

KARADENİZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YAZILIM MÜHENDİSLİĞİ ANABİLİM DALI

METASEZGİSEL ARAMA ALGORİTMALARINDA YAKINSAMA
DENKLEMİ TASARIMI İÇİN YENİ BİR YÖNTEMİN GELİŞTİRİLMESİ

YÜKSEK LİSANS TEZİ

Mehmet KATI

ŞUBAT 2024

TRABZON



KARADENİZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YAZILIM MÜHENDİSLİĞİ ANABİLİM DALI

**METASEZGİSEL ARAMA ALGORİTMALARINDA YAKINSAMA
DENKLEMİ TASARIMI İÇİN YENİ BİR YÖNTEMİN GELİŞTİRİLMESİ**

Mehmet KATI

Karadeniz Teknik Üniversitesi Fen Bilimleri Enstitüsünde
"YAZILIM YÜKSEK MÜHENDİSİ"
Unvanı Verilmesi İçin Kabul Edilen Tezdir.

Tezin Enstitüye Verildiği Tarih : 28 / 12 / 2023

Tezin Savunma Tarihi : 02 / 02 / 2024

Tez Danışmanı : Prof. Dr. Hamdi Tolga KAHRAMAN

Trabzon 2024

ÖNSÖZ

“Metasezgisel Arama Algoritmalarında Yakınsama Denklemi Tasarımı İçin Yeni Bir Yöntemin Geliştirilmesi” isimli bu tez Karadeniz Teknik Üniversitesi Fen Bilimleri Enstitüsü Yazılım Mühendisliği Anabilim Dalı, Yüksek Lisans Programı’nda hazırlanmıştır.

Tez çalışmam boyunca hiçbir desteğini esirgemeyen ve bilim insanı olma yolundaki ilk adımlarımı atarken kıymetli bilgileriyle ve tecrübesiyle yolumu aydınlatan danışman hocam Sayın Prof. Dr. Hamdi Tolga KAHRAMAN’a teşekkürü borç bilirim.

Ayrıca, tez çalışmam sürecinde değerli dostum Şule Deniz ÖZKAYA'ya özel bir teşekkür sunmak isterim. Zorlu anlarda paylaştığı bilgi ve deneyimleri, motivasyonumu yüksek tutmamda önemli bir rol oynadı. Kendisiyle geçirdiğimiz zaman içinde, samimi destekleriyle bu çalışmanın başarılı bir şekilde tamamlanmasına katkı sağladı. Şule Deniz'e içten teşekkürlerimi sunarım.

Son olarak, doğduğum günden beri elimi hiç bırakmayan, hayatım boyunca aldığım tüm kararlarda bilgi ve tecrübelerinden faydalandığım ve her kararında arkamda olan, maddi ve manevi desteklerini hiçbir zaman esirgemeyen ve bugünlere gelmemde en büyük paya sahip olan annem Hatice KATI ve babam İzzet KATI'ya teşekkür eder, minnettarlığımı sunarım.

Bu tez çalışmasının bundan sonraki çalışmalara katkı sağlamasını temenni ederim.

Mehmet KATI

Trabzon 2024

TEZ ETİK BEYANNAMESİ

Yüksek lisans tezi olarak sunduğum “Metasezgisel Arama Algoritmalarında Yakınsama Denklemi Tasarımı İçin Yeni Bir Yöntemin Geliştirilmesi” başlıklı bu çalışmayı baştan sona kadar danışmanım Prof. Dr. Hamdi Tolga Kahraman’ın sorumluluğunda tamamladığımı, verileri/örnekleri kendim topladığımı, deneyleri/analizleri ilgili laboratuvarlarda yaptığımı/yaptırdığımı, başka kaynaklardan aldığım bilgileri metinde ve kaynakçada eksiksiz olarak gösterdiğimi, çalışma sürecinde bilimsel araştırma ve etik kurallara uygun olarak davrandığımı ve aksinin ortaya çıkması durumunda her türlü yasal sonucu kabul ettiğimi beyan ederim. 02/02/2024

Mehmet KATI

İÇİNDEKİLER

	<u>Sayfa No</u>
ÖNSÖZ.....	III
TEZ ETİK BEYANNAMESİ.....	IV
İÇİNDEKİLER.....	V
ÖZET	VII
ŞEKİLLER DİZİNİ	IX
TABLolar DİZİNİ.....	XV
SEMBOLLER VE KISALTMALAR DİZİNİ	XVI
1. GENEL BİLGİLER	18
1.1. Giriş.....	18
1.2. Metasezgisel Optimizasyonun Temelleri	21
1.3. Metasezgisel Algoritmaların Arama Süreci	23
1.3.1. Popülasyon Oluşturma Yöntemleri.....	26
1.3.2. Rehber Seçim Yöntemleri	32
1.3.3. Komşuluk Araması ve Çeşitlilik	38
1.3.4. Güncelleme Mekanizması Tasarımı.....	42
1.4. Diferansiyel Evrim Algoritması	43
1.4.1. Arama Süreci Yaşam Döngüsü	44
2. YAPILAN ÇALIŞMALAR.....	50
2.1. Önerilen Yöntem	50
2.1.1. Halka Arama Denklemlerinin Oluşturulması	55
2.1.2. Optimum Halka Eşik Değerinin Belirlenmesi	57
2.1.3. Optimum Zincirlerin Araştırılması	60
3. BULGULAR VE TARTIŞMA	61
3.1. Deneysel Çalışma Çevresi.....	61
3.1.1. Ayarlar ve Adillik	62
3.1.2. Kısıtsız Test Problemleri.....	62
3.1.3. Gerçek Dünya Kısıtlı Mühendislik Test Problemleri	63
3.1.4. İstatiksel Test Yöntemleri	65
3.2. Optimum Yüzde Hata Değişim Oranının Belirlenmesi	65
3.3. Optimum Zincirlerin Araştırılması.....	70

3.3.1. Çok Halkalı Optimum Zincir Araştırılması	70
3.3.2. İki Halkalı Optimum Zincir Araştırılması	86
3.3.3. İkili ve Çok Halkalı En İyi Durumların Karşılaştırılması.....	104
3.3.4. İkili ve Çok Halkalı En İyi Durumların Rakip Algoritmalarla Karşılaştırılması	104
3.4. Gerçek Dünya Kısıtlı Mühendislik Problemlerinden Elde Edilen Bulgular	113
4. SONUÇLAR.....	123
5. ÖNERİLER.....	124
6. KAYNAKLAR	125
7. EKLER.....	132
ÖZGEÇMİŞ	



Yüksek Lisans Tezi

ÖZET

METASEZGİSEL ARAMA ALGORİTMALARINDA YAKINSAMA DENKLEMİ TASARIMI İÇİN YENİ BİR YÖNTEMİN GELİŞTİRİLMESİ

Mehmet KATI

Karadeniz Teknik Üniversitesi
Fen Bilimleri Enstitüsü
Yazılım Mühendisliği Anabilim Dalı
Danışman: Prof. Dr. Hamdi Tolga KAHRAMAN
2024, 155 Sayfa

Bilgisayarların yaygın kullanımıyla birlikte optimizasyon problemlerini çözmek için doğadan esinlenen metasezgisel arama (MSA) algoritmaları popüler hale gelmiştir. MSA algoritmalarının performansı, özellikle komşuluk araması ve çeşitlilik görevlerindeki başarılarına bağlıdır. Ancak karmaşık problemlerin çözümünde MSA algoritmalarının karşılaştığı iki temel zorluk bulunmaktadır. Birincisi, çok modlu problemlerin arama uzaylarında sıkça bulunan yerel minimum tuzaklarına takılmasıdır. Temel zorluklardan ikincisi ise arama sürecinin sonunda yeterince global çözüme yaklaşamamalarıdır.

Bu çalışmada MSA algoritmalarındaki performansı artırmak için yeni bir yöntem olan "Yüzde Hata Değişimi Esaslı Yakınsama Denklemi Tasarlama Yöntemi " önerilmektedir. Bu yöntemde, rehber seçim yöntemleri kullanılarak yakınsama denklemlerinin farklı versiyonları ve halkalar tasarlanmaktadır. Tasarlanan halkaların skorları hesaplanarak, en yüksek skora sahip halkalardan yakınsama zincirleri oluşturulur. MSA algoritmalarında arama sürecinde hatanın yüzde değişimi takip edilmektedir. Hatanın değişimi belirlenen yüzde hata değişim oranının altında ise tasarlanan zincirler kullanılarak algoritmanın arama performansını iyileştirmeyi amaçlamaktadır. Önerilen yöntemin DE algoritması üzerindeki olumlu etkileri açıkça gözlemlenmiştir.

Anahtar Kelimeler: Yüzde Hata Değişimi, Halkalar, Zincirler, Metasezgisel Algoritmalar

MSc. Thesis

SUMMARY

DEVELOPMENT OF A NEW METHOD FOR APPROXIMATION EQUATION DESIGN IN METAHEURISTIC SEARCH ALGORITHMS

Mehmet KATI

Karadeniz Technical University
The Graduate School of Natural and Applied Sciences
Software Engineering Sciences Graduate Program
Supervisor: Prof. Dr. Hamdi Tolga KAHRAMAN
2024, 155 Pages

With the widespread use of computers, nature-inspired metaheuristic search (MSA) algorithms have become popular for solving optimization problems. The performance of MSA algorithms depends on their success, especially in neighborhood search and diversity tasks. However, there are two main challenges that MSA algorithms face when solving complex problems. The first main challenge is that multimodal problems often get stuck in local minimum traps in their search spaces. The second main difficulty is that they cannot sufficiently converge to the global solution at the end of the search process.

In this study, a new method, "Percentage Error Variation Based Convergence Equation Design Method" is proposed to improve the performance of MSA algorithms. In this method, different versions of convergence equations, rings, are designed by using guide selection methods. By calculating the scores of the designed rings, convergence chains are formed from the rings with the highest scores. In MSA algorithms, the percentage change of the error is monitored during the search process. If the change of the error is below the specified percentage error change rate, it is aimed to improve the search performance of the algorithm by using the designed chains. The positive effects of the proposed method on the DE algorithm have been clearly observed.

Key Words: Percent Error Change, Rings, Chains, Metaheuristic Algorithms

ŞEKİLLER DİZİNİ

	<u>Sayfa No</u>
Şekil 1. Çok modlu arama uzayında yerel ve global çözüm noktaları	26
Şekil 2. DE popülasyon bireyleri başlangıç noktaları	27
Şekil 3. DE algoritmasında bulunan en iyi çözüm	28
Şekil 4. DE popülasyon bireyleri başlangıç noktaları	29
Şekil 5. DE algoritmasında bulunan en iyi çözüm	29
Şekil 6. DE popülasyon bireyleri başlangıç noktaları	30
Şekil 7. DE algoritmasında bulunan en iyi çözüm	31
Şekil 8. Turnuva yöntemi 3'lü turnuva akışı	34
Şekil 9. $maxFE=1000$, $as=0.2$ ve frekans sayısı $f=1$ için w_{dFDB}	37
Şekil 10. $maxFE=1000$, $as=0.2$ ve frekans sayısı $f=10$ için w_{dFDB}	38
Şekil 11. DE popülasyon bireyleri başlangıç noktaları	40
Şekil 12. DE algoritması bulunan en iyi çözüm	40
Şekil 13. DE popülasyon bireyleri başlangıç noktaları	41
Şekil 14. DE algoritması bulunan en iyi çözüm	42
Şekil 15. Diferansiyel evrim algoritması temel basamakları.....	44
Şekil 16. Mevcut popülasyon	47
Şekil 17. Halkalardan zincir oluşturma	53
Şekil 18. Yöntem akış diyagramı	55
Şekil 19. Önerilen yöntemin akış diyagramı	59
Şekil 20. 0.01 HED değeri için halka skorları.....	66
Şekil 21. 0.02 HED değeri için halka skorları.....	66
Şekil 22. 0.05 HED değeri için halka skorları.....	66
Şekil 23. 0.1 HED değeri için halka skorları.....	67
Şekil 24. 0.2 HED değeri için halka skorları.....	67
Şekil 25. 0.5 HED değeri için halka skorları.....	67
Şekil 26. 1 HED değeri için halka skorları.....	68
Şekil 27. 2 HED değeri için halka skorları.....	68
Şekil 28. 5 HED değeri için halka skorları.....	68
Şekil 29. CEC20 10D 1.problem çok halkalı durumlar için Friedman analiz sonuçları.....	72

Şekil 30. CEC20 10D 2.problem çok halkalı durumlar için Friedman analiz sonuçları.....	72
Şekil 31. CEC20 10D 3.problem çok halkalı durumlar için Friedman analiz sonuçları.....	72
Şekil 32. CEC20 10D 4.problem çok halkalı durumlar için Friedman analiz sonuçları.....	73
Şekil 33. CEC20 10D 5.problem çok halkalı durumlar için Friedman analiz sonuçları.....	73
Şekil 34. CEC20 10D 6.problem çok halkalı durumlar için Friedman analiz sonuçları.....	73
Şekil 35. CEC20 10D 7.problem çok halkalı durumlar için Friedman analiz sonuçları.....	74
Şekil 36. CEC20 10D 8.problem çok halkalı durumlar için Friedman analiz sonuçları.....	74
Şekil 37. CEC20 10D 9.problem çok halkalı durumlar için Friedman analiz sonuçları.....	74
Şekil 38. CEC20 10D 10.problem çok halkalı durumlar için Friedman analiz sonuçları.....	75
Şekil 39. CEC20 30D 1.problem çok halkalı durumlar için Friedman analiz sonuçları.....	75
Şekil 40. CEC20 30D 2.problem çok halkalı durumlar için Friedman analiz sonuçları.....	75
Şekil 41. CEC20 30D 3.problem çok halkalı durumlar için Friedman analiz sonuçları.....	76
Şekil 42. CEC20 30D 4.problem çok halkalı durumlar için Friedman analiz sonuçları.....	76
Şekil 43. CEC20 30D 5.problem çok halkalı durumlar için Friedman analiz sonuçları.....	76
Şekil 44. CEC20 30D 6.problem çok halkalı durumlar için Friedman analiz sonuçları.....	77
Şekil 45. CEC20 30D 7.problem çok halkalı durumlar için Friedman analiz sonuçları.....	77
Şekil 46. CEC20 30D 8.problem çok halkalı durumlar için Friedman analiz sonuçları.....	77
Şekil 47. CEC20 30D 9.problem çok halkalı durumlar için Friedman analiz sonuçları.....	78
Şekil 48. CEC30 10D 10.problem çok halkalı durumlar için Friedman analiz sonuçları.....	78
Şekil 49. CEC20 50D 1.problem çok halkalı durumlar için Friedman analiz sonuçları.....	78

Şekil 50. CEC20 50D 2.problem çok halkalı durumlar için Friedman analiz sonuçları.....	79
Şekil 51. CEC20 50D 3.problem çok halkalı durumlar için Friedman analiz sonuçları.....	79
Şekil 52. CEC20 50D 4.problem çok halkalı durumlar için Friedman analiz sonuçları.....	79
Şekil 53. CEC50 10D 5.problem çok halkalı durumlar için Friedman analiz sonuçları.....	80
Şekil 54. CEC20 50D 6.problem çok halkalı durumlar için Friedman analiz sonuçları.....	80
Şekil 55. CEC20 50D 7.problem çok halkalı durumlar için Friedman analiz sonuçları.....	80
Şekil 56. CEC50 10D 8.problem çok halkalı durumlar için Friedman analiz sonuçları.....	81
Şekil 57. CEC20 50D 9.problem çok halkalı durumlar için Friedman analiz sonuçları.....	81
Şekil 58. CEC20 50D 10.problem çok halkalı durumlar için Friedman analiz sonuçları.....	81
Şekil 59. CEC20 100D 1.problem çok halkalı durumlar için Friedman analiz sonuçları.....	82
Şekil 60. CEC20 100D 2.problem çok halkalı durumlar için Friedman analiz sonuçları.....	82
Şekil 61. CEC20 100D 3.problem çok halkalı durumlar için Friedman analiz sonuçları.....	82
Şekil 62. CEC20 100D 4.problem çok halkalı durumlar için Friedman analiz sonuçları.....	83
Şekil 63. CEC20 100D 5.problem çok halkalı durumlar için Friedman analiz sonuçları.....	83
Şekil 64. CEC20 100D 6.problem çok halkalı durumlar için Friedman analiz sonuçları.....	83
Şekil 65. CEC20 100D 7.problem çok halkalı durumlar için Friedman analiz sonuçları.....	84
Şekil 66. CEC20 100D 8.problem çok halkalı durumlar için Friedman analiz sonuçları.....	84
Şekil 67. CEC20 100D 9.problem çok halkalı durumlar için Friedman analiz sonuçları.....	84
Şekil 68. CEC20 100D 10.problem çok halkalı durumlar için Friedman analiz sonuçları.....	85
Şekil 69. ÇHZ-Durum için Friedman analiz sonucu	86
Şekil 70. CEC20 10D 1.problem iki halkalı durumlar için Friedman analiz sonuçları	89

Şekil 71. CEC20 10D 2.problem iki halkalı durumlar için Friedman analiz sonuçları	90
Şekil 72. CEC20 10D 3.problem iki halkalı durumlar için Friedman analiz sonuçları	90
Şekil 73. CEC20 10D 4.problem iki halkalı durumlar için Friedman analiz sonuçları	90
Şekil 74. CEC20 10D 5.problem iki halkalı durumlar için Friedman analiz sonuçları	91
Şekil 75. CEC20 10D 6.problem iki halkalı durumlar için Friedman analiz sonuçları	91
Şekil 76. CEC20 10D 7.problem iki halkalı durumlar için Friedman analiz sonuçları	91
Şekil 77. CEC20 10D 8.problem iki halkalı durumlar için Friedman analiz sonuçları	92
Şekil 78. CEC20 10D 9.problem iki halkalı durumlar için Friedman analiz sonuçları	92
Şekil 79. CEC20 10D 10.problem iki halkalı durumlar için Friedman analiz sonuçları.....	92
Şekil 80. CEC20 30D 1.problem iki halkalı durumlar için Friedman analiz sonuçları	93
Şekil 81. CEC20 30D 2.problem iki halkalı durumlar için Friedman analiz sonuçları	93
Şekil 82. CEC20 30D 3.problem iki halkalı durumlar için Friedman analiz sonuçları	93
Şekil 83. CEC20 30D 4.problem iki halkalı durumlar için Friedman analiz sonuçları	94
Şekil 84. CEC20 30D 5.problem iki halkalı durumlar için Friedman analiz sonuçları	94
Şekil 85. CEC20 30D 6.problem iki halkalı durumlar için Friedman analiz sonuçları	94
Şekil 86. CEC20 30D 7.problem iki halkalı durumlar için Friedman analiz sonuçları	95
Şekil 87. CEC20 30D 8.problem iki halkalı durumlar için Friedman analiz sonuçları	95
Şekil 88. CEC20 30D 9.problem iki halkalı durumlar için Friedman analiz sonuçları	95
Şekil 89. CEC20 30D 10.problem iki halkalı durumlar için Friedman analiz sonuçları.....	96
Şekil 90. CEC20 50D 1.problem iki halkalı durumlar için Friedman analiz sonuçları	96

Şekil 91. CEC20 50D 2.problem iki halkalı durumlar için Friedman analiz sonuçları	96
Şekil 92. CEC20 50D 3.problem iki halkalı durumlar için Friedman analiz sonuçları	97
Şekil 93. CEC20 50D 4.problem iki halkalı durumlar için Friedman analiz sonuçları	97
Şekil 94. CEC20 50D 5.problem iki halkalı durumlar için Friedman analiz sonuçları	97
Şekil 95. CEC20 50D 6.problem iki halkalı durumlar için Friedman analiz sonuçları	98
Şekil 96. CEC20 50D 7.problem iki halkalı durumlar için Friedman analiz sonuçları	98
Şekil 97. CEC20 50D 8.problem iki halkalı durumlar için Friedman analiz sonuçları	98
Şekil 98. CEC20 50D 9.problem iki halkalı durumlar için Friedman analiz sonuçları	99
Şekil 99. CEC20 50D 10.problem iki halkalı durumlar için Friedman analiz sonuçları.....	99
Şekil 100. CEC20 100D 1.problem iki halkalı durumlar için Friedman analiz sonuçları.....	99
Şekil 101. CEC20 100D 2.problem iki halkalı durumlar için Friedman analiz sonuçları.....	100
Şekil 102. CEC20 100D 3.problem iki halkalı durumlar için Friedman analiz sonuçları.....	100
Şekil 103. CEC20 100D 4.problem iki halkalı durumlar için Friedman analiz sonuçları.....	100
Şekil 104. CEC20 100D 5.problem iki halkalı durumlar için Friedman analiz sonuçları.....	101
Şekil 105. CEC20 100D 6.problem iki halkalı durumlar için Friedman analiz sonuçları.....	101
Şekil 106. CEC20 100D 7.problem iki halkalı durumlar için Friedman analiz sonuçları.....	101
Şekil 107. CEC20 100D 8.problem iki halkalı durumlar için Friedman analiz sonuçları.....	102
Şekil 108. CEC20 100D 9.problem iki halkalı durumlar için Friedman analiz sonuçları.....	102
Şekil 109. CEC20 100D 10.problem iki halkalı durumlar için Friedman analiz sonuçları.....	102
Şekil 110. İHZ-Durum için Friedman analiz sonucu	103
Şekil 111. ÇHZ-Durum ve İHZ-Durum çalışmalarının Friedman analiz sonuçları	104

Şekil 112. CEC20 10D rakip algoritmalar Friedman analiz sonucu	108
Şekil 113. CEC20 20D rakip algoritmalar Friedman analiz sonucu	109
Şekil 114. CEC20 50D rakip algoritmalar Friedman analiz sonucu	110
Şekil 115. CEC20 100D rakip algoritmalar Friedman analiz sonucu	111
Şekil 116. CEC20 rakip algoritmalar ortalama boyut Friedman analiz sonucu	112
Şekil 117. Kısıtlı mühendislik 0.01 HED değeri için Friedman analiz sonucu.....	118
Şekil 118. Kısıtlı mühendislik 0.02 HED değeri için Friedman analiz sonucu.....	119
Şekil 119. Kısıtlı mühendislik 0.05 HED değeri için Friedman analiz sonucu.....	119
Şekil 120. Kısıtlı mühendislik 0.1 HED değeri için Friedman analiz sonucu.....	119
Şekil 121. Kısıtlı mühendislik 0.2 HED değeri için Friedman analiz sonucu.....	120
Şekil 122. Kısıtlı mühendislik 0.5 HED değeri için Friedman analiz sonucu.....	120
Şekil 123. Kısıtlı mühendislik 1 HED değeri için Friedman analiz sonucu.....	120
Şekil 124. Kısıtlı mühendislik 2 HED değeri için Friedman analiz sonucu.....	121
Şekil 125. Kısıtlı mühendislik 5 HED değeri için Friedman analiz sonucu.....	121
Şekil 126. En iyi durumlar için kısıtlı mühendislik Friedman analiz sonuçları	122

TABLÖLAR DİZİNİ

	<u>Sayfa No</u>
Tablo 1. Mutant vektörün oluşturulması	47
Tablo 2. Çaprazlama aşaması	48
Tablo 3. Oluşan yeni kromozom	49
Tablo 4. Oluşturulan halka denklemleri	56
Tablo 5. CEC20 problem seti	63
Tablo 6. Kısıtlı mühendislik problemleri	64
Tablo 7. HED değerleri için Wilcoxon testi ikili karşılaştırma sonuçları	69
Tablo 8. HED değerleri için Friedman analiz sonuçları	69
Tablo 9. 0.01 HED değeri için 10 ve 100 boyutlarında oluşturulan çok halkalı zincir durumları	70
Tablo 10. 0.02 HED değeri için 30 boyutunda oluşturulan çok halkalı zincir durumları	71
Tablo 11. 5 HED değeri için 50 boyutunda oluşturulan çok halkalı zincir durumları	71
Tablo 12. Boyut ve problem bazlı olarak ayarlanan çok halkalı optimum zincir	85
Tablo 13. DE algoritması ve ÇHZ-Durum Wilcoxon analiz sonucu	86
Tablo 14. İki halkalı zincir için oluşturulan durumlar	87
Tablo 15. Boyut ve problem bazlı olarak ayarlanan iki halkalı optimum zincir	103
Tablo 16. DE algoritması ve İHZ-Durum Wilcoxon analiz sonucu	103
Tablo 17. ÇHZ-Durum ve İHZ-Durum çalışmalarının Wilcoxon analiz sonuçları	104
Tablo 18. Kısıtlı mühendislik için oluşturulan iki halkalı durumlar	113
Tablo 19. En iyi durumlar için kısıtlı mühendislik Wilcoxon analiz sonuçları	122

