelligence Method: Saplings Growing Up Algorithm*, in Proc. Ikecco'2007 International Kyrgyz-Kazak Electronics And Computer Conference, Bishkek-Almaty, 2007, pp.1-8.
- 23- M. Demir, M. Yiğiter, A. Karıcı, *Application of Saplings Growing Up Algorithm to Clustering Medical Data*", in Proc. Ikecco'2007 International Kyrgyz-Kazak Electronics And Computer Conference, Bishkek-Almaty , 2007, pp.9-15.
- 24- M. Demir, A. Karıcı, M. Özdemir, *Fidan Gelişim Algoritması Yardımı ile DNA Motiflerinin Keşfi*, Çankaya University Journal of Science and Engineering, volume 8, no. 1, 2011, pp. 51-62.
- 25- A.R. Mehrabian, C. Lucas, *A novel numerical optimization algorithm inspired from weed colonization*, Ecological Informatics, vol 1, no. 4, 2006, pp. 355-366.
- 26- A. Mucherino, O. Seref, *Monkey search: A novel metaheuristics search for global optimization I. Continuous parameter optimization*, Evolutionary Computation, vol 953, no. 1, 2007, pp. 25-49.
- 27- K.M. Passino, *Biomimicity of bacterial foraging for distributed optimization and control*, IEEE Control Systems Magazine, vol 22, no. 3, 2002, pp. 52-67.
- 28- M. Canayaz, A. Karıcı, *A New Metaheuristic Cricket-Inspired Algorithm*, in Proc. 2nd International Eurasain Conference on Mathematical Sciences and Applications, Sarajevo, Bosnia and Hercegovina, Aug. 26-29, 2013, pp. 176.
- 29- E. Deniz Ülker, A. Haydar, *Comparing the Robustness of Evolutionary Algorithms on the Basis of Benchmark Functions* , Advances in Electrical and Computer Engineering, vol. 13, no. 2, 2013, pp. 59-64.
- 30- D. Niizuma, K. Yasuda, A. Ishigame, *Multi-point tabu search for traveling salesman problems*, IEEE Transactions on Electrical and Electronic Engineering, vol 1, no. 1, 2006, pp. 126-129.
- 31- E.A. Gargari, C. Lucas, *Imperialist competitive algorithm: An algorithm for optimization inspired by imperialistic competition* , in Proc. IEEE congress on evolutionary computation, Singapore, 2007, pp. 4661-4667.
- 32- B. Alataş, *ACROA: Artificial Chemical Reaction Optimization Algorithm for Global Optimization* , Expert Systems with Applications, vol 38, pp. 13170-13180, 2011.
- 33- A. Karıcı, *A new Metaheuristic Algorithm Based Chemical Process: Atom Algorithm* , in Proc. 1st International Eurasain Conference on Mathematical Sciences and Applications, Prishtine, Kosovo, Sep. 3-7, 2012, pp. 83-84.
- 34- A.Erdoğan Yıldırım, A. Karıcı, *Solutions of Travelling Salesman Problem Using Genetic Algorithm and Atom Algorithm*, in Proc. 2nd International Eurasain Conference on Mathematical Sciences and Applications, Sarajevo, Bosnia and Hercegovina, Aug. 26-29, 2013, pp. 134.
- 35- A. Karadoğan, A. Karıcı, *Artificial Atom Algorithm for Reinforcement Learning* , in Proc. 2nd International Eurasain Conference on Mathematical Sciences and Applications, Sarajevo, Bosnia and Hercegovina, Aug. 26-29, 2013, pp. 379.
- 36- X.-S. Yang, *Firefly algorithm, Levy flights and global optimization*, in Proc. Research and Development in Intelligent Systems XXVI (Eds M. Bramer, R. Ellis, M. Petridis), Springer London, 2010, pp. 209-18.
- 37- X.-S. Yang, *Firefly algorithms for multimodal optimization*, Stochastic Algorithms: Foundations and Applications, Lecture Notes in Computer Science, Springer-Verlag, Berlin, vol 5792, 2009, pp. 169-178.
- 38- X.-S. Yang, *Firefly algorithm, stochastic test functions and design optimisation*, International Journal of Bio-Inspired Computation , vol 2, no. 2, 2010, pp. 78-84.
- 39- X.-S. Yang, *Harmony Search as a Metaheuristic Algorithm*, Music-Inspired Harmony Search Algorithm: Theory and Applications, Studies in Computational Intelligence, Springer Berlin, vol. 191, 2009, pp. 1-14.