SEMBOLLER VE KISALTMALAR DİZİNİ

HED	: Hata Eşik Değeri
MSA	: Metasezgisel Arama
MSO	: Metasezgisel Optimizasyon
GA	: Genetik Algoritma
GSA	: Yerçekimi Arama Algoritması
DE	: Diferansiyel Evrim
FDB	: Uygunluk-uzaklık Dengesi
dFDB	: Dinamik Uygunluk-uzaklık Dengesi
ABC	: Yapay Arı Kolonisi
SSA	: Salp Sürü Algoritması
GWO	: Gri Kurt Optimizasyon
SOGWO	: Seçmeli Karşıtlık Temelli Gri Kurt Optimizasyon
AFT	: Ali Baba ve Kırk Haramiler (Ali Baba and the Forty Thieves)
AVOA	: Afrika Akbabaları Optimizasyon Algoritması
GEO	: Altın Kartal Optimizasyonu
PPSO	: Fazör Parçacık Sürü Optimizasyonu
RIME	: Kırakı-buz Fenomeni Optimizasyonu
LSA	: Yıldırım Arama Algoritması
EMA	: Döviz Piyasası Algoritması
BMO	: Denizkestaneleri Çiftleşme Optimizasyonu
CKGSA	: Kaotik Kbest Yerçekimi Arama Algoritması
AO	: Şahin Optimizasyonu
PPA	: Parazitizm-predasyon Algoritması
HHO	: Harris Şahinleri Optimizasyonu
HHO_JOS	: Harris Şahinleri Ortak Karşı Seçim Optimizasyonu
HSSOGSA	: Hibrid Sperm Sürü ve Yerçekimi Arama Optimizasyonu
EDO	: Üstel Dağıtım Optimizasyonu
AOA	: Arşimet Optimizasyon Algoritması
SOA	: Martı Optimizasyon Algoritması
POA	: Bulmaca Optimizasyon Algoritması
MSA	: Güve Arama Algoritması

GJO	: Altın Çakal Optimizasyonu
LHHO	: Lider Harris Şahinleri Optimizasyonu
BWOA	: Karadul Optimizasyon Algoritması
WOA	: Balina Optimizasyon Algoritması
LSO	: Işık Spektrumu Optimizasyonu
MFO	: Ateşböceği Optimizasyonu
HGSO	: Henry Gazı Çözünürlüğü Optimizasyonu
MSGO	: Modifiye Edilmiş Sosyal Grup Optimizasyonu
CDDO	: Çocuk Çizim Gelişimi Optimizasyonu
SCA	: Sinüs Kosinüs Algoritması
LFD	: Levy Uçuş Dağılımı Algoritması
CHIMP	: Şempanze Optimizasyon Algoritması
BWO	: Karadul optimizasyonu
TSA	: Ağaç Tohum Algoritması
CRYSTAL	: Kristal Yapı Algoritması
FHO	: Ateş Şahini Optimizasyonu
RSO	: Sıçan Sürüsü Optimizasyonu
MAO:	: Meksika Axolotl Optimizasyonu
BOA	: Kelebek Optimizasyon Algoritması
TSO	: Geçici Arama Optimizasyonu
DDAO	: Dinamik Diferansiyel Dinlendirilmiş Optimizasyonu
SFO	: Ayçiçeği Optimizasyonu
RSA	: Sürünge Arama Algoritması
SAO	: Koku Ajanı Optimizasyonu
GPC	: Giza Piramitleri İnşası Optimizasyonu
CGSA	: Kaotik Kütle Sabitli Yerçekimi Arama Algoritması
DAOA	: Dinamik Aritmetik Optimizasyon Algoritması
CSA	: Karga Arama Algoritması
NCHHO	: Doğrusal Olmayan Tabanlı Kaotik Harris Şahinleri Optimizasyonu
CEC20	: IEEE CEC 2020
İHZ	: İki Halkalı Zincir
ÇHZ	: Çok Halkalı Zincir

1. GENEL BİLGİLER

Bu bölümde optimizasyondan bahsedilmiş olup, metasezgisel optimizasyonla ilgili temel bilgiler açıklanmıştır.

1.1. Giriş

Günlük yaşantımızda karşılaştığımız problemlere, farkında olarak ya da olmayarak her zaman en uygun olabilecek çözümler ararız. Bir başka deyişle, verilen şartlar altında en iyi sonucu elde etmeye çalışırız. Modelleri oluşturulmuş çeşitli problemlerin en uygun çözümünü aramak veya bulmak için karar değişkenleri ve kısıtlar altında bir araya getirilen yaklaşıma optimizasyon denmektedir. Optimizasyon, yıllar boyunca çeşitli yöntemlerle yapılsa da günümüzdeki şartlar ile daha hızlı ve verimli çözümler sunmaktadır. Gerçek yaşantımızdaki problemleri çözmek için farklı optimizasyon algoritmaları kullanılmaktadır. Bir amaç fonksiyonu minimize veya maksimize etmek için karar değişkenlerini belirlemek optimizasyonun temel amacıdır (Özbay & Özbay, 2021).

Çok sayıda değişken ve kısıtlama içeren problemleri çözmek oldukça zordur. Klasik optimizasyon yöntemleri, karmaşıklık düzeyi düşük olan, nispeten küçük arama uzaylarındaki basit optimizasyon problemleri için kabul edilebilir bir sürede etkili çözümler sunabilir. Ancak, gelişen teknoloji ile gerçek dünya mühendislik optimizasyon problemlerinin de karmaşıklık seviyesi artmaktadır. Özellikle, problemin tasarım parametre sayısı yani problemin boyutu arttıkça, arama uzayı da artmaktadır. Bu gibi sebeplerden dolayı klasik optimizasyon yöntemleri ile oluşturulan modellerin çözümlenmesi uzun zaman almaktadır. Üstelik çoğu kez problemler için küresel en iyi çözüme de ulaşamamaktadır. Bu durum geleneksel optimizasyon yöntemlerinin hem süre hem de çözümlerin kalitesi açısından yetersizliğini ortaya koymaktadır.

Problemi çözmek için geçen süre arttıkça ve bulunan çözümlerin kalitesi düştükçe, klasik optimizasyon yöntemlerinin uygulanabilirliği azalmaktadır. Bu durum araştırmacıları yeni optimizasyon yöntemlerini geliştirmeleri konusunda motive etmiştir (Pham & Karaboga, 2000). Klasik optimizasyon yöntemlerine alternatif olarak geliştirilen doğa tabanlı sezgisel algoritmalar, karmaşıklık düzeyi yüksek arama problemlerinin çözümlenmesinde yardımcı olmuştur. Sezgisel yaklaşımların geliştirilmesi, doğadakine

benzer şekilde probleme özgü arama yöntemlerinin önerilmesini sağlamıştır. Sezgisel arama yöntemleri sayesinde, arama uzayındaki yerel çözüm tuzaklarından kurtulmak ve makul bir süre içinde kabul edilebilir çözümler elde etmek mümkün olmaktadır. Arama uzayındaki yerel çözüm tuzaklarını aşmanın ve küresel en iyi çözümlere ulaşmanın yolu olarak komşuluk araması ve çeşitlilik yetenekleri ön plana çıkmaktadır.

Sezgisel yaklaşımların kullanıldığı alanları sadece optimizasyon problemleri ile sınırlandırmak doğru değildir. Sezgisel yaklaşımlar, sadece kestirim problemleri için küreel en iyi çözümü aramakla kalmayıp, tahmin, sınıflandırma ve kümeleme problemlerinin çözülmesinde kullanılan melez yapay zekâ algoritmalarının tasarlanması ve geliştirilmesinde de başarıyla uygulanmaktadırlar. Makine öğrenimi algoritmalarında k-en yakın komşu (Kahraman, Sagirolu, & Colak, 2013), karar ağaçları (Lai, Zhou, Hu, Bian, & Song, 2019; Pham & Triantaphyllou, 2009) ve bulanık mantık tabanlı algoritmaların (Kahraman H. T., 2016) melezleştirilmesinde, ayrıca son dönemde popüler hale gelen derin öğrenme (Fong, Yang, & Deb, 2018), büyük veri uygulamaları (Tayal & Singh, 2018) ve Endüstri 4.0 (Dosoglu, Kahraman, Sonmez, Guvenc, & Duman, 2018; Jian Zhang, Qin, & Fu, 2019) gibi modern otomasyon sistemlerinin geliştirilmesinde, metasezgisel optimizasyon teknikleri etkili bir şekilde kullanılmaktadır.

Metasezgisel arama (MSA) algoritmaları doğadaki farklı süreçlerden esinlenilerek geliştirilseler de ortak özelliklere sahiptirler. MSA algoritmalarında başlangıç işlemleri olarak gerçekleştirilen popülasyon (çözüm adayları topluluğu) oluşturma ve popülasyon üyelerinin (çözüm adaylarının) amaç fonksiyonu için performanslarını temsil eden uygunluk değerlerini bulma ortaktır. Benzer şekilde, MSA algoritmalarında arama süreci iteratiftir. Bu iteratif süreç ise ortak adımlardan oluşmaktadır. Bu ortak adımlar sırasıyla, arama sürecine rehberlik edecek olan çözüm adaylarının popülasyon içinden seçilmesi, arama uzayındaki referans konumların yakın komşuluğunda arama yapmak, arama uzayında amaç fonksiyonu değeri iyi olan yeni konumları keşfetmek ve popülasyon üyeleri için hayatta kalma mekanizmasının oluşturulmasıdır. Bu açıklamalara göre, metasezgisel arama algoritmalarında arama süreci yaşam döngüsü ortak adımlardan oluşmaktadır. Bu ortak adımların tasarımında kullanılan yöntemler ise metasezgisel arama algoritmalarının hem hızını hem de buldukları çözümlerin kalitesini belirlemektedir. Bu açıklamalara göre, metasezgisel arama algoritmalarının arama performansını etkileyen süreçlerden bahsedildi. Bu süreçlerin tasarlanmasında kullanılan öğeler sırasıyla, (i) popülasyon oluşturma yöntemleri, (ii) rehber seçim yöntemleri, (iii) yakınsama denklemleri ve

son olarak (iv) güncelleme mekanizması tasarım yöntemleridir. Bu açıklamalara göre, MSA algoritmaları üzerine yürütülen araştırmalar bu dört konu üzerinedir. Bu tez çalışmasından önce rehber seçim yöntemleri ve güncelleme mekanizması tasarımı üzerine araştırmalar yürüttük. Bu araştırmalar neticesinde, MSA algoritmalarında rehberlik mekanizmasının tasarımı için dinamik ve uyarlanır uygunluk-uzaklık mesafesi yöntemlerini (Kahraman, Bakır, Duman, Katı ve & Aras, 2022; Duman, Kahraman & Katı, 2023) ve MSA algoritmalarında güncelleme mekanizmasının tasarımı için doğal hayatta kalma yöntemini (Kahraman, Katı, Aras & TAŞCI, 2023) geliştirdik.

Bu tez çalışmasında, MSA algoritmalarında yakınsama denklemlerinin tasarımı üzerine bir araştırma yürütülmektedir. Araştırmanın yürütülmesindeki başlıca motivasyon MSA algoritmalarında yakınsama denklemlerinin tasarımının belirsizlikler içermesi ve bu konuda büyük bir araştırma boşluğunun bulunmasıdır. MSA algoritmalarında yakınsama denklemlerinin tasarımı için halihazırda literatürde yaygın olarak kullanılan sistematik bir yönteme rastlanılmamıştır. Yakınsama denklemlerinin tasarım süreci MSA algoritmalarının tasarlanmasında ilham alınan doğal yöntemin işleyişine dayandırılmaktadır. Buna göre, yakınsama denklemi tasarımında MSA algoritmasının ilham kaynağı olan doğal yöntemin işleyişinin analiz edilmesi ve bu sürecin matematiksel olarak modellenmesi söz konusudur. Bununla birlikte, bu sürecin sonucunda elde edilen matematiksel modelin doğal süreci ne kadar temsil ettiği bilinmemektedir. Çoğu kez bunun bir önemi de olmamaktadır. Çünkü MSA algoritmasından beklenen başlıca yetenek optimizasyon problemleri için küresel en iyi çözümlere etkili bir şekilde yakınsayabilmesidir. Bu açıklamalara göre, yakınsama denklemi oluşturma sürecinin başarısı, MSA algoritmasının optimizasyon problemleri üzerindeki performansına bağlı olarak değerlendirilmektedir. Bununla birlikte, yakınsama denkleminin tasarımında kullanılan sistematik bir yöntem olmadığı gibi tasarlanan denklemlerin ne kadar başarılı olduğu da belirsizdir. Bu belirsizliğin üstesinden gelebilmek için bu tez çalışmasında MSA algoritmalarında yakınsama denklemlerinin tasarımı için sistematik bir yöntem olarak optimum zincir yaklaşımı tanıtılmaktadır. Optimum zincir yaklaşımının geliştirilmesi ve test edilmesi ile ilgili bilgiler sırasıyla “2. Yapılan Çalışmalar” ve “3. Bulgular ve Tartışma” bölümlerinde açıklanmaktadır. Tezin takip eden bölümlerinde metasezgisel optimizasyonun temelleri açıklanmaktadır.

1.2. Metasezgisel Optimizasyonun Temelleri

Metasezgisel optimizasyon (MSO), optimizasyon problemlerinin sezgisel arama algoritmaları kullanılarak çözümlenmesidir. Optimizasyon problemleri amaç fonksiyon, kısıt fonksiyonları, tasarım parametreleri ve tasarım parametrelerinin değer uzayından oluşmaktadır. Bir optimizasyon probleminin tanımlanabilmesi için asgari şart amaç fonksiyonuna ve tasarım parametrelerine ihtiyaç duyulmasıdır. Tasarım parametrelerinin alabilecekleri değerlerin uzayı problemin arama uzayını ifade eder. Tasarım parametrelerinin arama uzayında alabileceği değerleri sınırlandıran özel koşullar olması durumunda bu koşulları temsil eden kısıt fonksiyonları tanımlanır. nümerik tipli tasarım parametrelerinden oluşan bir optimizasyon probleminde kısıtların tanımlanmasında eşitlik ve/veya eşitsizlik kısıtlarına ihtiyaç duyulabilir. Bu açıklamalara göre, sürekli değerli arama uzayına ve kısıtlara sahip bir optimizasyon probleminin matematiksel ifadesi Denklem 1’de verildiği gibi tanımlanır (Kahraman, Aras, & Gedikli, 2020).

$$\begin{aligned} & \underset{x \in \mathbb{R}^n}{\text{minimize / maximize}} A = f(x_1, x_2, \dots, x_m) \\ & H_j(x) = 0, (j = 1, 2, \dots, J), \\ & G_k(x) \leq 0, (k = 1, 2, \dots, K) \end{aligned} \quad (1)$$

Denklem 1 incelendiğinde, m -adet tasarım parametresi $X \equiv [x_1, x_2, \dots, x_m]$ vektörü ile temsil edilmekte olup, $m \in \mathbb{N}^+, \forall_{i=1}^m x_i \in X$ dır. Amaç fonksiyon A ile temsil edilmektedir. J -adet eşitlik kısıtları $H_j(x)$ ile ve K -adet eşitsizlik kısıtları $G_k(x)$ ile temsil edilmektedir. Bu açıklamalara göre, Eşitlik T1’de verilen optimizasyon problemi için m -adet tasarım parametrelerinin alabilecekleri değerlerin uzayı (sınırları) $[X1_alt_sınır, X1_üst_sınır], [X2_alt_sınır, X2_üst_sınır], \dots, [Xm_alt_sınır, Xm_üst_sınır]$ olarak tanımlanır. Tek amaç fonksiyonundan oluşan bu optimizasyon problem için $H_j(x)$ eşitlik ve $G_k(x)$ eşitsizlik kısıtlarını karşılayan ve A amaç fonksiyonunu minimum/maksimum yapan bir adet küresel en iyi çözüm vardır. Optimizasyon algoritmalarının amacı probleme ait arama uzayındaki küresel en iyi çözümü bulmaktır. Küresel çözümün bulunamadığı durumlarda ise, arama uzayında ona en yakın olan yerel çözümler arasından bir vektör problemin çözümü olarak kabul edilir. Gerçek dünya optimizasyon problemlerinde çoğu kez, global çözüm bilinmemektedir. Bu durumda, mevcut çözüm adayları arasında kısıtları sağlayan ve amaç fonksiyonu en iyileyen (minimizasyon problemlerinde amaç fonksiyon değerini asgari yapan ve maksimizasyon problemlerinde ise amaç fonksiyon değerini azami yapan durum) tasarım değişkenleri

problemin çözümü olarak kabul edilmektedir. MSA algoritmaları bu süreçte, Denklem 1’de verilen problem için en iyi çözümü bulmaya çalışmaktadırlar.

MSA algoritmaları, topluluk ya da sürü zekasına dayalı olarak arama yapan algoritmalarıdır. Çözüm adayları tarafından oluşturulan bu topluluk MSA literatüründe popülasyon olarak adlandırılmaktadır. Esasında popülasyon terimi genetik algorithmadan literatüre yayılmıştır. Genetik algorithma popülasyonlar bireylerden oluşmakta ve her bir birey problem için çözümü temsil eden kromozom olarak adlandırılmaktadır. Kromozom ise genlerden oluşmaktadır. Genetik algorithma genler optimizasyon probleminin tasarım parametrelerine karşılık gelmektedir. Bu açıklamalara göre, bir MSA algoritmasında n -adet çözüm adayından (bireyden) oluşan çözüm adayları topluluğu P vektörü $\langle p_1, p_2, p_3, \dots, p_n \rangle$ Denklem 2’de verildiği gibi temsil edilir.

$$P \equiv \begin{bmatrix} p_1 \\ \vdots \\ p_n \end{bmatrix} \equiv \begin{bmatrix} x_{11} & \cdots & x_{1m} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{nm} \end{bmatrix}_{n \times m} \quad (2)$$

P ’deki her bir birey Denklem 1’de verilen optimizasyon problemi için tanımlanan kısıtları sağlayan arama uzayındaki bir konumu temsil etmektedir. Bu açıklamaya göre, P -deki bireylerin çözüm adayı olarak kabul edilebilmeleri için problemin kısıtlarını sağlamaları şarttır. Kısıtları sağlayan çözüm adaylarının probleme ait olan amaç fonksiyonu için sahip olduğu değer o çözüm adayının problem için uygunluğunu temsil eder. Bu açıklamaya göre, P ’deki çözüm adaylarının uygunluk değerleri Denklem 1’deki A-amaç fonksiyonunda değerlendirilerek hesaplanır. P ’deki n -adet çözüm adayının uygunluk değeri F -matrisi ile Denklem 3’te verildiği gibi gösterilir.

$$F \equiv \begin{bmatrix} f_1 \\ \vdots \\ f_n \end{bmatrix}_{n \times 1} \quad (3)$$

Çözüm adaylarının amaç fonksiyon için uygunluk parametrelerinin değerleri hesaplandıktan sonra, normalize edilmiş uygunluk değeri hesaplanır. Uygunluk değerlerinin normalize edilmesinde Denklem 4 kullanılır.

$$\begin{aligned}
n \in N^+, \forall_{i=1}^n, \quad A_i &= f_A(P_i), \quad \infty < A_i < +\infty \\
normA_i &= \frac{A_i - A_{min}}{A_{max} - A_{min}} \\
f_i &\equiv \begin{cases} \text{if } (A \text{ is minimizasyon problemi}) \rightarrow f_i = 1 - normA_i \\ \text{else } (A \text{ is maksimizasyon problemi}) \rightarrow f_i = normA_i \end{cases}
\end{aligned} \tag{4}$$

Normalizasyon işleminin amacı, çeşitli skor hesaplamalarında kullanılan parametrelerin birbirlerine baskınlık kurmasını önlemektir. Örneğin arama sürecinde rehberlik yapacak çözüm adaylarını belirlerken popülasyondaki tüm çözüm adaylarının uygunluk değerleri ve karar uzayında birbirlerine uzaklık değerleri hesaplanır. Bu iki parametre kullanılarak skor hesabı gerçekleştirilir. Uygunluk ve uzaklık değerleri arasında baskınlık olmaması için normalize edilmiş değerler kullanılır.

1.3. Metasezgisel Algoritmaların Arama Süreci

MSA algoritmaları optimizasyon problemlerinin çözümlenmesinde kullanılan, doğa tabanlı geliştirilmiş yöntemlerdir. 1950'li yıllara dayanan araştırmalar, MSA yöntemlerinin temelini atmıştır. Genetik Algoritma (GA), bu geliştirme çabalarının önemli bir dönüm noktası olarak kabul edilir. Michigan Üniversitesi'nden Prof. John Holland ve öğrencileri tarafından ortaya konan bu algoritma, evrimsel algoritmaların öncülerinden biridir ve MSA alanındaki gelişmelere yeni bir perspektif kazandırmıştır (Holland, 1992). Bu algoritmalar, gelişmiş çalışma ortamında etkili arama yöntemleri kullanarak, çözüm uzayındaki en iyi çözüme optimum bir şekilde erişim sağlamaktadır.

MSA algoritmalarının geliştirilme sürecinde bilim insanları, doğadaki fiziksel, biyolojik ve kimyasal olaylardan, hatta doğadaki canlıların yaşam biçimlerinden ilham almışlardır. Elektromanyetik kuvvet, eylemsizlik kuvveti ve yerçekimi gibi fiziksel kanunlar, çeşitli yöntemlerin önerilmesinde temel alınan unsurlardır. Doğadaki fizik kanunlarından esinlenerek geliştirilen algoritmalarından, en bilinenlerinden birkaçına Büyük-Patlama (Erol & Eksin, 2006), Kara Delik (Hatamlou, 2013), Merkezi Kuvvet Optimizasyon Algoritması (Formato, 2009) ve Yer Çekimi Arama Algoritması (GSA) (Rashedi, Nezamabadi-Pour, & Saryazdi, 2009) örnek olarak verilebilir. Metasezgisel yöntemlerin önerilmesinde etkili olan bir diğer önemli esin kaynağı da doğadaki canlıların sürü halindeki yaşam biçimlerinden alınan ilhamdır. Parçacık Sürü Optimizasyonu (Kennedy & Eberhart, 1995), Karınca Koloni Optimizasyonu (Dorigo, Birattari, & Stutzle, 2006), Yapay Arı Kolonisi (Karaboga & Basturk, 2007), Ateş Böceği Algoritması (Yang, 2010) gibi

algoritmalar sürü tabanlı metasezgisel yöntemlere örnek olarak sunulabilir. Evrimsel temelli algoritmalar ise Genetik Algoritma, Diferansiyel Evrim (Storn, 1995) ve Evrim Stratejisi (Beyer & Schwefel, 2002) en yaygın kullanılanlardır.

Metasezgisel arama algoritmalarının tasarımları incelendiğinde, Algoritma 1’de açıklandığı gibi ortak adımlardan oluştukları görülmektedir.