- 40- J. Kennedy, R.C. Eberhart, *Particle swarm optimization*, in Proc. of IEEE international conference on neural Networks, Australia, 1995, vol 4, pp. 1942-1948.
- 41- N. A. El-Hefnawy, *Solving Bi-level Problems Using Modified Particle Swarm Optimization Algorithm*, International Journal of Artificial Intelligence, vol. 12, no. 2, 2014, pp. 88-101.
- 42- Y. Ortakçı, Cevdet GÖLOĞLU, *Parçacık Sürü Optimizasyonu İle Küme Sayısının Belirlenmesi*, Akademik Bilişim'12 - XIV. Akademik Bilişim Konferansı, 1-3 Şubat 2012, Uşak Üniversitesi, Uşak, 335-341.
- 43- R. Eberhart, J. Kennedy, *A New Optimizer Using Particle Swarm Theory*, Sixth International Symposium on Micro Machines and Human Science, 1995, pp. 39-43.
- 44- M. Y. Özsağlam, M. Cunkaş, *Optimizasyon Problemlerinin Çözümü için Parçacık Sürü Optimizasyonu Algoritması*, Politeknik Dergisi, Cilt 11, sayı 4, 2008, 299-305.
- 45- S.C. Chu, P.W. Tsai, J.S. Pan, *Cat swarm optimization*, PRICAI 2006: Trends in Artificial Intelligence Lecture Notes in Computer Science, Volume 4099, 2006, pp 854-858.
- 46- K.S. Lee, Z.W. Geem, *A new metaheuristics algorithm for continues engineering optimization: Harmony search theory and practice*, Computer Methods in Applied Mechanics and Engineering, vol 194, no.e 36-38, 2005, pp. 3902-3933.
- 47- D. Karaboğa, *Yapay Zekâ Optimizasyon Algoritmaları*, Nobel Yayın Dağıtım, 2011.
- 48- H. ALP, *Yarasaların Özellikleri ve Yarasalarla Mücadele Yöntemleri*, Dicle Üniversitesi Veteriner Fakültesi Dergisi, 2009: 2 (4): 57-63.
- 49- G. Markowsky, *Misconceptions about the Golden Ratio*, The College Mathematics Journal, Vol. 23, No. 1, 1992, pp. 2-19.
- 50- C. Falbo, *The Golden Ratio: A Contrary Viewpoint*”, The College Mathematics Journal, vol 36, no. 2, 2005, pp. 123-134.
- 51- A. L. Goldberger, B.J. West, T. Dresselhaus, V. Bhargava, *Bronchial Asymmetry and Fibonacci Scaling.*, Experientia, vol. 41, issue 12, 1985, pp. 1537-1538.
- 52- <http://wellaware1.com/docs/ear/fibonaccipdf.pdf>, 2015.
- 53- M. S. Durmuş, S. İplikçi, *Veri Kümeleme Algoritmalarının Performansları Üzerine Karşılaştırmalı Bir Çalışma*, Akademik Bilişim'07 - IX. Akademik Bilişim Konferansı Bildirileri, Kütahya, 2007, pp:393-400.
- 54- A.K. Jain, M.N. Murty, P.J. Flynn, *Data Clustering: A Review*, ACM Computing Surveys, Vol. 31, No. 3, September 1999, pp:265-323.
- 55- Hartigan J. A., *Clustering Algorithms*, John Wiley & Sons Inc., ISBN 0-471-35645-X, NewYork, 1975.
- 56- Hösel V., Walcher S., *Clustering Techniques: A Brief Survey*, AMS Subject Classification, 62H30, 68T10, 62-07, Germany, 2000.
- 57- Jain A. K., Dubes R. C., *Algorithms for Clustering Data*, Prentice Hall, Englewood Cliffs, NJ, 1988.
- 58- H. Özkan, *K-Means Kümeleme ve K-NN Sınıflandırma Algoritmalarının Öğrenci Notları Ve Hastalık Verilerine Uygulanması*, İstanbul Teknik Üniversitesi Fen Edebiyat Fakültesi Matematik Mühendisliği Programı, Bitirme Ödevi, Ocak 2013.
- 59- S. Abudalfa and M. Mikki, *A Dynamic Linkage Clustering using KD-Tree*, The International Arab Journal of Information Technology, Vol. 10, No. 3, May 2013, pp. 283-289.
- 60- M.B. Pouyan, R. Yousefi, S. Ostadabbas, and M. Nourani, *A Hybrid Fuzzy-Firefly Approach for Rule-Based Classification*, in Proc. FLAIRS Conference, 2014, pp. 357-362.
- 61- G. Komarasamy, A. Wahi, *A New Algorithm for Selection of Better K value Using Modified Hill Climbing in K-Means Algorithm*, Journal of Theoretical and Applied Information Technology, Vol. 55, No. 3, September 2013, pp. 307-314.
- 62- K. Karteeika Pavan, A. Appa Rao, A.V. Dattatreya Rao, *An Automatic Clustering Technique for Optimal Clusters*, International journal of Computer Science Engineering and Applications, Vol., No.4, 2011, pp 133-144.