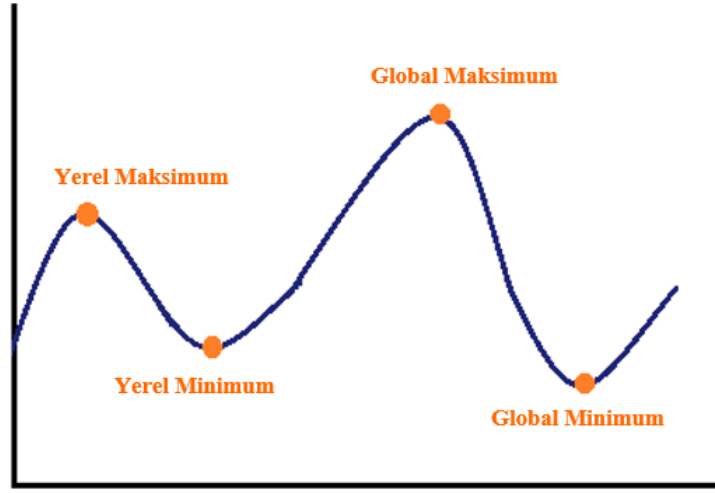
Algoritma 1. MSA algoritmalarının genel adımları (Kahraman, Aras, & Gedikli, 2020)

1. **Başlangıç İşlemleri**
 2. P : P ’i Denklem 2’de verildiği gibi n -adet çözüm adayı ile oluştur.
 3. **for** $i=1: n$
 4. F : P ’deki bireylerin uygunluk değerlerini hesapla
 5. **end**
 6. **while** (arama süreci yaşam döngüsü)
 7. Adım 1: Rehber seçim süreci
Arama sürecine rehberlik edecek bireylerin seçilmesi
 8. Adım 2: Yakınsama işlemleri süreci
 9. Komşuluk araması işlemleri
 10. Çeşitlilik işlemleri
 11. Adım 3: Güncelleme süreci
 P -deki çözüm adayları arasından hayatta kalanların belirlenmesi
 12. **Sonlandırma kriteri sağlanıncaya kadar arama sürecinin devam etmesi**
 13. **end**
-

Algoritma 1 incelendiğinde, MSA algoritmalarında optimizasyon süreci çözüm adaylarının oluşturulması ile başladığı görülmektedir. Denklem 1’de matematiksel tanımı verilen optimizasyon problemi üzerinden değerlendirildiğinde, çözüm adayları optimizasyon problemindeki m -adet tasarım parametresi için bu tasarım parametrelerinin arama uzayındaki kısıtlarını sağlayan m -boyutlu vektörlerdir. Algoritma 1’de ikinci satırda verildiği gibi, Denklem 2 kullanılarak n -adet çözüm adayından oluşan bir P -popülasyonu oluşturulur. Denklem 1’de verilen amaç fonksiyonu kullanılarak P ’deki her bir çözüm adayının uygunluk değeri hesaplanmak suretiyle Denklem 4’teki F -matrisi oluşturulur. Algoritma 1’de 6-12 satırları arasında MSA algoritmalarının arama süreci yaşam döngüsü

işletilir. Bu süreçte esas olarak üç işlem gerçekleştirilir. Bu işlemlerin ilki Adım-1’de sunulan rehber seçim sürecidir. Rehber seçim sürecinde, P’deki çözüm adayları arasından seçilenler popülasyon üyelerinin arama uzayındaki yeni konumlarının belirlenmesinde kullanılırlar. Rehber seçim sürecinde rastgele, sıralı, uygunluk değeri esaslı aç gözlü, uygunluk-uzaklık dengesi esaslı aç gözlü ve olasılıksal yöntemler olmak üzere çeşitli rehber seçim yöntemlerinden birkaçı kullanılır. Algoritma 1’deki ikinci adımda yakınsama işlemleri gerçekleştirilir. Yakınsama işlemi arama uzayındaki bir konumun yakın komşuluğunda ya da arama uzayındaki herhangi bir konumda yapılan aramaya dayanmaktadır. Bilinen bir konumun yakın çevresinde yürütülen aramalar genellikle komşuluk araması olarak ifade edilirken, bilinmeyen ya da amaç fonksiyon değeri hakkında bilgi sahibi olunmayan bir bölgede yapılan arama çeşitlilik olarak tanımlanır. Bu süreçte yakınsama denklemleri kullanılır. Yakınsama denklemlerindeki parametrelerin ince ayar yöntemleri ile belirlemesi ise hiper parametre optimizasyonu olarak bilinir. Yakınsama denklemlerindeki katsayıların ve çeşitli parametrelerin değerlerinin atanmasında sıklıkla kaotik haritalar ve Levy adımları olarak bilinen yöntemler kullanılır. Algoritma 1’de son adımda güncelleme işlemleri gerçekleştirilmektedir. Güncelleme işleminin amacı P-deki vektör sayısını sabit tutmak ve sadece en iyi çözümlerin P’de olmasını sağlamaktır. MSA algoritmalarında güncelleme mekanizmasının tasarımında uygunluk değeri esaslı yöntem en çok kullanılandır. Bu yöntemin dışında yeni oluşturulan çözüm adayının öncekinin yerine yazılması ya da doğal hayatta kalan mekanizması esaslı popülasyon sabitleme yöntemleri de kullanılmaktadır.

Optimizasyon problemlerini çözmek için, global arama ve yerel arama teknikleri kullanılmaktadır. Yerel arama, çözüm uzayının belirli bir bölgesindeki en iyi çözümleri arayarak işlem yapar. Global arama ise genel olarak en iyi çözümü bulmayı hedefler. (Şekil 1)Yerel arama ile bulunan en iyi noktanın, çözüm uzayındaki genel en iyi noktayı garanti etmediği bir durum söz konusudur.



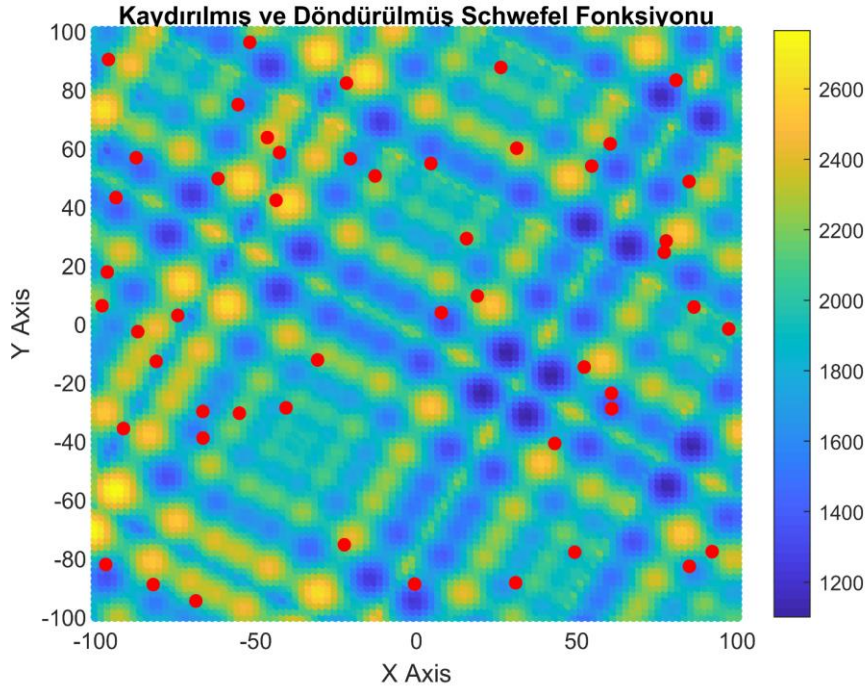
Şekil 1. Çok modlu arama uzayında yerel ve global çözüm noktaları

Doğadan esinlenilerek geliştirilen metasezgisel algoritmalarından 1.2’de bahsedilmişti. Her biri, farklı metasezgisel algoritma olmasına rağmen problemin çözümüne farklı yaklaşımlar sergilese de temelde hepsinin arama süreci yaşam döngüsü aynıdır ve 4 aşamadan oluşur. Popülasyon oluşturma, rehber seçim, komşuluk araması ve çeşitlilik ve son olarak güncelleme mekanizması tasarımı. Her bir adım, arama performansını önemli bir şekilde etkiler. Aşağıda metasezgisel algoritmaların temel adımları açıklanmıştır.

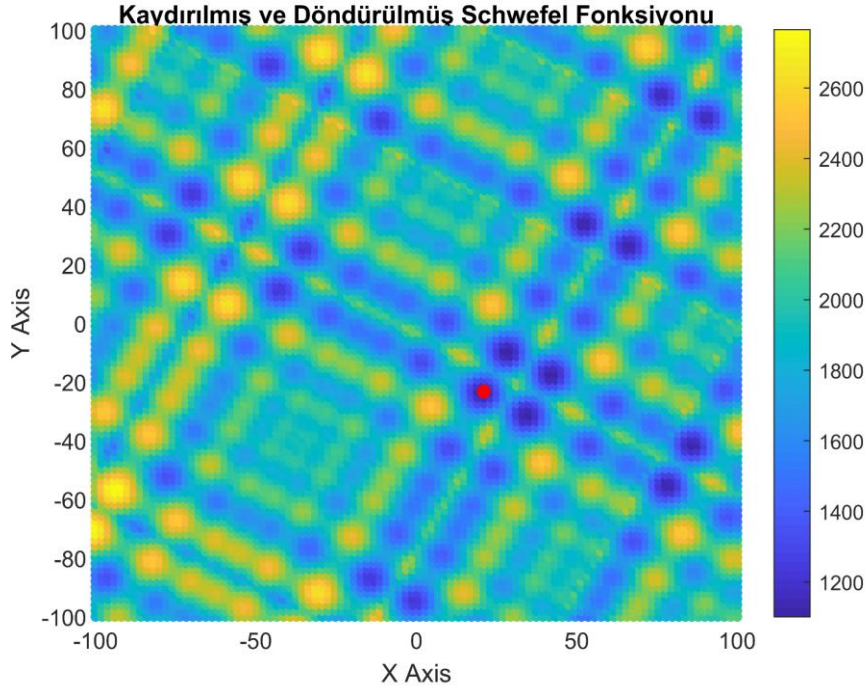
1.3.1. Popülasyon Oluşturma Yöntemleri

MSA algoritmalarının arama performansı üzerine etki eden 4 aşaması olduğundan 1.3 kısmında bahsedilmiştir. İlk aşama olan popülasyonun oluşturulma aşaması, diğer aşama olan komşuluk araması ve çeşitlilik operatörlerinin performansını etkilemektedir. Büyük ölçekli ve çok modlu problemler için, popülasyon oluşturma etkisi daha belirgindir ve birçok algoritma, farklı popülasyon oluşturmada global optimum bulma olasılığında farklılıklar gösterebilir. Oluşturulan popülasyon ve arama uzayında homojen dağılım oranı popülasyondaki bireylerin birbirlerine katkı verme oranını değiştirecektir. Yüksek homojenlikte dağılım, popülasyondaki bireylerin birbirlerine katkısını arttıracığından, algoritmanın erkenden yerel çözüm tuzaklarına takılmasını engelleyebilecektir. Bu durumda çeşitlilik operatörünün işlevini tam yerine getirmesine olanak sağlayacaktır. Bu durumu IEEE CEC 2020 (CEC20) (Yue, ve diğerleri, 2019) karşılaştırma problemlerinden 2.’si olan Kaydırılmış ve Döndürülmüş Schwefel Fonksiyonu üzerinden incelenmiştir.

Schwefel fonksiyonunda çoklu yerel çözüm tuzakları bulunmaktadır. MSA algoritmalarının yerel çözüm tuzaklarına takılması olası bir durumdur. Algoritmanın yerel çözüm tuzaklarına takılmasını engellemek için popülasyonun erkenden yerel bir tuzak noktasında toplanmadan, popülasyonda çeşitlilik sağlanılarak arama yapılması gerekmektedir. Çeşitliliğin büyük oranda sağlanabilmesi için popülasyon bireylerinin dağınık olmasında büyük fayda vardır. Şekil 2’de popülasyon büyüklüğü 50 olan DE algoritmasının genlerinin başlangıç noktaları görülmektedir. Arama genleri, arama uzayına karışık bir şekilde dağılmıştır. Bu durumda algoritma, komşuluk araması ve çeşitlilik adımlarını daha verimli bir şekilde gerçekleştirecektir. Bu durum Şekil 3’te görüldüğü üzere, popülasyondaki 50 gende global çözüm noktasında konumlanmıştır.

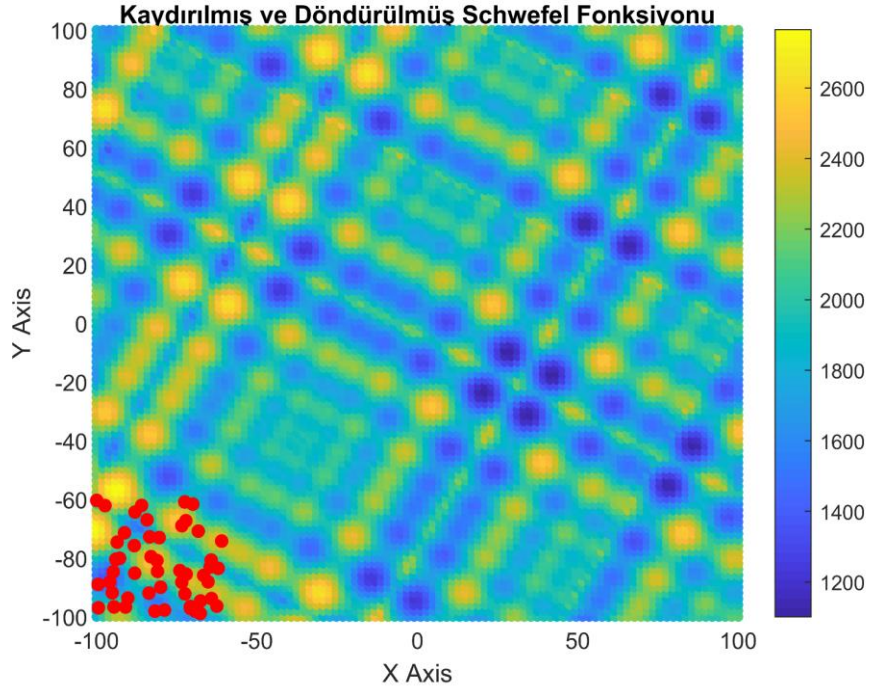


Şekil 2. DE popülasyon bireyleri başlangıç noktaları

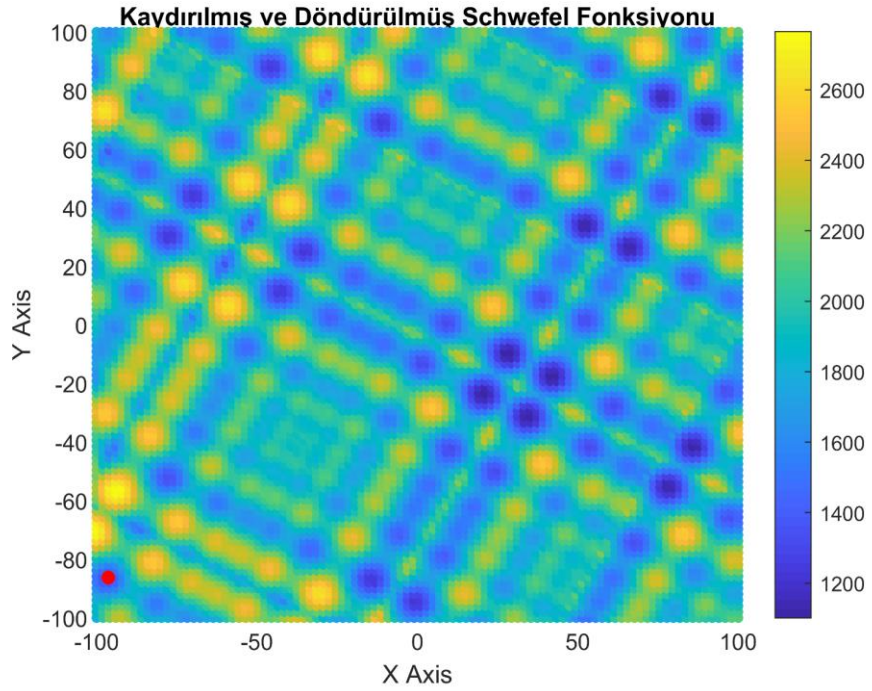


Şekil 3. DE algoritmasında bulunan en iyi çözüm

Başlangıç popülasyonunun homojen dağılmadığı bir durumda ise algoritmanın performansı Şekil 4'te görülmektedir. Şekil 4'te popülasyon, x ekseninde $[-50, -100]$ arasında dağılırken, y ekseninde de $[-60, -100]$ arasında rastgele dağılım göstermiştir. DE popülasyonundaki genler birbirlerine çok yakın olduğundan dolayı x $[-50, 0]$ ve y $[-60, -40]$ bölgesini geçememektedir. Genleri bu bölgeyi geçememesinin sebebi, bu bölgeyi geçmelerini sağlayacak bir çeşitlilik geninin bulunmamasıdır. Algoritmaların çeşitlilik çarpanına sahip olması durumunda, bu çeşitlilik çarpanının yüksek bir değere sahip olması, beklenen çeşitlilik etkisinin oluşturulmasına katkı sağlayabilir. DE algoritması normal değerleriyle çalıştırıldığında Şekil 5'te görüleceği üzere yerel çözüm tuzağına takılmıştır ve beklenen sonucu bulamamıştır.



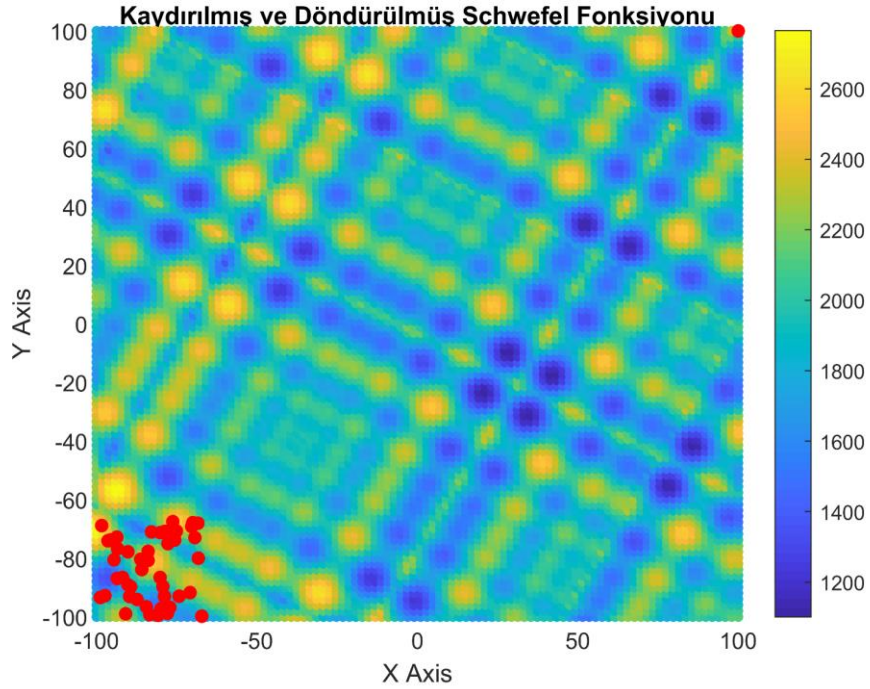
Şekil 4. DE popülasyon bireyleri başlangıç noktaları



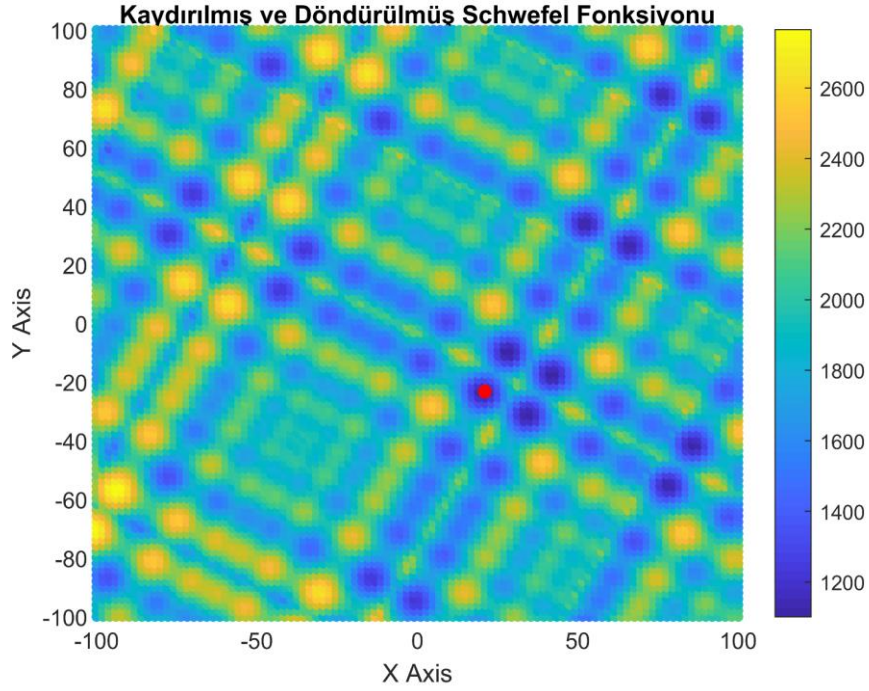
Şekil 5. DE algoritmasında bulunan en iyi çözüm

Şekil 6'da popülasyon büyüklüğü 50 olan DE algoritmasının 46 geni x, y [-65, -70] arasında rastgele dağılmıştır. Bu dağılım popülasyonun 0,92'sini kapsamaktadır.

Algoritmanın 4 geni ise x, y $[100,100]$ noktasında konumlandırılmıştır. Şekil 1.3.1.3. incelendiğinde, arama yaşam döngüsünde x $[-50,0]$ ve y $[-60,-40]$ bölgesini geçmesini sağlayacak bir çeşitlilik geninin bulunmamasından dolayı yerel çözüm tuzakına takıldığı belirtilmişti. Fakat Şekil 6’da bu çeşitliliği sağlayacak 4 gen $[100,100]$ noktasına konumlandırılmıştır. Bu durumda örneğin $[-65, -65]$ ve $[100, 100]$ ’e konumlandırılmış iki genin arasındaki öklidyen uzaklık mesafesi 233.21 birim olacaktır. Aradaki mesafe çok büyük olduğundan dolayı x ekseninde $[-50, -100]$ ve y ekseninde de $[-60, -100]$ bölgesine konumlandırılmış genleri daha ileri noktalara ulaşmasına imkân sağlayacaktır. Bu durum Şekil 7’de gözlemlenmektedir, algoritma global çözüm noktasını 4 gen sayesinde bulmayı başarmıştır.



Şekil 6. DE popülasyon bireyleri başlangıç noktaları



Şekil 7. DE algoritmasında bulunan en iyi çözüm

Literatürdeki bazı popülasyon oluşturma yöntemlerinden bahsedecek olursak, çoğu esas olarak arama uzayında düzgün yayılma fikrine dayanmaktadır. Uygulamaları kolaydır. Aşağıda popülasyon oluşturma yöntemlerinin birkaçından bahsedilmiştir:

Latin Hiperküp Örnekleme: Latin hiperküp örnekleme arama uzayını eşit aralıklı parçalara bölerek bir ızgara oluşturur. Ardından, her boyuttaki değişkenleri temsil etmek üzere belirli aralıklarla rastgele noktalar seçer. Bu noktalar, temsil edilecek değişkenlerin tamamen katmanlı bir özellik uzayında yer almasını sağlamak için yardımcı değişkenler kullanılarak elde edilir. Latin hiperküp örnekleme kullanımında, örnek noktalarının aşırı toplanmasını engellemek için, arama uzayında düşük yoğunluklu bir şekilde örnek üretme amacı güdülür (McKay, Beckman, & Conover). Araştırmalar, bu tür örneklemenin, homojen dağılımlara kıyasla daha etkili bir yayılma sağlayabildiğini gösterse de yüksek boyutlu problemler için belirgin bir avantaj sağlamadığını ortaya koymaktadır.

Tekdüze Dağılım: Tekdüze dağılımlar popülasyon oluşturmada yaygın olarak kullanılırlar. Eşit olasılıklı sonuçlara sahip olasılık dağılımlardır. $[a, b]$ aralığı boyunca düzgün bir dağılım tanımlanır ve genellikle $X \sim U(a, b)$ şeklinde yazılmaktadır.

Beta Dağılımı: Beta dağılımı $(0, 1)$ aralığı boyunca sürekli bir olasılık dağılımıdır. $X \sim Be(a, b)$ şeklinde yazılabilir.

Normal Dağılım: En önemli ve en yaygın kullanılan dağılım biçimidir. İlk kez Carl Friedrich Gauss tarafından keşfedilmiştir ve bu yüzden Gauss dağılımı olarak da adlandırılır. Sürekli dağılım türlerinden biridir. Biyolojik, psikolojik, sosyolojik, ekonomik, kimyasal ve fiziksel birçok değişkenin genellikle normal dağılım gösterdiği gözlemlenmiştir. İki adet parametreye sahiptir. Çoğu dağılım gibi normal dağılımın da bir ortalaması (μ) ve standart sapması (σ) vardır.

1.3.2. Rehber Seçim Yöntemleri

Rehber seçim yöntemlerinde, arama sürecini yönlendirecek/rehberlik edecek konumlar/bireyler belirlenir. Komşuluk araması ve çeşitlilik formüllerinde kullanılacak çözüm adaylarını kapsamaktadır. Keşfedilecek yeni konumların verimliliği seçilen bireylerin uyumuna bağlıdır. Verimsiz bireylerin seçilmesi durumunda algoritmanın yerel çözüm tuzaklarına takılarak kabul edilebilir çözümler bulması zorlaşır. Birbirleriyle en iyi etkileşimde bulunacak bireylerin seçilmesinde seçim yöntemleri birleştirilerek popülasyon içerisinde daha uyumlu bireyler seçilebilir.

Açgözlü Seçim Yöntemi (Uygunluk Değeri Esaslı): Açgözlü seçim yöntemi deterministik olup doğadaki elitizm mantığını esas alır. Evrim sürecinde olduğu gibi topluluk üyeleri arasından en güçlü olanın seçildiği yöntemdir. Arama sürecine rehberlik edecek bireylerin seçiminde uygunluk değeri kullanılmaktadır. Bundan dolayı bu seçimin yapılmasından önce popülasyondaki bireylerin uygunluk değerlerinin hesaplanması gerekmektedir. Örneğin Simbiyotik Organizmalar Algoritmasının (Cheng & Prayogo, 2014) kommensalizm ve mutualizm evrelerinde kullanılan en iyi organizma açgözlü yöntemine göre seçilmektedir.

Sıralı Seçim Yöntemi: Sıralı seçim yöntemi deterministik bir yöntemdir. Arama sürecine rehberlik edecek birey ya da bireylerin seçiminde herhangi bir kontrol yapılmaz. Mevcut popülasyondaki tüm bireyler gezilerek popülasyon güncellenir. Popülasyon güncellenirken kullanılacak rehberlerden bir tanesi, popülasyonun o anki çözüm adaydır. Birçok algorithmada rehber seçim yöntemlerine sıralı seçim yöntemi de dâhil edilmektedir.

Olasılıksal Seçim Yöntemi: Olasılıksal seçim yöntemi, MSA algoritmalarında arama sürecine rehberlik edecek bireylerin seçiminde yaygın olarak kullanılmaktadır. Rulet tekerleği ve turnuva yöntemleri en sık kullanılan olasılıksal iki seçim yöntemidir.

Rulet tekerleği MSA algoritmalarının miladı sayılabilecek genetik algoritmaların ilk seçim metodudur. Popülasyon içerisindeki çözüm adaylarının uygunluk fonksiyonu değerlerine bağlı olarak seçilme olasılıkları hesaplanılmaktadır. Popülasyon bireylerinin uygunluk değerleri oranınca rulet çemberi üzerinde bölge ayrılmaktadır. Uygunluk değeri yüksek olan çözüm adayının çember üzerinde daha fazla alana sahip olmaktadır. Bundan dolayı uygunluk değeri iyi olan bireylerinde seçilme olasılığı artmaktadır. Rastgele bir sayı üretilerek, üretilen sayının çember üzerinde temsil ettiği alan içerisindeki çözüm adayları seçilmektedir. Rulet Tekerleğinin sözde kodu Algoritma 2’de verilmiştir.

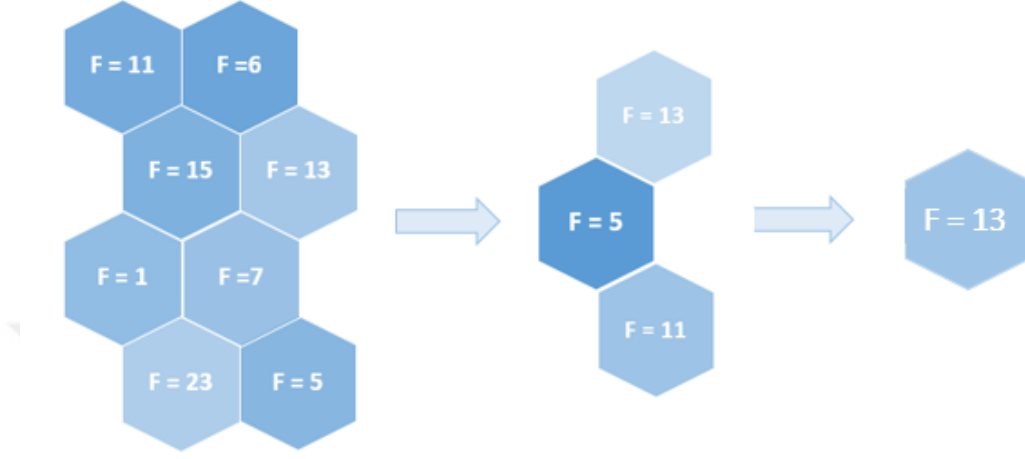
Algoritma 2. Rulet Seçim Yöntemi Sözde Kodu

```

1.  p = MSA algoritmasında çözüm adayı sayısı
2.  d = Optimizasyon probleminin boyutu
3.   $X_{[p,d]}$  = Çözüm adayları topluluğu,
4.   $U_{[p]}$  = Çözüm adaylarının uygunluk değerleri
5.   $Y_{[p]}$  = Çözüm adaylarının rulet tekerleği yüzdeleri
6.   $K_{[p]}$  = Çözüm adaylarının rulet tekerleği konumları,
7.  t = 0,  $K_{[0]} = 0$ ;
8.      for i=1:p
9.          t=t+  $U_{[p]}$ 
10.         end
11.         for i=1:n
12.              $Y_{[i]} = U_{[i]} / U_{[p]}$ 
13.              $K_{[i]} = Y_{[i]} + K_{[i-1]}$ 
14.         end
15.         konum=rand (0,1) // Rulet tekerleğini döndür ve durduğu konumu
16.         belirle
17.         for i=1:n
18.             if ( $K_{[i-1]} < \text{konum} \leq K_{[i]}$ )
19.                 Seçilen çözüm adayı= $X_{[i]}$ 
20.             end

```

Turnuva seçim yönteminde popülasyondan belirlenen bir k adeti kadar popülasyondan rastgele birey seçilir. Seçilen bireyler uygunluk değerlerine göre kıyaslanır. Uygunluk değeri en yüksek olan birey seçilmiş olur. Turnuva yönteminin akışı Şekil 8’de görülmektedir.



Şekil 8. Turnuva yöntemi 3'lü turnuva akışı

Uygunluk-Uzaklık Dengesi (FDB): Seçim yöntemleri MSA algoritmalarında arama operatörlerinde kullanılan rehber çözüm adaylarının belirlenmesinde kullanılmaktadır. MSA algoritmalarının tasarımlarında olduğu gibi, seçim yöntemlerinin de tasarlanmasında doğadaki süreçler referans alınmaktadır. Doğadaki işleyişi daha iyi uygulayabilmek için son zamanlarda geliştirilen Uygunluk-Uzaklık Dengesi (Kahraman, Aras, & Gedikli, 2020) güçlü bir rehber seçim yöntemidir. FDB seçim yöntemi doğadaki işleyişi daha etkili bir şekilde taklit edebilmek amacıyla yürütülen araştırmaların sonucunda geliştirilmiştir.

MSA algoritmalarında komşuluk araması ve çeşitlilik görevlerinde birden fazla çözüm adayı görev almaktadır. Bu noktalarda seçilecek çözüm adaylarının birbirleriyle etkileşime girerek mevcut popülasyon için daha güçlü bireylerin üretilmesi hedeflenmektedir. Bu noktalardaki bireylerin birbirleriyle uyumunu maksimuma çıkartacak bireylerin seçiminde FDB kullanılmaktadır. FDB’de ara sürecinde popülasyondaki en iyi çözüm adayına en fazla katkıyı verecek bireylerin belirlenmesini sağlamaktadır. Bunun için FDB seçim yönteminde çözüm adaylarının arama sürecine katkılarını gösterebilecek skor değerleri hesaplanmaktadır. Skor hesabında çözüm adaylarının iki özelliği kullanılmaktadır. Bunlar popülasyondaki bireylerin konumlarının ve uygunluk değerlerinin en iyi çözüm adayına olan uzaklıklarıdır. Hesaplanan FDB skorları büyükten küçüğe sıralanarak, en iyi skora sahip

olan bireyler rehber çözüm adayları olarak seçilir. Bu açıklamalara göre, FDB seçim yönteminin matematiksel ifadesi aşağıda açıklanmaktadır.

Bir optimizasyon probleminde optimizasyon probleminin boyutu m ve bu problemin çözümlenmesinde kullanılan MSA algoritmasında çözüm adayı sayısı n olmak üzere, popülasyondaki i . çözüm adayı P_i 'nin, popülasyondaki en iyi uygunluk değerine sahip olan P_b 'e olan oklit uzaklığı Denklem 5'te verildiği gibi hesaplanır.

$${}_{i=1}^n \forall P_i, D_{P_i} = \sqrt{(p_{i[1]} - p_{b[1]})^2 + (p_{i[2]} - p_{b[2]})^2 + \dots + (p_{i[m]} - p_{b[m]})^2} \quad (5)$$

P-popülasyonundaki her bir çözüm adayının FDB skorları hesaplanır. Bu süreçte, çözüm adaylarının uygunluk değerleri ve Denklem 5'teki uzaklık değerleri kullanılır. Bu iki parametrenin skor hesaplamasında birbirlerine baskınlık kurmalarının önüne geçilmesi için hem uygunluk değerlerinin hem de uzaklık değerlerinin normalize edilmeleri gerekmektedir. Buna göre, her iki parametrenin $[0, 1]$ aralığında normalize edilmiş uygunluk ve uzaklık değerleri sırasıyla $normF$ ve $normD_p$ ile Denklem 6 ve 7'de verildiği gibi temsil edilir.

$$normF_{[i]} = \frac{F_i - F_{min}}{F_{max} - F_{min}} \quad (6)$$

$$normD_{p[i]} = \frac{Dp_{[i]} - Dp_{[min]}}{Dp_{[max]} - Dp_{[min]}} \quad (7)$$

Yukarıdaki açıklamalara bağlı olarak bir P-popülasyonundaki çözüm adaylarının FDB skorları Denklem 8 kullanılarak hesaplanır.

$${}_{i=1}^n \forall P_i, \quad S_{P_{[i]}} = w * normF_{[i]} + w * normD_{P_{[i]}} \quad (8)$$

Bu açıklamalara göre, bir P-popülasyonundaki çözüm adayları arasından rehber seçileceği anda tüm çözüm adaylarının FDB skorları hesaplanarak n -boyutlu SP vektörü elde edilir. SP vektöründeki elemanlar FDB skoru en iyi olandan en kötü olana doğru sıralandığında, SP vektöründeki ilk eleman FDB skoru en iyi olandır. Rehber seçiminde

FDB skorları dikkate alınarak ihtiyaç duyulduğu kadar çözüm adayının rehber olarak seçilmesi mümkündür.

Dinamik Uygunluk-Uzaklık Dengesi(dFDB): FDB arama sürecine rehberlik eden rehber çözüm adaylarının seçiminde kullanılmaktadır. Seçim için popülasyondaki bireylerin skorları hesaplanmaktadır. Popülasyondaki bireylerin konumlarının ve uygunluk değerlerinin en iyi çözüm adayına olan uzaklıkları hesaplanarak bireylerin skorları hesaplanır. Skor hesaplamasında kullanılan 2 özelliğinde skor üzerindeki etkisi eşittir. Arama süreci yaşam döngüsünde FDB'nin farklı etkiler yaratmasını sağlamak için 2 özelliğin etki oranları değiştirilebilir yapılarak dFDB (Kahraman, ve diğerleri, 2022) tasarlanmıştır. Örneğin arama süreci yaşam döngüsü sonlarına doğru, azalan oranda bireylerin konumları ve artan oranda ise uygunluk değerlerinin etki edilmesi istenilebilir. Bu uygulandığında arama süreci sonlarına doğru ağırlıklı olarak popülasyondaki çözüm adaylarının uygunluk değer skorlarının en yüksek olanının seçilmesini sağlayacaktır. Bu durumda %100 oranda uygunluk değer bazlı skollama yapılırsa, skoru en yüksek çıkacak olan en kötü birey olacaktır. Algoritmaların iç tasarım dengelerine göre, etki oranlarını dinamik olarak değiştirerek dFDB'nin etki oranları artırılabilir. Bu açıklamalara göre, dFDB yönteminde skor hesabının matematiksel tanımı aşağıda sunulmaktadır.

Yukarıdaki açıklamalardan anlaşıldığı gibi klasik FDB yönteminde rehber adaylarının uygunluk değerleri ve uzaklık değerlerinin skor hesabında eşit derecede etkili oldukları görülmektedir. 8 numaralı denklemde görüldüğü gibi bu iki parameterenin ağırlık katsayısı $w=0.5$ dir. Dinamik FDB yönteminde w -ağırlık katsayısı arama süreci yaşam döngüsü boyunca değişmektedir. Buna göre, arama süreci yaşam döngüsünün her bir adımı güncel iterasyon numarası yani hh -parametresi ile temsil edildiğinde, amaç fonksiyonu azami değerlendirme sayısı $maxFE$ 'e bağlı olarak dFDB skoru Denklem 9'da verildiği gibi hesaplanır.

$$w_{dFDB} = \frac{hh}{maxFE} * (1 - as) + as \quad (9)$$

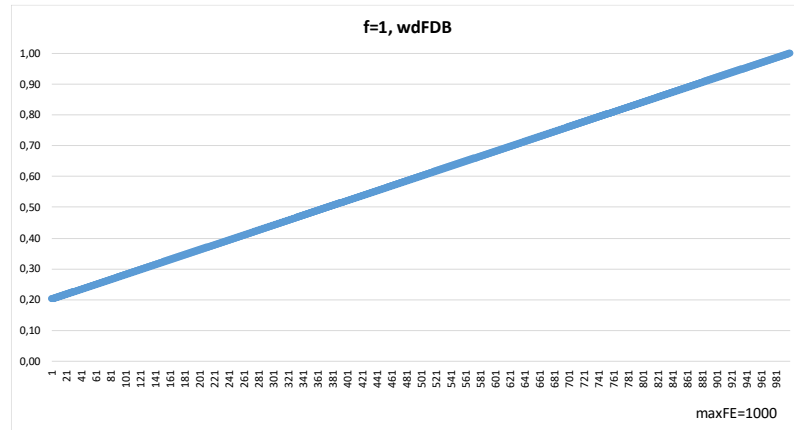
$0 < as < 0.5, as \in R$ olmak üzere as parametresi w_{dFDB} 'nin alabileceği asgari değeri temsil etmektedir. as –parametresi, rehber seçiminde uygunluk değerinin etkisinin belirli bir eşik değerin altına düşmemesini sağlamaktadır. Böylelikle, arama süreci boyunca uygunluk değerinin etkisinin baskın olması ve arama sürecinin sadece uzaklığa bağlı olarak gerçekleşmesi önlenmiş olur. Uzaklık çeşitliliği sağlarken, uygunluk değerinin

gözetilmediği bir uzaklık ise arama sürecinin rastgele aramaya dönüşmesine neden olur. Buna göre, frekansı ayarlanabilir bir testere dişi sinyali üretmek için Algoritma 3'te verilen kod kullanılabilir. Böylelikle w_{dFDB} sinyalinin testere dişi şeklinde değişimi sağlanmış olur.

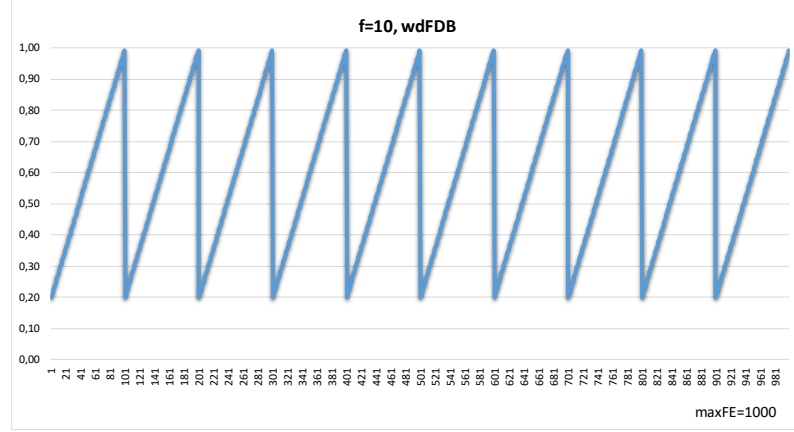
Algoritma 3. Frekansı ayarlanabilir bir testere dişi sinyalinin üretilmesi için sözde kod

1. **Başla**
 2. $maxFE$ ve f parametrelerine değer ata
 3. **if** ($maxFE/f \geq 1$)
 4. $mod_ref = round(maxFE / f)$
 5. **for** $ii=1: maxFE$
 6. $h = (ii \bmod mod_ref)$
 7. $w_{dFDB} = \frac{hh}{maxFE} * (1 - as) + as$
 8. **end**
 9. **end**
 10. **End**
-

Algoritma 3'te verilen sözde kodda $as=0.2$ ve frekans sayısı $f=1$ ve 10 olan w_{dFDB} sinyalleri üretilmiştir. $maxFE=1000$ iken üretilen testere dişi sinyaller (w_{dFDB}) Şekil 9 ve Şekil 10'da verilmiştir.



Şekil 9. $maxFE=1000$, $as=0.2$ ve frekans sayısı $f=1$ için w_{dFDB}



Şekil 10. $maxFE=1000$, $as=0.2$ ve frekans sayısı $f=10$ için w_{dFDB}

Bu açıklamalara bağlı olarak dFDB yönteminde skor Denklem 10'da verildiği gibi hesaplanır.

$$i=1^m \forall P_i, \quad S_{P_i} = w_{dFDB} * normF_i + (1 - w_{dFDB}) * normD_{P_i} \quad (10)$$

1.3.3. Komşuluk Araması ve Çeşitlilik

MSA algoritmalarının arama süreçlerinde temel iki adım bulunmaktadır. Bu adımlar komşuluk araması ve çeşitliliktir. Bu iki adım arasındaki uyum algoritmanın performansını önemli ölçüde etkilemektedir. Çeşitlilik adımında MSA algoritmaları arama uzayında büyük atlamalar yaparak keşfedilmemiş bölgelerin bulunmasını amaçlanmaktadır. Komşuluk aramasında ise belirli bir nokta etrafında küçük adımlarla arama/sömürü yapılmasıdır. Çeşitliliğin arama sürecinde hızlı bir azalış yapıldığında keşfedilecek yeni noktaların bulunması zorlaşacağından global çözümler yerine yerel çözüm tuzaklarına takılabilmektedir. Komşuluk araması yoğun bir şekilde uygulanması durumunda ise engelli yakınsama problemine takılarak yerel çözüm tuzağına yakalanmaya neden olmaktadır. Yerel çözüm tuzaklarına yakalanmamak için komşuluk araması yapılırken aynı zamanda çeşitlilik yapılmalıdır.

Çeşitlilik ve komşuluk araması arama parametreleri ve popülasyon bireyleriyle yapılmaktadır.

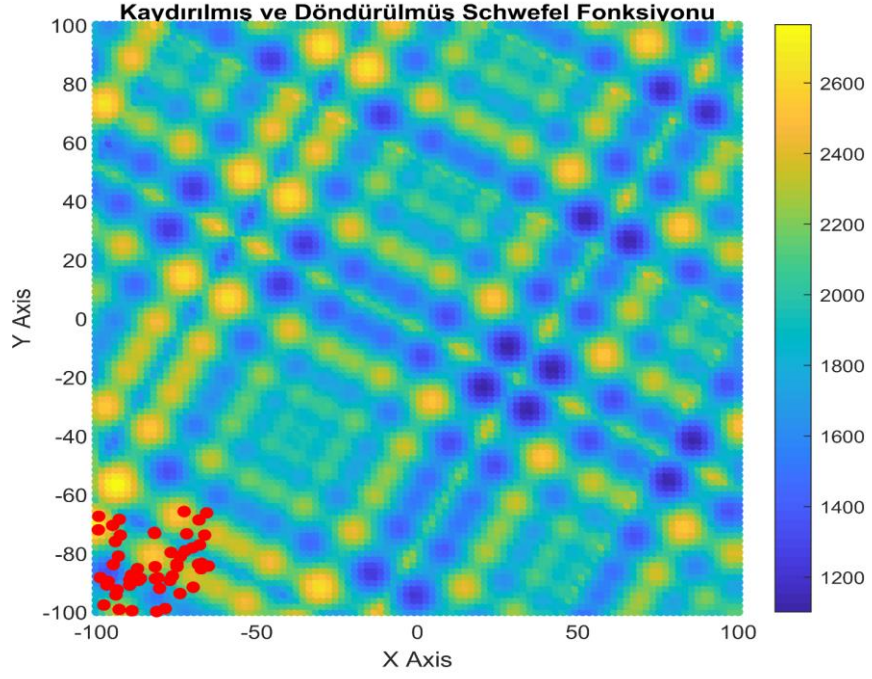
Komşuluk araması MSA algoritmalarında genellikle en iyi çözüm adayları etrafında yapılmaktadır. Örneğin Yapay Arı Kolonisi (ABC) algoritmasında kullanılan işçi arıların

görevi belirlenen besin kaynaklarının etrafında arama yaparak komşuluk araması görevini yerine getirirler.

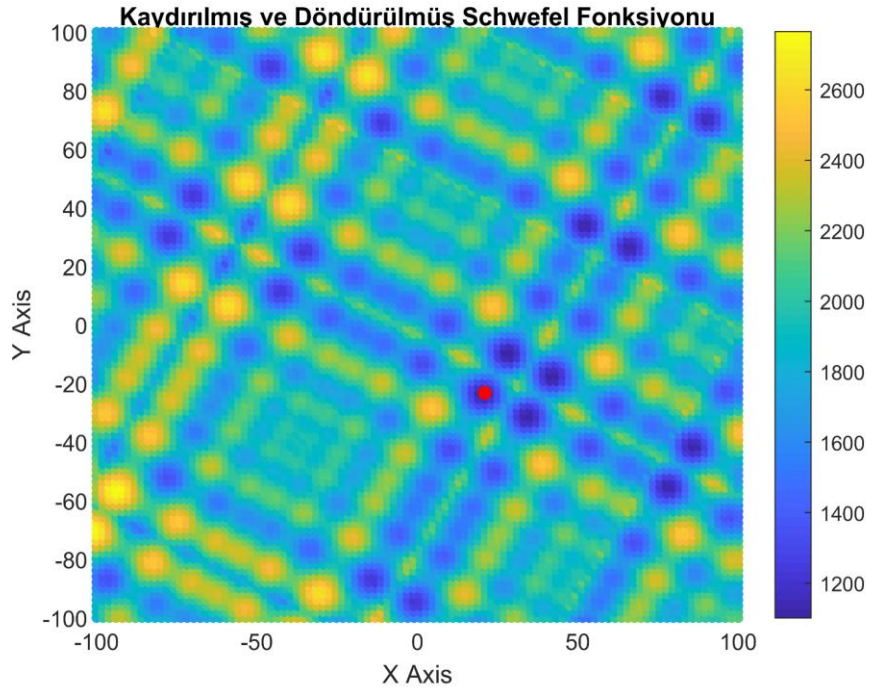
Çeşitlilik ise birbirlerinden farklı çözüm adaylarıyla yapılmaktadır. Popülasyondaki en iyi iki birey aynı noktayı temsil edebileceğinden, bu bireylerin etkileşiminden üretilecek yeni birey arama uzayında keşfedilmemiş bir noktası temsil etme ihtimali çok düşüktür. Bundan dolayı çeşitlilik adımıyla birbirlerine benzemeyen bireylerin kullanılmasında fayda olacaktır. Örneğin popülasyondaki en kötü birey ile en iyi bireyin ve en iyi iki bireyin çeşitlilik adımıyla kullanıldığında en kötü ve en iyi birey arasında benzerlik oranı düşük olabileceğinden yeni bir noktanın keşfedilme ihtimali artacaktır. Çeşitlilik adımıyla popülasyon dışında anlık rastgele bireylerde oluşturulmaktadır. Örneğin ABC algoritmasında kullanılan kâşif arılar problemin parametrelerinin alt ve üst sınırları kullanılarak popülasyon dışında anlık rastgele yeni bir arı oluşturulur. Bu adımda çeşitlilik sağlanmış olur.