- 63- C. F. Eick , N. Zeidat, and R. Vilalta, *Using Representative-Based Clustering for Nearest Neighbor Dataset Editing*, in Proc. ICDM'04 Fourth IEEE International Conference 1-4 November 2004, pp. 375-378.
- 64- M. Swain, S. K. Dash, S. Dash and A. Mohapatra, *An Approach for Iris Plant Classification Using Neural Network*, International Journal on Soft Computing ( IJSC ) Vol.3, No.1, February 2012, pp. 79-89.
- 65- G. Liu , H. Huang , S. Wang , Z. Chen, *A Novel Spatial Clustering Analysis Method Using Bat Algorithm*, International Journal of Advancements in Computing Technology(IJACT), Vol. 4, No. 20, November 2012, pp. 561-571.
- 66- Y. Zheng, Y. Zhou, J. Qu, *An Improved PSO Clustering Algorithm with Entropy-based Fuzzy Clustering*, WSEAS Transactions on Computers, Vol. 14, 2015, pp. 88-96.
- 67- F. Camastra and A. Verri, *A Novel Kernel Method for Clustering*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 27, No. 5, May 2005, pp. 801-805.
- 68- S. Singh, S. Saini, M. Singh, *Cancer Detection Using Adaptive Neural Network*, International Journal of Advancements in Research & Technology, Volume 1, Issue 4, September-2012, pp. 1-5.
- 69- I. Muhic, *Fuzzy Analysis of Breast Cancer Disease using Fuzzy c-means and Pattern Recognition*, Southeast Europe Journal of Soft Computing, Vol. 2, No. 1, 2013, pp. 50-55.
- 70- K. Polat, S. Güneş, *Breast Cancer Diagnosis Using Least Square Support Vector Machine*, Digital Signal Processing, Vol. 17, No. 4, July 2007, pp. 694-701.
- 71- J. Addeh, A. Ebrahimzadeh, *Breast Cancer Recognition Using a Novel Hybrid Intelligent Method*, Journal of Medical Signals & Sensors, Vol. 2 , No. 2, Apr-Jun 2012, pp. 95-102.
- 72- T. Hassanzadeh, M. R. Meybodi, *A New Hybrid Approach for Data Clustering using Firefly Algorithm and K-means*, in Proc. Artificial Intelligence and Signal Processing (AISP), 2012 16th CSI International Symposium, 2-3 May 2012, pp. 7-11.
- 73- <http://archive.ics.uci.edu/ml/machine-learning-databases/iris/>, 2013.
- 74- <http://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/>, 2015.
- 75- Ponnappa Musikapun and Pupong Pongcharoen, *Solving Multi-Stage Multi-Machine Multi-Product Scheduling Problem Using Bat Algorithm*, 2nd International Conference on Management and Artificial Intelligence IPEDR Vol.35, Singapore, 2012, pp. 98-102.

## ÖZGEÇMİŞ

**Adı Soyadı:** Murat DEMİR

**Doğum Yeri ve Tarihi:** Ergani 1979

**Adres:** Muş Alparslan Üniversitesi Teknik Bilimler Meslek Yüksekokulu Müdürlüğü

**E-Posta:** m.demir@alparslan.edu.tr

**Lisans:**Fırat Üniversitesi Bilgisayar Mühendisliği

**Yüksek Lisans:**Fırat Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği

**Mesleki Deneyim:**Muş Alparslan Üniversitesi öğretim görevlisi 2002-devam ediyor.

### Yayın Listesi:

#### Uluslararası SCI-SCI Expanded Sınıfı Makaleler

- 1- A. KARCI, M. DEMİR, “Estimation of protein structures by classification of angles between a-carbons of amino acids based on artificial neural Networks”, Expert Systems with Applications, vol.36, pp.5541-5548, 2009.
- 2- M. DEMİR, A. KARCI, Implementational analyses of proteins and genes obtained from cancer patients, Turkish Journal of Electrical Engineering & Computer Sciences, Doi: 10.3906/elk-1402-100, September 2014.
- 3- M. DEMİR, A. KARCI, Data Clustering on Breast Cancer Data Using Firefly Algorithm with Golden Ratio Method, Advances in Electrical and Computer Engineering, Vol. 11, No. 2, pp. 75-84, 2015.

#### Ulusal Makaleler :

- 1- M.Demir, A.Karcı, M.Özdemir, “Fidan Gelişim Algoritması Yardımı ile DNA Motiflerinin Keşfi”, Çankaya University Journal of Science and Engineering, vol.8, pp:51-62, 2011.

#### Uluslar arası Bildiriler

- 1- A. KARCI, M. YİĞİTER, M. DEMİR, Natural Inspired Computational Intelligence Method:Saplins Growing Up Algorithm, Ikecco'2007 International Kyrgyz-Kazak Electronics And Computer Conference, 2007
- 2- M. DEMİR, M. YİĞİTER, A. KARCI, Application of Saplings Growing up Algorithm to Clustering Medical Data, Ikecco'2007 International Kyrgyz-Kazak Electronics and Computer Conference, 2007.
- 3- M. DEMİR, A. KARCI, Proteinlerin Peptit Bağ Açılarının Tahmininde Yapay Sinir Ağları Yönteminin Kullanılması, ELECO2006, BURSA, 2006.
- 4- M. DEMİR, A. KARCI, Veri Kümelemede Fidan Gelişim Algoritmasının Kullanılması, Elektrik Elektronik Bilgisayar Biyomedikal Mühendisliği 12. Ulusal Kongresi ve Sergisi, 2007 ESKİŞEHİR.
- 5- M.DEMİR, A.KARCI, Biyoinformatik Verilerin Uygulamalı Analizi, TOK-2013, Malatya, pp:507-512, 2013.