Popülasyon bireylerinin dışında bu sürece dâhil olan parametrelerde vardır. Algoritmelerde belirlenen ve rastgele üretilen bir çarpan ile komşuluk araması ve çeşitlilik oranları artırılabilir. Popülasyon Oluşturma Yöntemleri kısmında DE algoritmasının popülasyonu belli bir noktada toplanmıştı. MSA algoritmaları arama sürecinin sonlarına doğru yüksek keşif yaptığından popülasyon bir noktada toplanır. Popülasyonun bir noktada toplanmadan önce arama uzayında çeşitlilik adımıyla yeni noktalar bulunarak sömürülmesi gerekmektedir. DE algoritmasında tüm popülasyon birbirlerine çok yakın olduğundan dolayı büyük atlamalar gerçekleştirecek bir gen bulunamadığı için algoritma yerel çözüm tuzağına takılmaktadır. Bu durumdan kurtulmak için komşuluk araması ve çeşitlilik için kullanılan F parametresinin değeri artırılarak çeşitlilik adımı desteklenir.

Şekil 11’de yapılan çalışmada DE algoritmasındaki F parametresi 0.2- 0.8 arasında her gen için rastgele üretilmektedir. Bu değeri 0.7 – 1.7 değerlerinde ayarlama yapıldığında elde edeceğimiz sonuç Şekil 12’deki gibi olacaktır.



Şekil 11. DE popülasyon bireyleri başlangıç noktaları

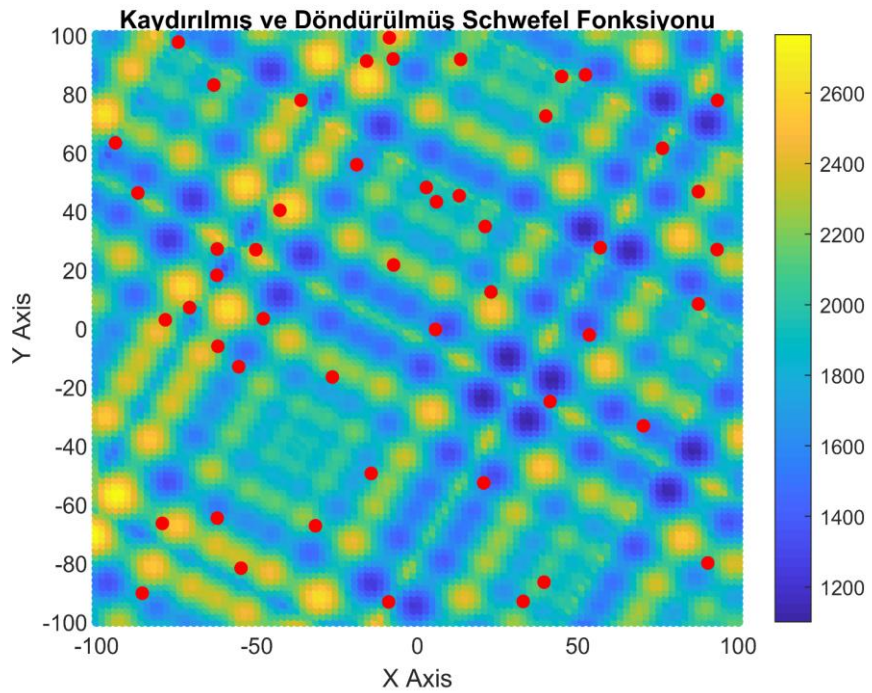


Şekil 12. DE algoritması bulunan en iyi çözüm

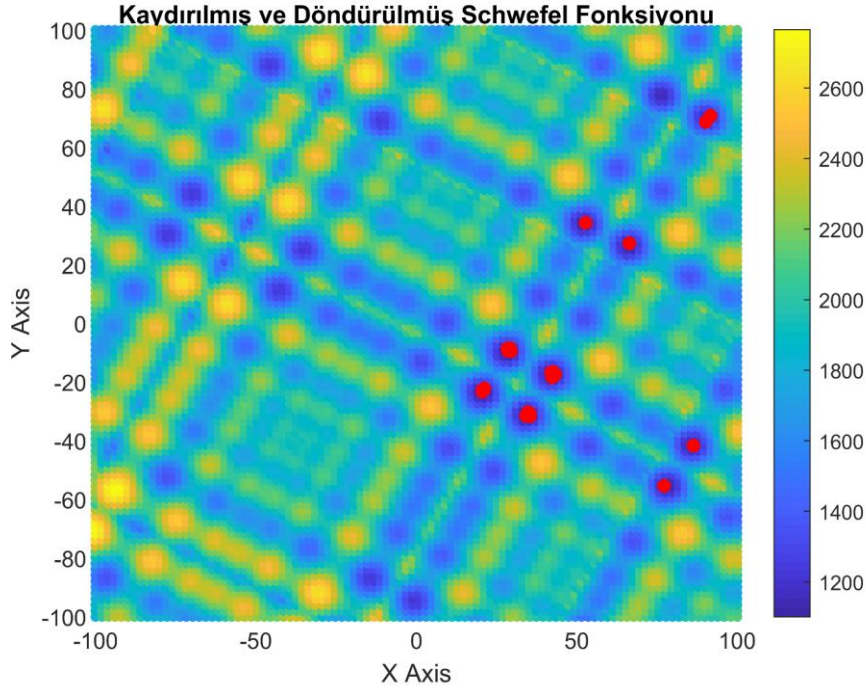
Şekil 12 incelendiğinde DE algoritması global çözüm noktasını bulmuştur. Algoritmadaki bu değişiklik F değerinde yapılan değişiklik sayesinde olmuştur. Fakat aynı

değişiklik, popülasyonun arama uzayında homojen dağıldığında yapılmış olsaydı algoritmanın performansı Şekil 14'deki gibi olacaktır.

Şekil 14 incelendiğinde DE algoritması global çözüm noktasına çok yakın bir noktayı bulabilmiştir. Fakat popülasyon bir noktada toplanmak yerine 9 noktaya dağılmıştır. Bu durumda çeşitlilik oranı yüksek uygulandığı için arama yaşam döngüsünün sonlarında en iyi çözüm noktası etrafında komşuluk araması yapmak için yeterince çözüm adayının olmamasına neden olmuştur. Normal ayarlarla çalıştırıldığında global çözüm bulunurken yüksek F çarpanının etkisi yüzünden global çözüm bulunamamıştır.



Şekil 13. DE popülasyon bireyleri başlangıç noktaları



Şekil 14. DE algoritması bulunan en iyi çözüm

Sonuç olarak çeşitlilik ve komşuluk adımının performansı popülasyondaki dağılım oranından etkilenmektedir. Yerel çözüm tuzaklarından kaçılmak için yüksek mutasyon katsayıları algoritmaların performanslarında olumlu etkiler yaratabilmektedir.

1.3.4. Güncelleme Mekanizması Tasarımı

MSA algoritmalarında popülasyon boyutu arama süreci öncesinde sabit oranda belirlenir. Arama sürecinde popülasyon büyüklüğü belirlenen sabiti aşmamaktadır. Su süreç içinde popülasyon boyutu sabit kaldığı gibi bazı algoritmalarda zaman içerisinde popülasyon büyüklüğü düşürülmektedir. Popülasyon boyutu sabit kaldığı için, arama sürecinde üretilen yeni bireylerin mevcut popülasyona alınıp alınmayacağının kararının verilmesi gerekmektedir. Popülasyona yeni bireyin alınması durumunda, popülasyon boyutu aşıldığı için yeni eklenen bireylerin sayısı kadar popülasyondan bireylerin atılması/öldürülmesi gerekmektedir.

Güncelleme mekanizmalarında bireylerden kimin ölüp kimin yaşayacağı karar verilir. Bu kıyaslama mevcut popülasyondaki birey ile onun yerine üretilmiş olan çözüm adayı arasında gerçekleştirilir. Literatürde bu noktalarda kullanılan 3 tip mekanizma vardır. Bunlar uygunluk esaslı, doğrudan ve doğal seçim mekanizmalarıdır.

Uygunluk değeri esaslı mekanizma MSA algoritmalarında çok sık kullanılan yöntemdir. Güçlü olan hayatta kalır mantığı esas alınmıştır. Güç kavramı MSA algoritmaları için uygunluk değerini ifade etmektedir. Arama sürecinde oluşturulan yeni bireyler ile popülasyondan seçilen bireyler uygunluk değerine göre kıyaslama yapılır. Bu kıyaslamada uygunluk değeri iyi olan birey popülasyona alınırken, uygunluk değeri kötü olan birey ise öldürülür. Bu mekanizmayı kullanan algoritmalar Uyarlanabilir GÜDÜMLÜ Diferansiyel Evrim Algoritması (Agde) (Mohamed & Mohamed, 2019) ve Stokastik fraktal Arama Algoritması (SFS) (Salimi, 2015)

Doğrudan mekanizmasında popülasyon güncelleme işleminde herhangi bir kıyaslama yapılmaz. Arama sürecinde oluşturulan bireyler popülasyona direkt dahil edilip, popülasyondan ilgili bireyler çıkartılır. Bu mekanizmayı kullanan algoritmalar Salp Sürü Algoritması (SSA) (Mirjalili, ve diğerleri, 2017) ve Gri Kurt Optimizasyon Algoritması (GWO) (Mirjalili, Lewis, & Mirjalili, 2014) örnek verilebilir.

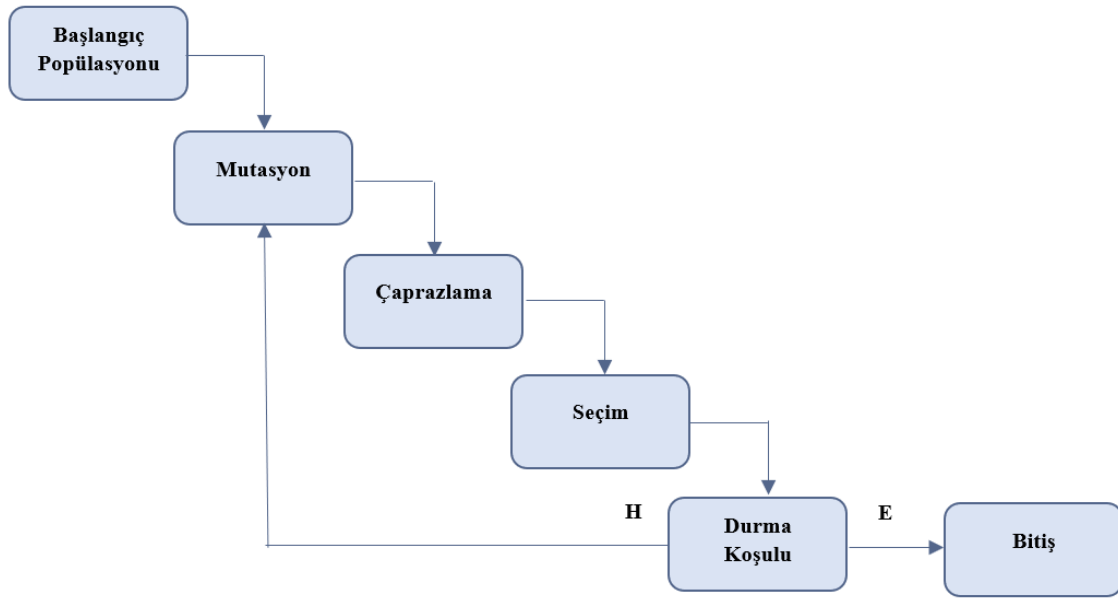
Doğal seçim mekanizmasında (Kahraman, Katı, Aras, & Taşci, 2023) uygunluk değeri esaslı mekanizmanın savunduğu “Güçlü Olan Hayatta Kalır” ilkesi yerine, bireylerin çevreleriyle olan ilişkilerinden esinlenen “Çevreye En İyi Adapte Olan Hayatta Kalır” ilkesini savunmaktadır. Süreçte hayatta kalacak ve elenecek bireyleri doğadaki gibi belirlemek için, üç kriteri içeren çok boyutlu bir değerlendirme mekanizması tasarlanmıştır. Bu mekanizmaya göre, popülasyon çeşitliliğine ve genetik çeşitlilik bakımından çiftleşme havuzundaki diğer bireylere kıyasla daha fazla katkı sağlayan ve aynı zamanda güçlü bireysel yeteneklere sahip olan çözüm adayı hayatta kalır. Bu mekanizmada yaşayacak olan birey bu 3 kriterde en iyi skoru elde eden aday olup, hayatta kalacak olandır.

1.4. Diferansiyel Evrim Algoritması

Diferansiyel Evrim (DE) algoritması, literatürde oldukça bilinen ve temel evrim algoritmalarından biridir. Bu çalışmada, yüzde hata değişimi temelli bir arama yöntemi üzerinde detaylı bir inceleme gerçekleştirilmiştir. Popülasyondaki en iyi bireyin uygunluk değerindeki iyileşme miktarları, arama süreci boyunca dikkatle takip edilmektedir. İyileşme miktarı belirlenen eşik değerden düşükse, algoritma tasarlanan halkalardan -yani yeni yakınsama denklemlerinden oluşan zincir üzerinde gezinerek- çeşitlilik adımlarını destekleyip, algoritmanın komşuluk ve çeşitlilik aramasını güçlendirerek daha kabul edilebilir çözümler elde etmeyi hedeflemektedir. Birinci bölümün ilerleyen kısmında MSA

algoritmalarının arama sürecinden bahsedilmiştir ve DE algoritmasına değinilip, arama süreci yaşam döngüsü örneklerle açıklanmıştır.

DE algoritması, K. Price ve R. Storn tarafından geliştirilen popülasyon tabanlı bir algoritmadır. Çaprazlama, mutasyon ve seçim gibi genetik algoritmalarda bulunan benzer operatörleri kullanır. Problem çözümü için, iterasyonlar boyunca operatör yardımları ile daha iyi sonuçlar elde edilmeye çalışılmaktadır. DE sadece bir birey ile değil, tüm popülasyon ile ilgilenir. Diferansiyel evrim algoritmasının temel basamakları Şekil 15'te görülmektedir:



Şekil 15. Diferansiyel evrim algoritması temel basamakları

Kromozomlar tek tek ele alınarak, rasgele seçilen diğer üç kromozom birlikte kullanılıp yeni bir kromozom-birey oluşturulmaktadır. Bu işlemler sırasında mutasyon ve çaprazlama operatörleri kullanılır. Seçim operatöründe var olan kromozom ile yeni oluşturulan kromozomun uygunlukları karşılaştırılarak uygunluğu daha iyi olan kromozom yeni birey olarak bir sonraki popülasyona aktarılır.

1.4.1. Arama Süreci Yaşam Döngüsü

Arama süreci yaşam döngüsünde 4 adımın bulunduğu 1.3 Metasezgisel Algoritmaların Arama Süreci kısmında bahsedilmiştir. DE algoritması için bu adımlar aşağıdaki gibi uygulanmaktadır:

Popülasyon Oluşturma Evresi: Kullanıcı tarafından belirlenen kromozom sayısı NP olarak temsil edilmektedir. Her bir kromozoma ait gen sayısı D olarak temsil edilip, probleme ait değişken/parametre sayısını belirlemektedir. DE’de yeni kromozom-bireylerin üretilmesi için var olan kromozom dışında üç adet kromozom gerektiği için NP her zaman dörtten büyük olmalıdır. Başlangıç popülasyonu oluşturan Denklem 11’de görülmektedir.

$$\forall \leq NP \wedge \forall \leq D: x_{i,j,G=0} = x_j^{(1)} + rand_j[0,1].(x_j^{(u)} - x_j^{(1)}) \quad (11)$$

$x_{i,j,G}$: G. jenerasyona ait kromozomun j. parametresi,

$x_j^{(1)}$: Parametreye ait alt sınır değeri,

$x_j^{(u)}$: Parametreye ait üst sınır değeri,

$rand_j[0,1]$: j parametresi için 0-1 arasında rastgele üretilen sayıyı gösterir.

Başlangıç popülasyonu üretildikten sonra DE adımları tekrarlanarak algoritma tamamlanır.

Mutasyon: Mutasyon, kromozomun bir kısım genlerinde rastgele değişiklik yapmaktır. Yapılacak değişikliğin büyük olması, benzersiz bir kromozom oluşturabileceğinden keşif kabiliyetini arttırmaktadır. DE algoritmasının mutasyon aşamasında popülasyondan rastgele seçilen 3 kromozomun genleri etkileşime girerek yeni bir kromozom oluşturulur. Seçilecek 3 farklı kromozom 1’den N_p ’ye kadar mevcut olan kromozomlar içinden seçilir. Seçilen kromozomlardan ilk ikisinin farkı alınarak, genellikle 0-2 arasında değer alan F parametresi ile çarpılır ve böylece ağırlıklandırılmış fark vektörü oluşur. Oluşturulan ağırlıklandırılmış fark vektörü ile seçilen üçüncü kromozom toplanır ve böylece çaprazlamada kullanılacak olan kromozom mutasyon aşamasında elde edilir. Mutasyon denklemi Denklem 12’de görülmektedir:

$$\forall j \leq D: n_{i,j,G+1} = x_{j,r_3,G} + F. (x_{j,r_1,G} - x_{j,r_2,G}) \quad (12)$$

Çaprazlama: Çaprazlama aşamasında bir çaprazlama oranı (CR) 0 ile 1 arasında belirlenmelidir. İterasyondaki kromozom ($x_{i,G}$) ile, mutasyon aşamasında ($u_{i,G+1}$), elde edilen mutasyon kromozomu genleri CR katsayısına göre işleme alınarak, popülasyon için aday kromozom üretilir. İki ebeveyn kromozomlarının genleri CR olasılığına göre seçilecektir. Mutasyon kromozomuna ait her bir gen CR olasılıkla mutasyon kromozomundan, 1-CR olasılıkla mevcut kromozomdan seçilir. CR katsayısının oranı

mevcut kromozom üzerinde ne kadar değişiklik yapılacağını belirlemektedir. Bu aşamada sömürü veya keşif gerçekleştirilir. Çaprazlama denklemi Denklem 13'te görülmektedir.

$$\forall j \leq D: x_{j,u,G+1} = \begin{cases} x_{j,n,G+1} & \text{rand } [0,1] \leq CR \\ x_{j,i,G} & \text{aksi durumlar için} \end{cases} \quad (13)$$

Güncelleme Mekanizması Tasarımı: Mutasyon ve çaprazlama operatörleri kullanılarak yeni bir kromozom-birey elde edilmiştir. Yeni jenerasyona geçecek olan kromozom ise uygunluk değeri ile belirlenmektedir. Seçim operatörü var olan kromozom ile yeni oluşturulan kromozomlarla birlikte değerlendirilerek popülasyon güncellenir. Uygunluğu iyi olan kromozom popülasyonun yeni bireyi olarak seçilir. Seçim operatörü Denklem 14'te görülmektedir.

$$\forall i \leq D: x_{i,G+1} = \begin{cases} x_{u,G+1} & \text{eğer } f(x_{u,G+1}) \leq f(x_{i,G}) \\ x_{i,G} & \text{diğer durumlarda} \end{cases} \quad (14)$$

Diferansiyel evrim algoritmaları bir örnek ile aşağıdaki gibi incelenmiştir.

Şekil 16'te mevcut popülasyon görülmektedir. Çaprazlamaya uğrayacak hedef kromozom dışında mevcut popülasyondan rasgele 3 kromozom daha seçilir (r_1, r_2, r_3).

Mevcut Popülasyon	Hedef Kromozom		Rastgele Seçilen Birbirinden Farklı Üç Kromozom		
	X_{1G}	X_{2G}	X_{3G}	X_{4G}	X_{5G}
	1.5731	1.5698	3.6501	1.3619	1.9138
	2.3423	3.6754	1.9721	2.0034	3.7510
	1.9876	2.4567	2.3664	3.8946	3.7031
	2.7812	1.4156	3.2134	2.9987	2.6094
	3.6548	3.1821	1.4444	1.7852	1.4883
	1.2567	3.0392	3.1245	3.5182	3.3067
	2.8990	2.6432	1.7986	1.4897	2.4598
	2.1874	1.2288	2.5713	3.4999	1.1434
	3.2901	1.9297	3.7678	2.2357	3.0029
	3.7399	2.8165	2.9875	2.7733	1.2487

Şekil 16. Mevcut popülasyon

Çaprazlamada kullanılacak mutasyon kromozomu elde etmek için seçilen kromozomlardan ikisinin farkı alınarak değeri 0.65 seçilen F parametresi ile çarpılarak ağırlıklandırılmış fark vektörü oluşturulur ve seçilen diğer kromozomla toplanarak çaprazlamada kullanılacak olan diğer kromozom yani mutant vektör oluşturulur. Bu aşamalar Tablo 1’de görülmektedir.

Tablo 1. Mutant vektörün oluşturulması

X_{5G}	$X_{2G} - X_{4G}$	Mutant Vektör
1.9138	0.2079	2.048935
3.7510	1.672	4.8378
3.7031	-1.4379	2.768465
2.6094	-1.5831	1.580385
1.4883	1.3969	2.396285
3.3067	-0.479	2.99535
2.4598	1.1535	3.209575
1.1434	-2.2711	-0.332815
3.0029	-0.306	2.804
1.2487	0.0432	1.27678

Tablo 2’de görüldüğü üzere, hedef kromozom ile oluşturulan yeni mutant kromozomu karşılaştırmak için her bir gen için 0-1 arasında rasgele bir sayı üretilir ve CR ile kıyaslanır. Üretilen sayının CR’den $\leq$ olması durumunda mutant kromozomun geni seçilir.

Tablo 2. Çaprazlama aşaması

X_{1g}	CR	Mutant Vektör	Çaprazlama
1.5731	rand [1] $\leq$ CR	2.048935	2.048935
2.3423		4.8378	2.3423
1.9876		2.768465	1.9876
2.7812	rand [4] $\leq$ CR	1.580385	1.580385
3.6548	rand [5] $\leq$ CR	2.396285	2.396285
1.2567	rand [7] $\leq$ CR	2.99535	1.2567
2.8990		3.209575	3.209575
2.1874		-0.332815	2.1874
3.2901	rand [10] $\leq$ CR	2.804	3.2901
3.7399		1.27678	1.27678

Bu örnekte uygunluk fonksiyonu genlerin toplamıdır. Tablo 3’te görüldüğü gibi mevcut popülasyondaki hedef kromozom için uygunluk değeri 25.7121 iken, yeni kromozomun uygunluk değeri 21.57606’dır. Bu aşamada seçim operatörü kullanılarak yeni popülasyonun bireyi seçilir.

De algoritmasında seçim mekanizması güçlü olan hayatta kalır mantığını esas alan uygunluk değeri esaslı mekanizma kullanılmaktadır. Minimizasyon problemlerinde uygunluk değeri en düşük olan en güçlü bireydir. Tablo 3’te yeni oluşturulan kromozomun uygunluk değeri daha düşük olduğu için daha güçlü kabul edilir. Bu durumda X_{1g} kromozomu popülasyondan atılırken, yeni oluşturulan kromozom popülasyona dahil edilir.

Tablo 3. Oluşan yeni kromozom

X_{1G}	Çaprazlama Sonucu Oluşan Kromozom
1.5731	2.048935
2.3423	2.3423
1.9876	1.9876
2.7812	1.580385
3.6548	2.396285
1.2567	1.2567
2.8990	3.209575
2.1874	2.1874
3.2901	3.2901
3.7399	1.27678
$\sum$ 25.7121	$\sum$ 21.57606

2. YAPILAN ÇALIŞMALAR

DE Algoritması, literatürde yaygın olarak kullanılan temel evrim algoritmalarından biridir. Bu tez çalışmasında, DE algoritması üzerinde yüzde hata değişimine dayalı arama yöntemi geliştirilmiştir. Çalışmada popülasyondaki en iyi bireyin uygunluk değerindeki iyileşme miktarları arama yaşam döngüsü boyunca takip edilmektedir. Eğer iyileşme miktarı belirlenen eşik değerden küçükse tasarlanmış olan halkalardan yani, yeni yakınsama denklemlerinden oluşan zincir üzerinde gezinerek algoritmanın komşuluk araması ve çeşitlilik adımlarının desteklenerek daha kabul edilebilir çözümler bulunması hedeflenmiştir.

2.1. Önerilen Yöntem

MSA algoritmalarının performansı, özellikle komşuluk araması ve çeşitlilik görevlerindeki başarılarına bağlıdır. Bununla birlikte, karmaşıklık düzeyi yüksek olan optimizasyon problemlerinin çözülmesinde bazı zorluklarla karşılaşmaktadır. Bu zorlukların başında arama uzayındaki yerel çözüm tuzakları gelmektedir. MSA algoritmalarının çok sayıda yerel çözüm tuzaklarına sahip olan problemlerin arama uzaylarındaki küresel en iyi çözüme yakınsayabilmeleri için arama süreci boyunca komşuluk araması-çeşitlilik dengesini etkili bir şekilde sürdürmeleri gerekmektedir. MSA algoritmalarının yerel çözüm tuzaklarından kurtulabilmeleri ve komşuluk araması-çeşitlilik dengesini etkili bir şekilde sağlayabilmeleri yakınsama denklemlerinin tasarımına bağlıdır. Bu tez çalışmasının amacı, MSA algoritmalarında yakınsama denklemlerinin tasarımı için yeni ve etkili bir yöntem geliştirmektir.

Literatüre yeni önerilen yüzde hata değişimi esaslı arama yöntemiyle algoritmaların yakınsama problemlerini çözmeye ve çeşitlilik ile komşuluk araması yetenekleri artırılarak problemlerin daha başarılı bir şekilde çözülmesi hedeflenmiştir. Bunun için arama süreci yaşam döngüsü boyunca, aramadaki iyileşmeye bağlı olarak yakınsama stratejisinin belirlenmesi amaçlanmaktadır. Yakınsama stratejisinden kastedilen konular aşağıda açıklanmaktadır.

- i) Komşuluk araması ve çeşitlilik operatörleri arasından hangisinin çalıştırılacağına karar verilmesi,

- ii) Rehberlik mekanizmasının tasarlanması ve hata değerine bağlı olarak rehberlerin belirlenmesi. Örneğin hatadaki değişim sıfırda komşuluk araması ve çeşitliliğin kademeli olarak uygulanması, komşuluk araması ve çeşitlilik denklemlerinin çeşitlendirilmesi ve alternatifler arasından ihtiyaca uygun olanın kullanılması.

MSA sürecinde yüzde hata değişiminde öncelikle hata hesaplanır. MSA algoritmalarında popülasyondaki en iyi bireyin amaç fonksiyon değeri f_{best} olmak üzere, minimizasyon problemlerinde bu amaç fonksiyon değeri minimize edilmeye çalışıldığı için bu değer hata (e) olarak kullanılmaktadır. Yani $e = f_{best}$ dir.

Hatadaki değişimin hangi sıklıkla hesaplanacağı önemlidir. Sıklığın belirlenmesinde başlıca kriter popülasyondaki tüm çözüm adayları için en az bir kere aramanın gerçekleştirilmesidir. Bu açıklamaya bağlı olarak, popülasyondaki tüm bireylerin amaç fonksiyon değerinin hesaplandığı periyot bir dönem (epoch) olarak tanımlanır. Periyot ise hata değişiminin kontrol edileceği dönem sıklığıdır. Bu durumda hata değişimi Periyot değerine bağlı olarak dönemlerdeki f_{best} deki değişim olarak hesaplanır. Yüzde hata değişimi Denklem 15 ve Denklem 16'daki gibi hesaplanmaktadır.

$$\Delta e_{ce} = e_{[ce-per]} - e_{ce} \quad (15)$$

$$P\Delta e_{ce} = \left(\frac{\Delta e_{ce}}{e_{[ce-per]}} \right) * 100 \quad (16)$$

Ce: Güncel dönem değeridir. Arama süreci yaşam döngüsü başladığında ilk değeri sıfırdır. Arama süreci popülasyon için tamamlandığında bir dönem değeri tamamlanmış olacak. Buna bağlı olarak ce değeri 1 artırılmaktadır.

Δe_{ce} : Ce dönemindeki hata değişimidir.

e_{ce} : ce'inci dönemdeki hata değeridir. Haliyle, arama süreci yaşam döngüsünün başından sonuna kadar tüm dönemler için hatanın tutulduğu bir dizidir.

per: Hata değişiminin kontrol edildiği dönem sıklığı yani periyottur. Bu değer 1'den küçük olamaz.

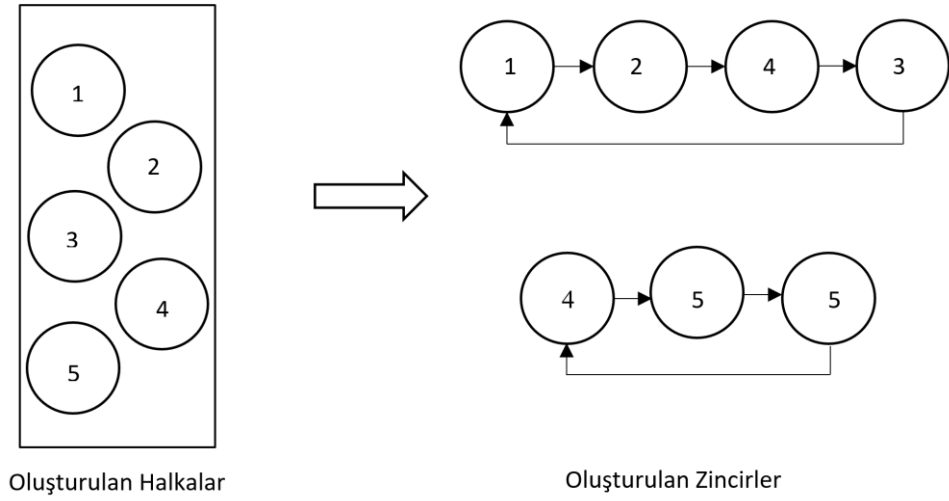
$P\Delta e_{ce}$: Ce dönemindeki yüzde hata değişimidir.

Algoritmanın yakınsama denklemlerindeki rehberlerin seçimi için farklı yöntemler kullanılır ve yakınsama denkleminin farklı versiyonları tasarlanır. Bu denklemlerin her birisi

algoritmanın yakınsama denklemi olarak kullanılmaya adaydır ve bu denklemlerin her birine halka adı verilir. Oluşturulacak halka sayısı için bir sınır bulanmamaktadır.

Oluşturulan halkalardan arama yaşam döngüsünde kullanılmak üzere zincir oluşturulur. Oluşturulan zinciri gerçek dünyada kullanılan zincirlere benzetilebilir. Halkalar bir sonraki halkalara bağlıdır ve zincir üzerinde sınırsız halka bulunabilir. Aynı mantıkla oluşturduğunuz halkalar birleştirilerek zincirler oluşturulur. Oluşturulan zincirler algoritmaların arama süreçlerine dahil edilmektedir. Arama süreçlerinde zincirin kullanıp kullanılmayacağını kararı ise yüzde hata değişimine göre yapılmaktadır. Zincirlerin aktif edilmesinde periyot değerine göre hesaplanan yüzde hata değişimine göre yapılmaktadır. Yüzde hata değişimi eğer belirlenen bir eşik yüzde hata değerinden daha düşükse bu durumda zincir arama sürecinde ilk halkasıyla kullanılmaya başlanır.

Yüzde hata değişiminin eşik yüzde hata değerinden düşük olması durumunda, algoritmanın yerel çözüm tuzağına takıldığı düşünülebilir. Bu durumda tasarladığımız zincirin, algoritmayı yerel çözüm tuzağından çıkartması beklenmektedir. Algoritmanın yerel çözüm tuzağından çıkması durumunda ise yüzde hata değişimi eşik yüzde hata değerinden büyük olacaktır. Bundan dolayı zincir aktifken yüzde hata değişimi takip edilmektedir. Zincir üzerindeki her bir halkaya algoritmayı yerel çözüm tuzağından çıkartması için bir periyot boyunca süre verilir. Yüzde hata değişiminde eğer beklenen iyileşme yapılamazsa, zincir üzerindeki bir sonraki halkaya geçiş yapılır. Yani arama sürecinde zincir üzerindeki sadece bir halka kullanılır. Zincir üzerindeki son halkaya gelinmesi durumunda ise, zincirin ilk halkasına tekrar dönülür. Bu süreç bir döngü olarak tekrarlanır. İlgili süreç Şekil 17’de görülmektedir.



Şekil 17. Halkalardan zincir oluşturma

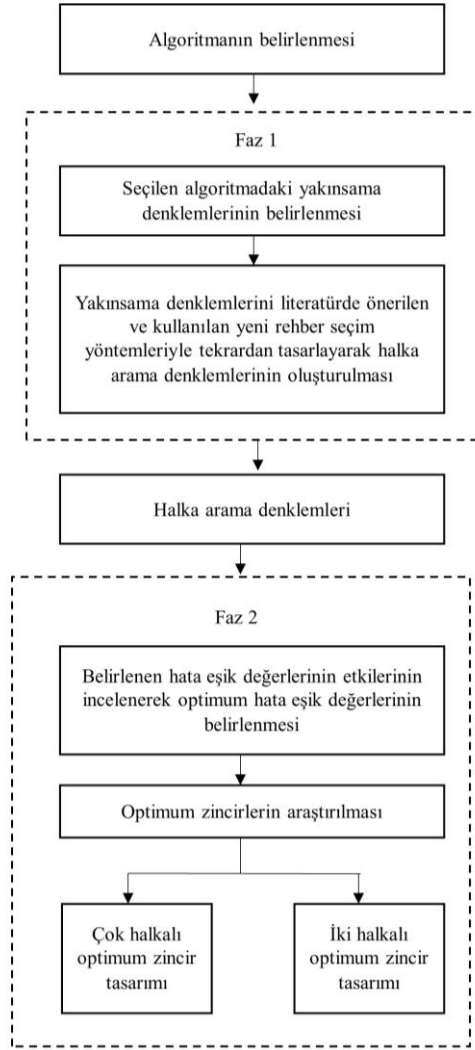
Şekil 17’de 1, 2, 3, 4, 5 numaralı örnek halkalar oluşturulmuştur. Oluşturulan halkalardan örnek 2 adet zincir oluşturulmuştur. İlk zincirde 1, 2, 4 ve 3 numaralı halkalar bulunmaktadır. Her bir halka bir periyot boyunca çalıştırılacaktır. 3 numaralı halkaya yani zincirin sonuna gelindiğinde, 1 numaralı halkaya yani zincirin başına dönüş yapılacaktır.

Halkalar tasarlanırken rehber seçim yöntemi olarak literatüre yeni kazandırılan FDB ve açgözlü yöntem kullanılmıştır. FDB ile tasarlanan yeni yakınsama denklem halkalarının skorları yani performansları ölçülmeye çalışılmıştır. Performans ölçme çalışmalarında CEC20 problem seti 10, 30, 50 ve 100 boyutlarında belirlenen 9 hata eşik değerleriyle çalıştırılmıştır. Yapılan çalışma sonucunda boyut bazlı en iyi eşik değerleri ve halka belirlenmeye çalışılmıştır. Belirlenen halkalardan kombinasyonlar oluşturularak zincirler oluşturulmuştur. Arama yaşam döngüsünde yüzde hata değeri hesaplanmıştır. Hesaplanan yüzde hata değeri, yüzde hata eşik değerinden küçük olması durumunda tasarlanan zincirler arama sürecinde kullanılmak üzere aktifleştirilmiştir. Yüzde hata değerindeki iyileşme, eşik değerini geçene kadar algorithmada sadece tasarlanan zincir kullanılmaktadır. Bu süreç MSA algoritmaları için Algoritma 4’te gösterilmektedir.

Algoritma 4. Yüzde Hata değişimli MSA algoritmalarında arama sürecinin genel adımları

1. Optimizasyon probleminin tanımlanması (amaç fonksiyonu, tasarım parametreleri vd. öğeler)
 2. Algoritma parametrelerinin tanımlanması ve çözüm adayları topluluğunun oluşturulması
 3. Aday çözümlerin uygunluk değerlerinin hesaplanması
 4. Zincir Aktif mi?
 - a. **Hayır (Adım 5'ten devam et)**
 - b. **Evet (Adım 6'dan devam et)**
 5. Arama Süreci Yaşam Döngüsü
 - a. Seçim süreci
 - b. Komşuluk araması ve çeşitliliğin sağlanması
 - c. Çözüm adayı topluluğunun güncellenmesi
 6. Zincirli Arama Süreci Yaşam Döngüsü
 - a. Seçim süreci
 - b. Komşuluk araması ve çeşitliliğin sağlanması
 - c. Çözüm adayı topluluğunun güncellenmesi
 7. **Yüzde hata değişimi kontrolü**
 8. Sonlandırma kriteri sağlandı mı?
 - a. **Hayır (Adım 4'e dön)**
 - b. **Evet (Arama sürecini sonlandır ve en iyi çözüm adayını kaydet)**
-

DE algoritmasındaki uygunluk değerindeki iyileşme yüzde hata değeri ile takip edilerek, algoritmanın olası yerel çözüm tuzaklarına takıldığında, bu tuzaklardan kurtulabilme kabiliyeti kazandırılmaya çalışılmıştır. DE algoritmasının arama yaşam süreci desteklenmiş ve algoritmanın performansında iyileştirmeler elde edilmeye çalışılmıştır. Önerilen yöntemin akış diyagramı Şekil 18'de görülmektedir.



Şekil 18. Yöntem akış diyagramı

2.1.1. Halka Arama Denklemlerinin Oluşturulması

DE algoritmasında mutasyon aşamasında kullanılan Denklem 12 arama sürecine yön vermektedir. Bu denklemde sırasıyla kullanılan $r3$, $r1$ ve $r2$ rehber çözüm adayları olup mevcut popülasyondan rastgele seçilmektedir. Rehber bireylerin seçiminde rulet seçim yöntemi, FDB ve açgözlü yöntemleri kullanılarak yeni arama denklemleri yani halkalar oluşturulmuştur. Oluşturulan halkalar Tablo 4'te görülmektedir. FDB seçim yönteminde Denklem 8'de FDB için n -boyutlu SP vektörü elde edilir. SP vektöründen hangi çözüm adayının seçildiği bilgisi sıra numarası olarak FDB teriminin yanına eklenmiştir. FDB1 ifadesi SP vektörünün ilk elamanını yani skoru en yüksek çözüm adayını ifade etmektedir.

Halkaların oluşturulmasında kullanılan rulet seçim yöntemi, FDB için oluşturulan n-boyutlu SP vektöründen seçim yapmada kullanılmıştır.

Mutasyon aşamasında kullanılan denklemdaki F çarpanının yerine bazı halkalarda “İterasyondaki Kromozomun FDB Skoru” kullanılmıştır. Bu ifade SP vektöründeki, iterasyondaki numaraya karşılık gelen SP elemanı yani FDB skorudur.

Tablo 4. Oluşturulan halka denklemleri

Numara	R3	F	R1	R2
1	FDB2	F	En İyi Çözüm Adayı	FDB1
2	R3	F	En İyi Çözüm Adayı	FDB1
3	FDB1	F	En İyi Çözüm Adayı	R2
4	R3	F	R1	R2
5	R3	İterasyondaki Kromozomun FDB Skoru	R1	R2
6	FDB1	F	FDB1	R2
7	FDB2	F	R1	FDB1
8	R3	İterasyondaki Kromozomun FDB Skoru	FDB1	FDB2
9	En İyi Çözüm Adayı	F	FDB1	R2
10	En İyi Çözüm Adayı	İterasyondaki Kromozomun FDB Skoru	FDB1	R2
11	En İyi Çözüm Adayı	F	FDB1	FDB2
12	R3	F	FDB1	FDB2
13	FDB2	F	FDB1	R2
14	En İyi Çözüm Adayı	İterasyondaki Kromozomun FDB Skoru	FDB1	FDB2
15	R3	İterasyondaki Kromozomun FDB Skoru	FDB1	FDB2
16	FDB2	İterasyondaki Kromozomun FDB Skoru	FDB1	R2
17	FDB1	F	FDB2	R2

Tablo 4'ün devamı

18	FDB1	İterasyondaki Kromozomun FDB Skoru	FDB2	R2
19	FDB1	F	R1	FDB2
20	FDB1	İterasyondaki Kromozomun FDB Skoru	R1	FDB2
21	FDB1	F	R1	R2
22	dFDB1	F	R1	R2
23	Rulet dFDB	F	R1	R2
24	Rulet FDB	F	R1	R2
25	R3	F	dFDB1	R2
26	R3	F	Rulet dFDB	R2
27	R3	F	Rulet FDB	R2

2.1.2. Optimum Halka Eşik Değerinin Belirlenmesi

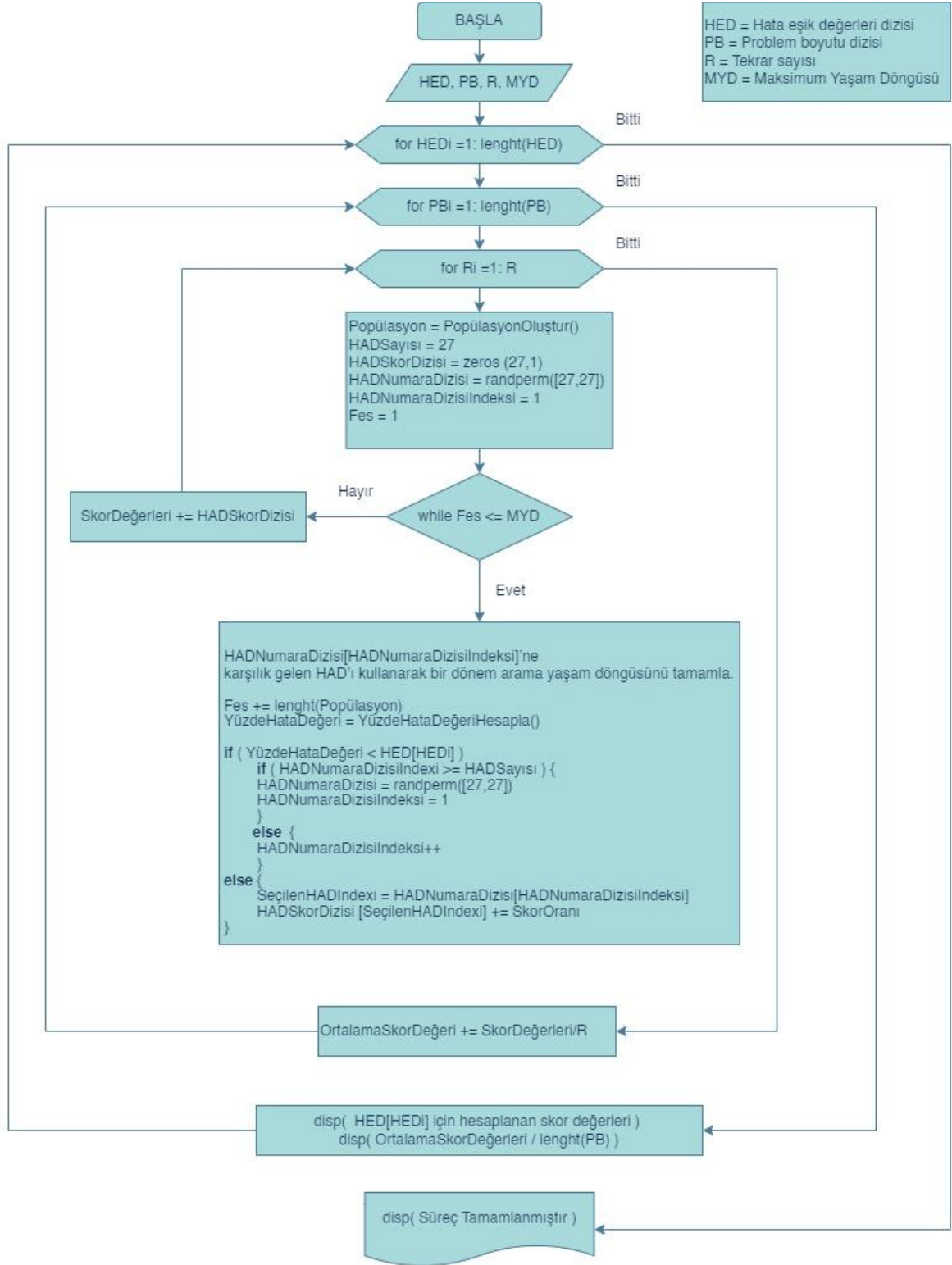
Optimum hata eşik değerlerinin (HED) belirlenmesinde CEC20 problem seti üzerinde 10, 30, 50 ve 100 boyutlarında çalışılmıştır. HED için aşağıdaki oranlar belirlenmiştir:

- $HED = \{0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1, 2, 5\}$

Her bir HED oranı için oluşturulan 27 adet halkanın skorları hesaplanmıştır. Skor hesabı için halkalar CEC20 problem seti üzerinde 10, 30, 50 ve 100 boyutunda çalıştırılmıştır. Her bir halkanın skorunu hesaplamak için, tüm halkaları içeren bir zincir tasarlanmıştır. Halkaların zincirdeki konumu rastgele olarak dağıtılmıştır. Arama süreci yaşam döngüsü boyunca zincir hep aktifte kalmıştır. Bir arama dönemi sonunda eğer ilgili halka ile yapılan aramada yüzde hata değişimi yaşam döngüsü için belirlenmiş HED değerinden büyükse, kullanılan halkaya belirlenen oranda skor eklemesi yapılmıştır. Yüzde hata değişiminin HED değerinden küçük olması durumunda ise, bir sonraki halkaya geçiş yapılmıştır. Zincir üzerinde eğer son halkaya gelinmiş ise, halkalar zincir üzerinde tekrardan rastgele dağıtılarak, zincirin başından işlem tekrardan başlatılmıştır. Halkalar süreçte rastgele dağıtılarak adilliğin sağlanması hedeflenmektedir. Süreç sonunda 10, 30, 50 ve 100 boyutunda çalıştırılan halkaların alınan tüm sonuçların ortalama skor değerleri hesaplanmıştır. 9 HED değeri için halkaların ortalama skorları büyükten küçüğe

sıralanmıştır. Skor sıralarına göre ilk 3 halka seçilir ve seçilen halkalardan zincirler oluşturulmuştur. İşlem sonucunda 9 HED değeri için 9 adet zincir oluşturulmuştur. Oluşturulan zincirler ile DE algoritması ilgili HED oranlarıyla CEC20 problem seti üzerinde 10, 30, 50 ve 100 boyutlarında çalışılmıştır. Alınan sonuçlar Friedman (Martin, Leblanc, & Toan, 1993) yöntemiyle analiz edilmiştir. Yapılan analiz sonucunda CEC20 problem seti üzerinde 10, 30, 50 ve 100 boyutları için optimum HED oranı Friedman skorları analizi sonucunda belirlenmiştir. Bu süreç aşağıdaki Şekil 19'da görülen akış diyagramında anlatılmıştır.





Şekil 19. Önerilen yöntemin akış diyagramı

2.1.3. Optimum Zincirlerin Araştırılması

Optimum zincirlerin araştırılmasında 2 farklı çalışma yapılmıştır. 2 farklı çalışma yapılmasının nedeni zincir tasarımının etkisinin daha iyi anlaşılmasıdır. Bu kapsamda çoklu halkalı ve iki halkalı optimum zincir tasarım çalışmaları yapılmıştır.

Çok Halkalı Optimum Zincirin Araştırılması: Optimum HED'ler belirlenirken, tüm halkaların skorları hesaplanmıştır. Skor değerleri büyükten küçüğe sıralanarak ilk 10 halka boyut zincir oluşturmak üzere seçilmiştir. Seçilen halkalar boyut bazlı HED'lere göre belirlenmiştir. Seçilen 10 halkadan 10 zincir oluşturulmuştur. Oluşturulan zincirler CEC20 10, 30, 50 ve 100 boyutlarında çalıştırılmıştır. Alınan sonuçlar Wilcoxon (Derrac, García, Molina, & Herrera, 2011) testi ile çalıştırılan boyutlardaki problem türlerine göre analiz edilmiştir. Bu durumda CEC20 problem setinde 10 adet problem bulunduğundan 4 boyut için toplamda 40 analiz yapılmıştır. Yapılan 40 analizin sonucunda, problem türü ve boyuta göre hangi zincirin kullanılması gerektiği belirlenmiştir. Tüm bu işlemlerin sonucunda tek bir durum oluşturularak, CEC20 problem seti için problem boyutu ve problem türüne göre otomatik zincir seçilmesi sağlanılarak DE algoritmasının arama performansı artırılmıştır.

İki Halkalı Optimum Zincirin Araştırılması: Bu çalışmada sadece iki halka içeren zincirler çalıştırılmıştır. Bu halkalardan bir tanesi DE algoritmasının baz mutasyon denklemidir. Diğer halka ise tasarlanmış halkalardan seçilecektir. Dolayısıyla yapılan çalışma, algoritmanın baz denklemiyle en uyumlu çalışan halkanın bulunması olarak da adlandırılabilir. Bu çalışmada 27 adet halka oluşturulmuştur. İçlerinden bir tanesi DE algoritmasının baz denklemi olduğundan, bu çalışmada 26 halka kullanılmıştır. 26 halkanın her biriyle bir zincir oluşturulmuştur. Zincir oluşturmada halkaların skorları dikkate alınmamıştır. Oluşturulan zincirler CEC20 10, 30, 50 ve 100 boyutlarında çalıştırılmıştır. Bu aşamada boyutlardaki optimum HED değerleri kullanılmıştır. Alınan sonuçlar Wilcoxon testi ile çalıştırılan boyutlardaki problem türlerine göre analiz edilmiştir. Bu durumda CEC20 problem setinde 10 adet problem bulunduğundan 4 boyut için toplamda 40 analiz yapılmıştır. Yapılan 40 analizin sonucunda, problem türü ve boyuta göre hangi zincirin kullanılması gerektiği belirlenmiştir. Tüm bu işlemlerin sonucunda tek bir durum oluşturularak, CEC20 problem seti için problem boyutu ve problem türüne göre otomatik zincir seçilmesi sağlanılarak, DE algoritmasının arama performansı artırılmıştır.

3. BULGULAR VE TARTIŞMA

Bu bölümde, önerilen yüzde hata değişimi esaslı arama yönteminin doğruluğunu, geçerliliğini ve DE algoritması üzerindeki performans etkilerini kapsamlı şekilde test etmek için farklı çalışmalar yapılmıştır. Yapılan çalışmaların ilk olarak en optimum HED değerlerinin belirlenmesi amaçlanmıştır. Bunun için CEC20 problem seti üzerinde 10, 30, 50 ve 100 boyutlarında çalıştırılan durumların sonuçları analiz edilmiştir. Belirlenen HED'lerden sonra skorları hesaplanmış halkalar ile zincirler oluşturulup CEC20 problem seti üzerinde çalıştırılmıştır. Bu çalışmada CEC20 problem setinde problem ve boyut bazlı en iyi zincirin belirlenmesi amaçlanmıştır. Bu kapsam bu çalışma ikili ve çoklu zincir çalışması olarak ikiye ayrılmıştır.

Yüzde hata değişimi etkini gerçek dünya problemlerindeki etkisini görmek içinde ikili zincir çalışması yapılmıştır.

Yapılan deneysel çalışmalar MATLAB®R2018a'da, Intel (R) Core™ i7-4770K CPU @ 3.50GHz ve 16 GB RAM ve x64 tabanlı işlemci üzerinde gerçekleştirilmiştir.

3.1. Deneysel Çalışma Çevresi

Literatürde sürekli değer optimizasyon problemleri genellikle dört ana kategoriye ayrılır: tek modlu, çok modlu, melez ve kompozisyon problemleri. Bu kategorizasyonlar, düzenli olarak gerçekleştirilen IEEE Evrimsel Hesaplama Konferanslarındaki (IEEE CEC) çalışmaların katkılarıyla şekillenmiştir (Suganthan, ve diğerleri, 2005; Liang, Qu, & Suganthan, 2013). MSA algoritmalarının performanslarını değerlendirmek ve rakip algoritmalarla karşılaştırarak arama etkinliklerini doğrulamak amacıyla, bu dört problem tipini içeren bir karşılaştırma problemleri havuzu oluşturulmaktadır.

Yüzde hata değişimi esaslı arama yönteminde ise geliştirilen yöntemin doğruluğunu ve geçerliliğini belirlemek için kısıtlı ve kısıtsız olmak üzere iki farklı problem seti kullanılmıştır:

Kısıtsız test problemlerinde CEC problem havuz seti olan CEC20 problem seti kullanılmıştır. Bu çalışmanın ayarlar ve adilliğinde CEC konferanslarındaki standartlar temel referans alınmıştır.

Kısıtlı test problemlerinde ise farklı kaynaklardan alınmış gerçek dünya kısıtlı mühendislik problemleri kullanılmıştır. Bu çalışmanın ayarlar ve adilliğinde CEC konferanslarındaki standartlar temel referans alınmıştır. Kısıt ihlali durumlarında ceza uygulaması yapılmıştır.

3.1.1. Ayarlar ve Adillik

Kısıtsız test problemleri çalışmasında, arama uzayı 10, 30, 50 ve 100 boyutlu olarak planlanmıştır. Bu sayede, algoritmaların küçük, orta ve büyük arama uzaylarındaki performansları kapsamlı bir şekilde test edilmiştir.

Kısıtlı test problemleri çalışmasında arama sürecinde uygunluk fonksiyonuna gönderilen parametrelerin kısıt ihlal durumları kontrol edilmiştir. Kısıtları ihlal oranları 10^6 değeriyle çarpılarak, elde edilen sonuç, uygunluk değeriyle toplanmıştır.

Kısıtsız ve kısıtlı problemlerinin arama sürecini sonlandırma kriteri olarak uygunluk fonksiyonunun (fitness function) çağrılma sayısı göz önünde bulundurulmuştur. Bu yaklaşım, algoritmalar arasında eşit fırsatlar ve adil bir karşılaştırma ortamı sağlamayı amaçlamaktadır. Uygunluk fonksiyonu, maksimum çağrılma sayısı olarak $10.000 \cdot d$ belirlenmiş olup, bu da problem boyutunun 10 bin katı olarak uygulanacaktır. Tüm bu ayarlamalarda CEC20 konferanslarındaki standartlar temel referans olarak kullanılmıştır.

Yapılan çalışmalarda HED kontrolü için periyot değeri 2 olarak belirlenmiştir.

3.1.2. Kısıtsız Test Problemleri

Kısıtsız test problemlerinde CEC20 problem havuzu kullanılacaktır. CEC20 de bulunan problemler Tablo 5’de görülmektedir. CEC20 problem havuzundaki problemlerin tamamı dinamik olarak boyutlandırılabilen problemlerle oluşturulmuştur. Böylelikle küçük boyutlu arama uzaylarında hızlı yakınsama özellikleri sayesinde başarılı görülen algoritmaların aynı problemlerin orta ve büyük boyutlu arama uzaylarındaki performanslarını da ortaya çıkarmak amaçlanmıştır.

Tablo 5. CEC20 problem seti

Problem Türü	Numara	Problem Adı	$Fi^* = Fi(x^*)$
Tek Modlu Fonksiyon (Unimodal Function)	1	Kaydırılmış ve Döndürülmüş Bükülmüş Puro Fonksiyonu (Shifted and Rotated Bent Cigar Function)	100
Temel Fonksiyonlar (Basic Functions)	2	Kaydırılmış ve Döndürülmüş Schwefel (Shifted and Rotated Schwefel's Function)	1100
	3	Kaydırılmış ve Döndürülmüş Lunacek bi-Rastrigin (Shifted and Rotated Lunacek bi-Rastrigin Function)	700
	4	Genişletilmiş Griewank plus Rosenbrock Fonksiyonu (Expanded Rosenbrock's plus Griewank's Function)	1900
Hibrit Fonksiyonlar (Hybrid Functions)	5	Hibrit Fonksiyon 1 (Hybrid Function 1) (N = 3)	1700
	6	Hibrit Fonksiyon 2 (Hybrid Function 2) (N = 4)	1600
	7	Hibrit Fonksiyon 3 (Hybrid Function 3) (N = 5)	2100
Kompozisyon Fonksiyonlar (Composition Functions)	8	Kompozisyon Fonksiyon 1 (Composition Function 1) (N = 3)	2200
	9	Kompozisyon Fonksiyon 2 (Composition Function 2) (N = 4)	2400
	10	Kompozisyon Fonksiyon 3 (Composition Function 3) (N = 5)	2500
Arama aralığı: [-100,100] D			

3.1.3. Gerçek Dünya Kısıtlı Mühendislik Test Problemleri

Gerçek dünya kısıtlı mühendislik test problemlerinde farklı kaynaklardan alınmış 10 adet problem bulunmaktadır. Arama sürecindeki rehber çözüm adaylarının belirlenen amaç fonksiyonunu minimize ederken aynı zamanda tasarım parametreleri için belirlenmiş

kısıtlarında sağlanması beklenmektedir. Kısıtlı mühendislik problemlerinin ayrıntıları Tablo 6’da verilmiştir. Problemlerin matematiksel tanımlamaları Ek-1’de verilmiştir.

Tablo 6. Kısıtlı mühendislik problemleri

Numara	Adı	Tasarım Parametresi Sayısı	Eşitlik Kısıtı Sayısı	Eşitsizlik Kısıtı Sayısı
1	Germe/Sıkıştırma Yayı Tasarımı (Kaveh & Dadras, 2017)	3	-	4
2	Kaynaklı Kiriş Tasarımı (Kaveh & Dadras, 2017)	4	-	7
3	Dişli Sistemi Tasarımı (Lin, Zhang, & Xu, 2018)	4	-	-
4	Üç Çubuk Kafes Tasarımı (Kumar, ve diğerleri, 2020)	2	-	3
5	Konsol Kiriş Tasarımı (Gupta & Deep, 2019)	5	-	1
6	Eşanjör Ağ Tasarımı (Kumar, ve diğerleri, 2020)	9	8	-
7	Harmanlama Havuzlama Ayırma Problemi (Kumar, ve diğerleri, 2020)	38	32	-
8	Propan, İzobütan, n-Bütan Keskin Olmayan Ayırma (Kumar, ve diğerleri, 2020)	48	38	-
9	Alkilasyon Süreci Optimizasyonu (Ravindran, Reklaitis, & Ragsdell, 2006)	10	7	-
10	Dört Kademeli Dişli Kutusu (Kumar, ve diğerleri, 2020)	22	-	86

3.1.4. İstatiksel Test Yöntemleri

Deneysel çalışmanın sonuçları, algoritmaların performanslarını değerlendirmek amacıyla istatistiksel test ve analiz yöntemleri kullanılarak incelenecektir. İki önemli test yöntemi olan Wilcoxon ve Friedman testleri, bu analizlerde temel araçlar olarak kullanılacaktır.

Wilcoxon testi, iki örneklem ortalaması arasındaki anlamlı farkları tespit etmeyi amaçlar. Eğer iki farklı algoritmanın çıkışlarının karşılaştırılması söz konusuysa, bu test, iki algoritmanın karşılıklı davranışlarını pratik bir şekilde değerlendirir.

Öte yandan, Friedman testi, çoklu karşılaştırmalarda algoritmaları, arama hatalarının performanslarına göre sıralamak için kullanılacaktır. Bu test, algoritmaların genel sıralamasını belirleyerek, hangi algoritmanın diğerlerine göre daha başarılı olduğunu ortaya koymak için etkili bir araçtır. Friedman skoru düşük olan en iyi sonucu veren algoritmadır.

Bu istatistiksel testler, deneysel sonuçların güvenilir bir şekilde değerlendirilmesine ve algoritmaların performanslarının karşılaştırılmasına yönelik objektif bir analiz sağlamak amacıyla kullanılacaktır.

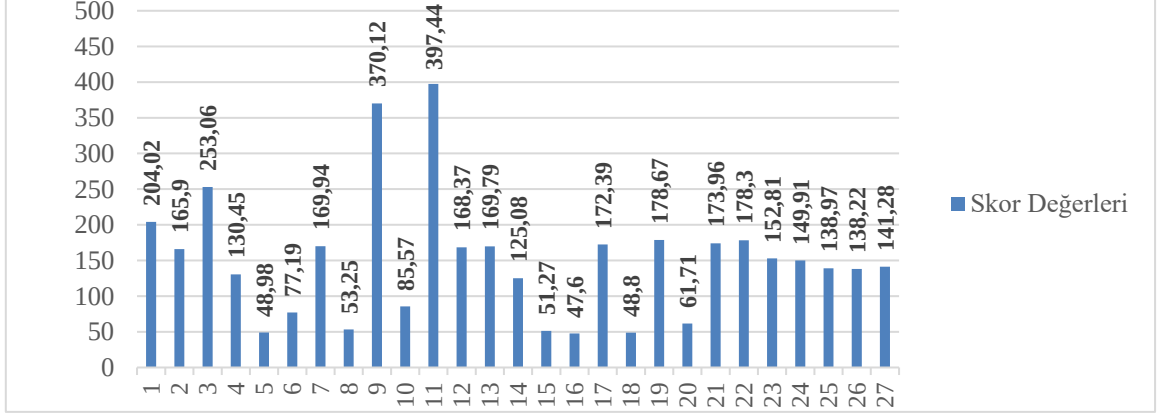
3.2. Optimum Yüzde Hata Değişim Oranının Belirlenmesi

HED'lerin belirlenmesinde CEC20 problem seti üzerinde 10, 30, 50 ve 100 boyutlarında çalışılmıştır. HED için aşağıdaki oranlar belirlenmiştir;

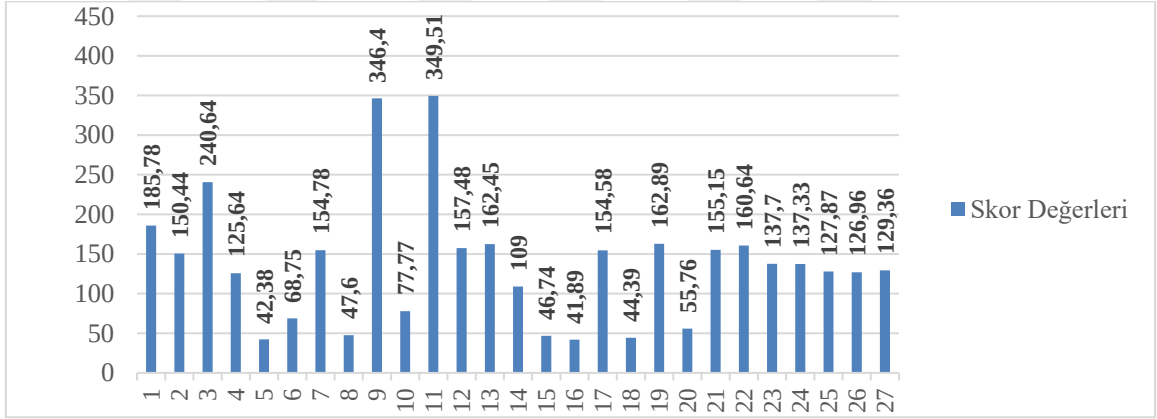
- $HED = \{0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1, 2, 5\}$

Her bir HED oranı için oluşturulan 27 adet halkanın skorları hesaplanmıştır. Skor hesabı için halkalar CEC20 problem seti üzerinde 10, 30, 50 ve 100 boyutunda çalıştırılmıştır. Sonuçlar Şekil 20 ve Şekil 28 arasındaki şekillerde verilmiştir.

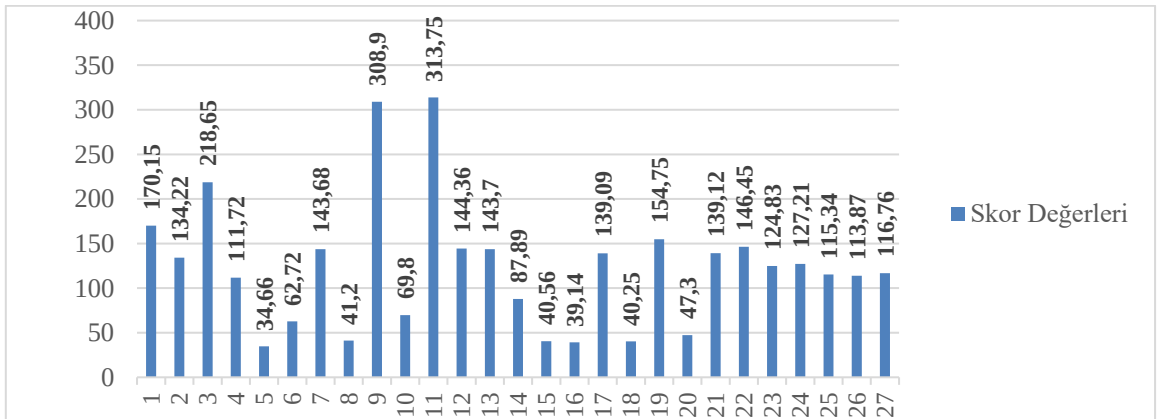
Şekil 20. 0.01 HED değeri için halka skorları



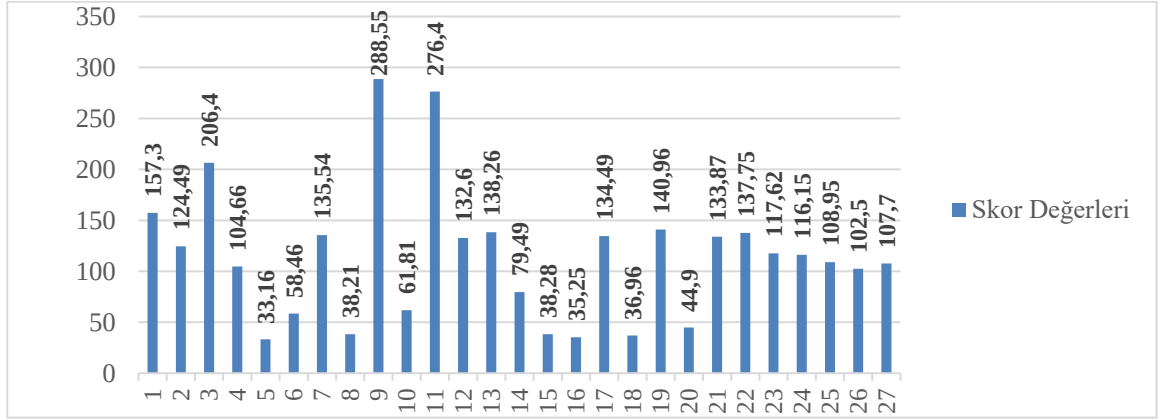
Şekil 21. 0.02 HED değeri için halka skorları



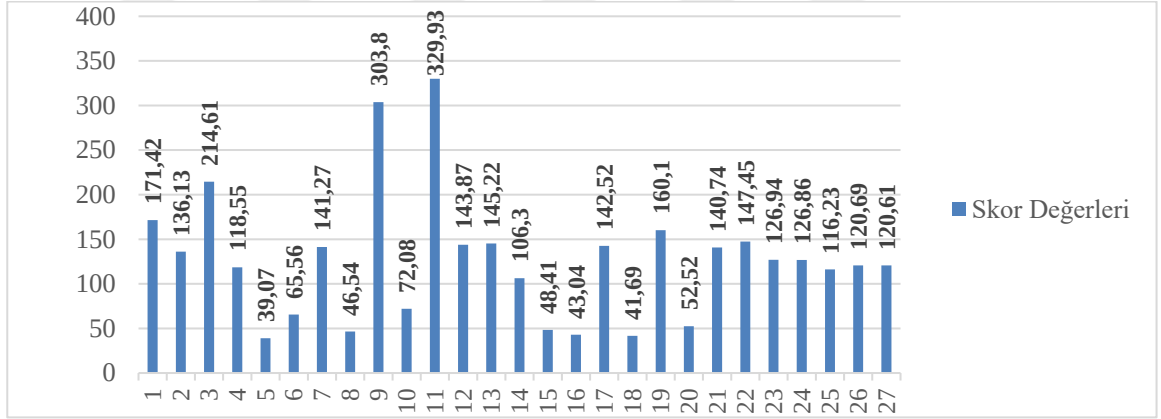
Şekil 22. 0.05 HED değeri için halka skorları



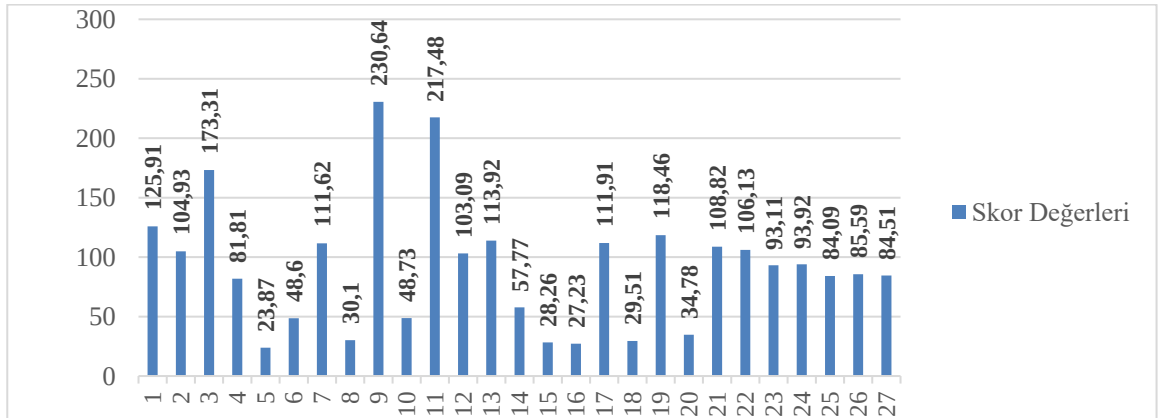
Şekil 23. 0.1 HED değeri için halka skorları



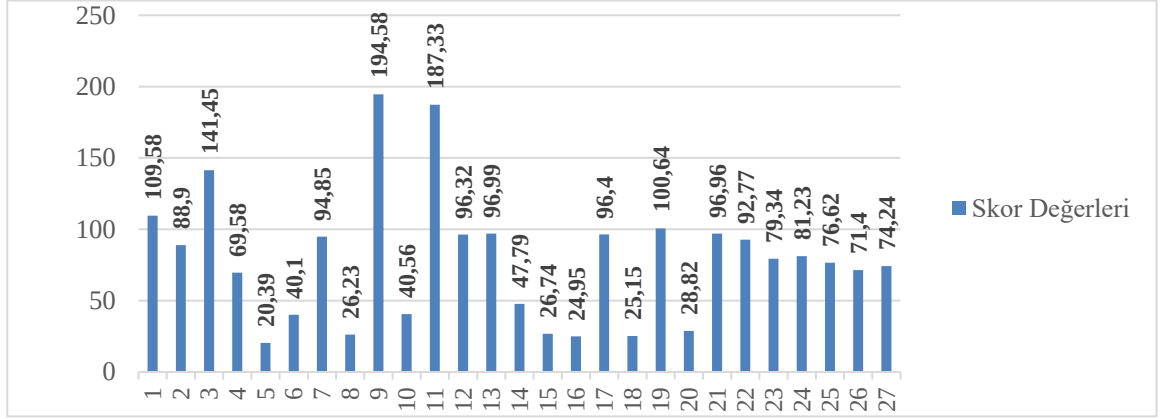
Şekil 24. 0.2 HED değeri için halka skorları



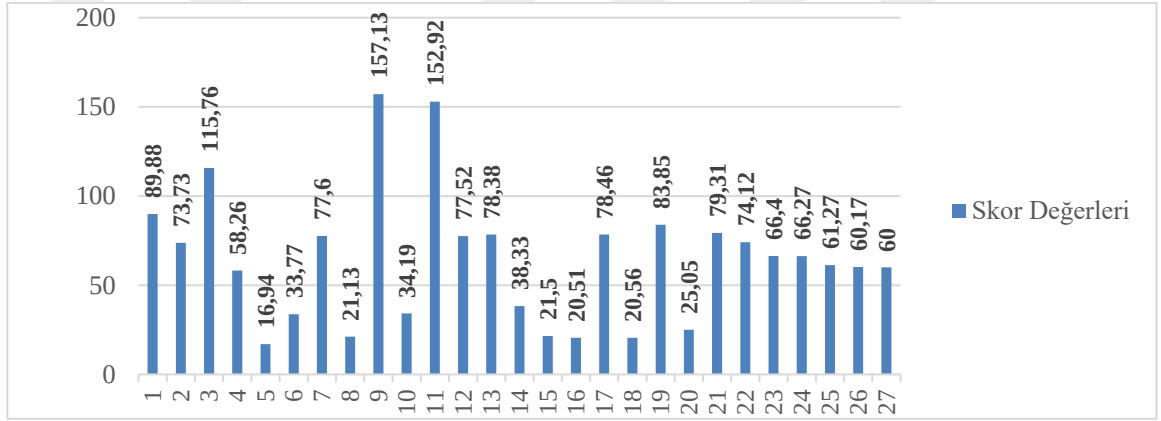
Şekil 25. 0.5 HED değeri için halka skorları



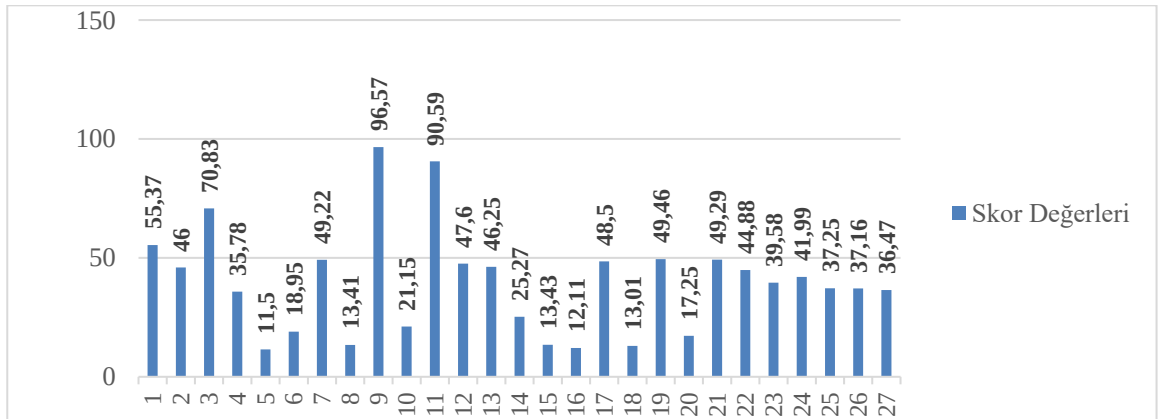
Şekil 26. 1 HED değeri için halka skorları



Şekil 27. 2 HED değeri için halka skorları



Şekil 28. 5 HED değeri için halka skorları



Yukarıdaki şekillerde her HED değeri için ilk üç sırada yer alan halkalar kullanılarak DE algoritması için zincirler oluşturulmuştur. Oluşturulan zincirler ile DE algoritması CEC20 problem setinde 10, 30, 50 ve 100 boyutta çalıştırılarak Friedman ve Wilcoxon analizleri yapılmıştır. Tablo 7 ve Tablo 8’da analiz sonuçları görülmektedir.

Tablo 7. HED değerleri için Wilcoxon testi ikili karşılaştırma sonuçları

	DE vs.+/-/-			
HED	D=10	D=30	D=50	D=100
0.02	8/1/1	8/1/1	8/1/1	8/1/1
0.1	6/3/1	8/0/2	8/1/1	8/1/1
0.2	7/2/1	8/1/1	8/1/1	8/1/1
0.01	7/3/0	8/1/1	8/1/1	8/1/1
5	7/2/1	8/0/2	8/1/1	8/1/1
2	6/3/1	8/0/2	8/1/1	8/1/1
1	6/3/1	8/1/1	8/1/1	8/1/1
0.5	6/4/0	8/0/2	8/1/1	8/1/1
0.05	6/3/1	8/0/2	8/1/1	8/1/1

Tablo 8. HED değerleri için Friedman analiz sonuçları

	Problem Boyutu				
HED	10	30	50	100	Ortalama
0.02	5.15	5.05	5.17	5.09	5.12
0.1	5.19	5.18	5.05	5.12	5.14
0.2	5.43	5.11	5.04	5.13	5.18
0.01	5.12	5.24	5.38	5.07	5.20
5	5.36	5.15	5.03	5.24	5.20
2	5.20	5.41	5.09	5.14	5.21
1	5.46	5.15	5.21	5.18	5.25
0.5	5.37	5.15	5.48	5.15	5.29
0.05	5.28	5.46	5.45	5.21	5.35
DE	7.44	8.08	8.11	8.68	8.08

Friedman tablosu incelendiğinde CEC20 problemlerinde kullanılacak HED oranları

- 10 boyut için 0.01
- 30 boyut için 0.02
- 50 boyut için 5
- 100 boyut için 0.01

olmaktadır. Yapılacak tüm CEC20 çalışmalarında bu oranlar kullanılacaktır.

3.3. Optimum Zincirlerin Araştırılması

Bu bölümde, çok halkalı ve iki halkalı zincirlerin performanslarına odaklanılmıştır. CEC20 problem seti üzerinde gerçekleştirilen deneysel çalışmalarda, çok halkalı ve iki halkalı zincirler arasında yapılan iki farklı çalışmada optimum zincirlerin belirlenmeye çalışılmıştır. Çok halkalı ve iki halkalı çalışmaların sonuçları, birbirleriyle karşılaştırılarak en üst düzeyde başarı sağlayan çalışmanın belirlenmesi amaçlanmıştır. Ayrıca, bu çalışmalarda belirlenen optimum zincirler, literatürde yer alan 51 rakip algoritma ile kıyaslanarak DE algoritmasına uygulanan yüzde hata değişimi yönteminin etkisi vurgulanmıştır.

3.3.1. Çok Halkalı Optimum Zincir Araştırılması

Daha önceki başlıklarda yapılan çalışmalarda CEC20 problem setinde, problem boyalarında kullanılacak HED değerleri belirlenmiştir. 3.2 bölümünün altındaki şekillerde ise HED değerleri için halkaların skorları verilmiştir. Boyutlarda seçilen HED değerlerine göre en iyi skor değerine sahip ilk 10 halka seçilmiştir. Seçilen bu halkalardan farklı boyutlar için toplam 30 durum oluşturulmuştur. Oluşturulan durumlar Tablo 9, Tablo 10 ve Tablo 11’de görülmektedir.

Tablo 9. 0.01 HED değeri için 10 ve 100 boyutlarında oluşturulan çok halkalı zincir durumları

Durum Adı	Zincirdeki halka numaraları
Durum 1	11
Durum 2	11- 9
Durum 3	11- 9- 3
Durum 4	11- 9- 3-1
Durum 5	11- 9- 3-1-19
Durum 6	11- 9- 3-1-19-22
Durum 7	11- 9- 3-1-19-22-21
Durum 8	11- 9- 3-1-19-22-21-17
Durum 9	11- 9- 3-1-19-22-21-17-7
Durum 10	11- 9- 3-1-19-22-21-17-7-13

Tablo 10. 0.02 HED değeri için 30 boyutunda oluşturulan çok halkalı zincir durumları

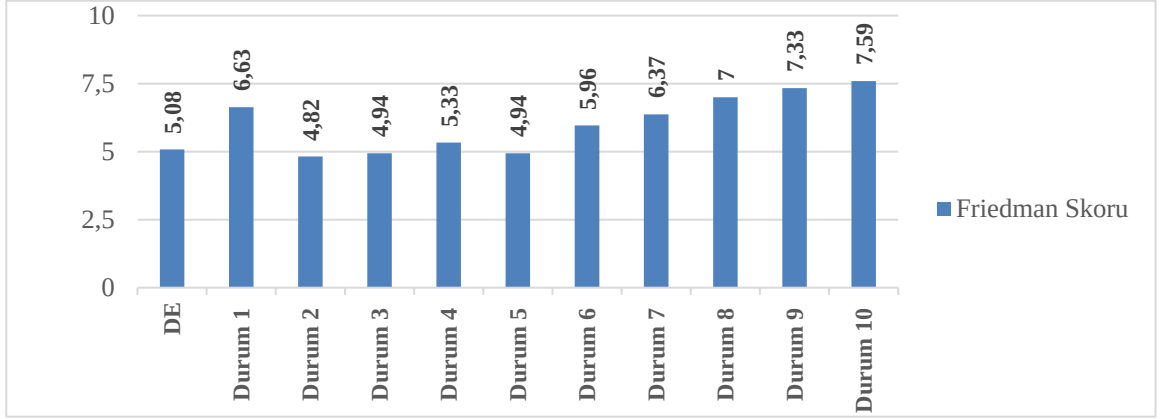
Durum Adı	Zincirdeki halka numaraları
Durum 11	11
Durum 12	11 - 9
Durum 13	11 - 9 - 3
Durum 14	11 - 9 - 3 - 1
Durum 15	11 - 9 - 3 - 1 - 19
Durum 16	11 - 9 - 3 - 1 - 19 - 13
Durum 17	11 - 9 - 3 - 1 - 19 - 13 - 22
Durum 18	11 - 9 - 3 - 1 - 19 - 13 - 22 - 12
Durum 19	11 - 9 - 3 - 1 - 19 - 13 - 22 - 12 - 21
Durum 20	11 - 9 - 3 - 1 - 19 - 13 - 22 - 12 - 21 - 7

Tablo 11. 5 HED değeri için 50 boyutunda oluşturulan çok halkalı zincir durumları

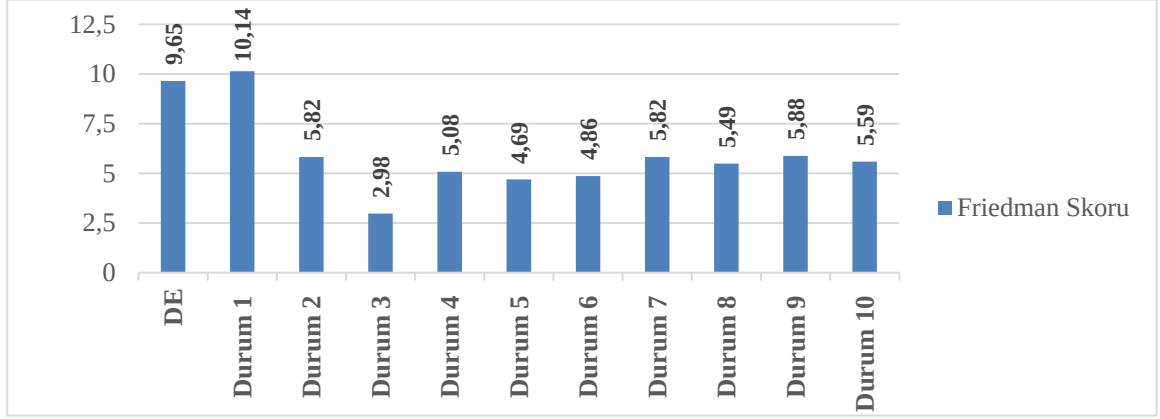
Durum Adı	Zincirdeki halka numaraları
Durum 21	11
Durum 22	11 - 9
Durum 23	11 - 9 - 3
Durum 24	11 - 9 - 3 - 1
Durum 25	11 - 9 - 3 - 1 - 19
Durum 26	11 - 9 - 3 - 1 - 19 - 21
Durum 27	11 - 9 - 3 - 1 - 19 - 21 - 7
Durum 28	11 - 9 - 3 - 1 - 19 - 21 - 7 - 17
Durum 29	11 - 9 - 3 - 1 - 19 - 21 - 7 - 17 - 12
Durum 30	11 - 9 - 3 - 1 - 19 - 21 - 7 - 17 - 12 - 13

Farklı boyutlar için oluşturulan durumlar CEC20 problem setinde boyut ve problem bazlı olarak çalıştırılarak analiz edilmiştir. Yapılan çalışmaların analiz sonuçları Şekil 29 ve Şekil 68 arasındaki şekillerde verilmiştir.

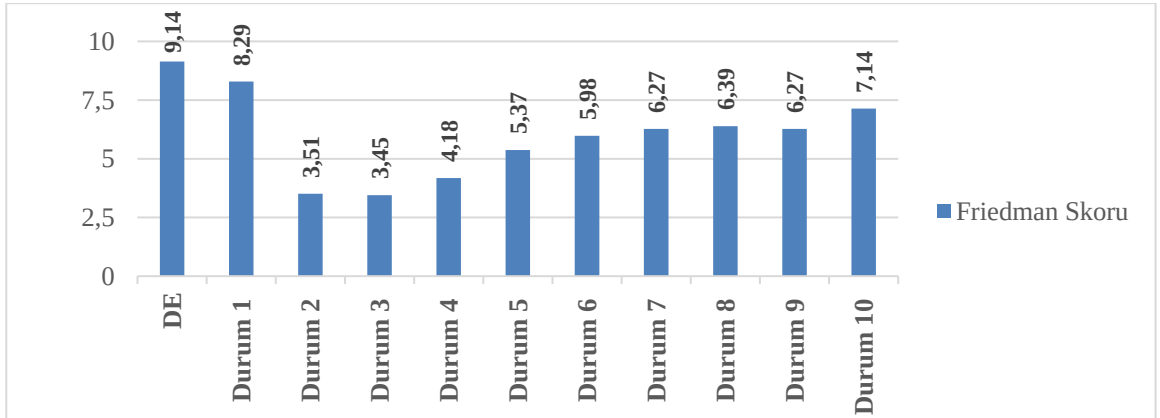
Şekil 29. CEC20 10D 1.problem çok halkalı durumlar için Friedman analiz sonuçları



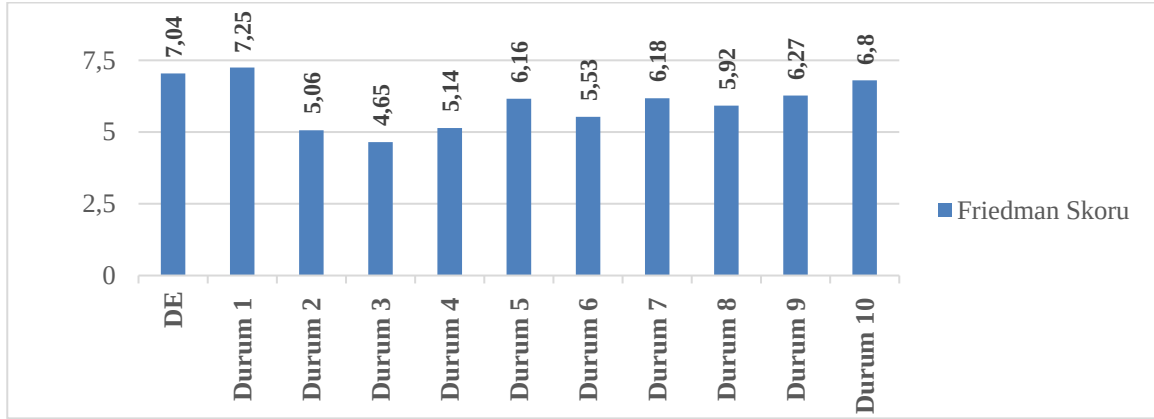
Şekil 30. CEC20 10D 2.problem çok halkalı durumlar için Friedman analiz sonuçları



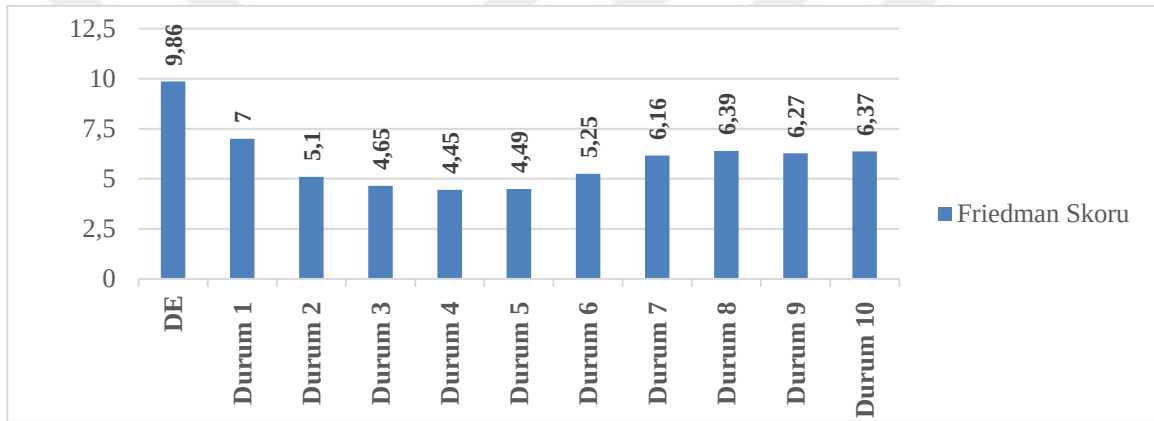
Şekil 31. CEC20 10D 3.problem çok halkalı durumlar için Friedman analiz sonuçları



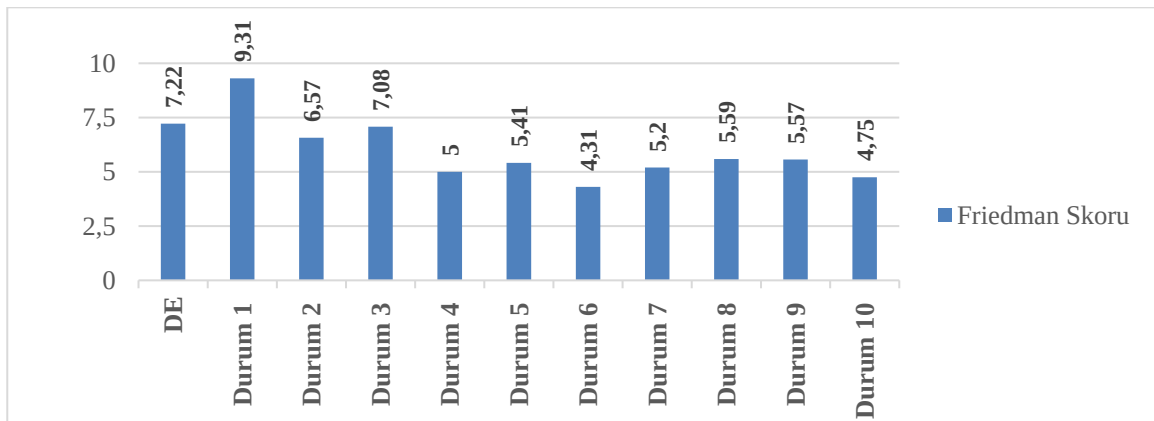
Şekil 32. CEC20 10D 4.problem çok halkalı durumlar için Friedman analiz sonuçları



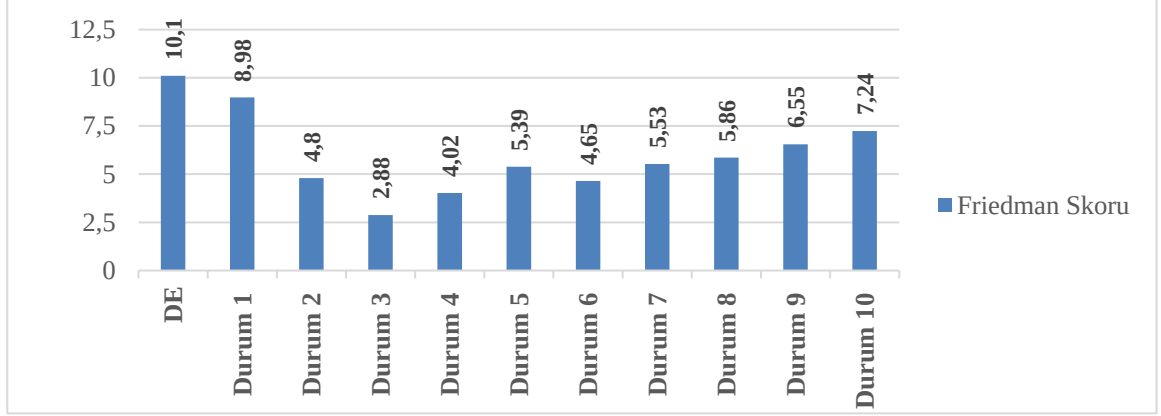
Şekil 33. CEC20 10D 5.problem çok halkalı durumlar için Friedman analiz sonuçları



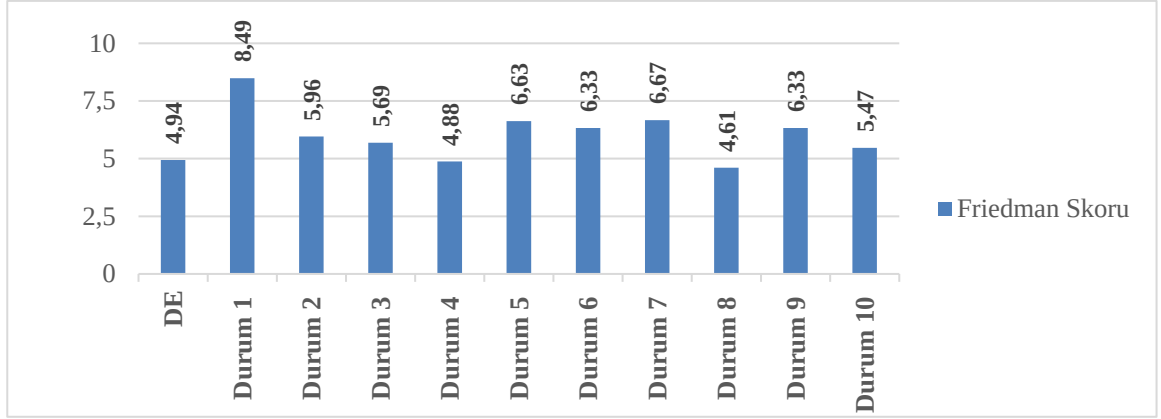
Şekil 34. CEC20 10D 6.problem çok halkalı durumlar için Friedman analiz sonuçları



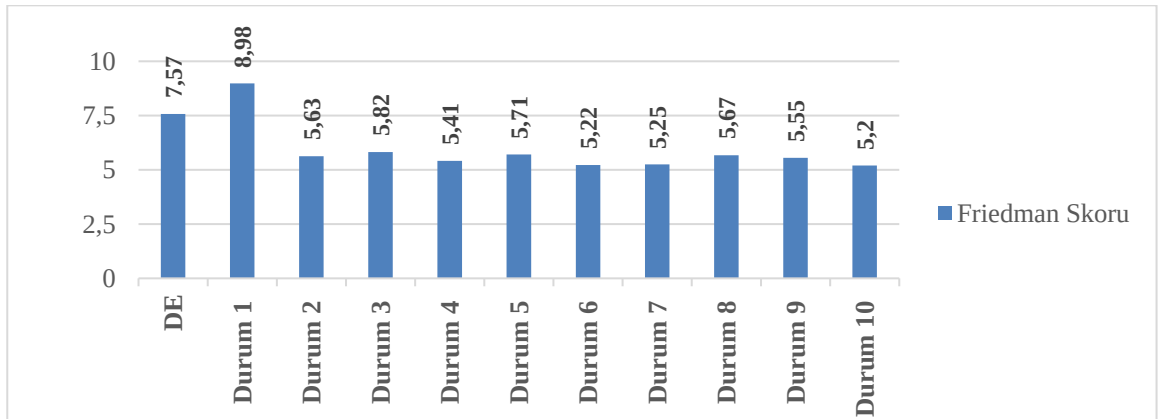
Şekil 35. CEC20 10D 7.problem çok halkalı durumlar için Friedman analiz sonuçları



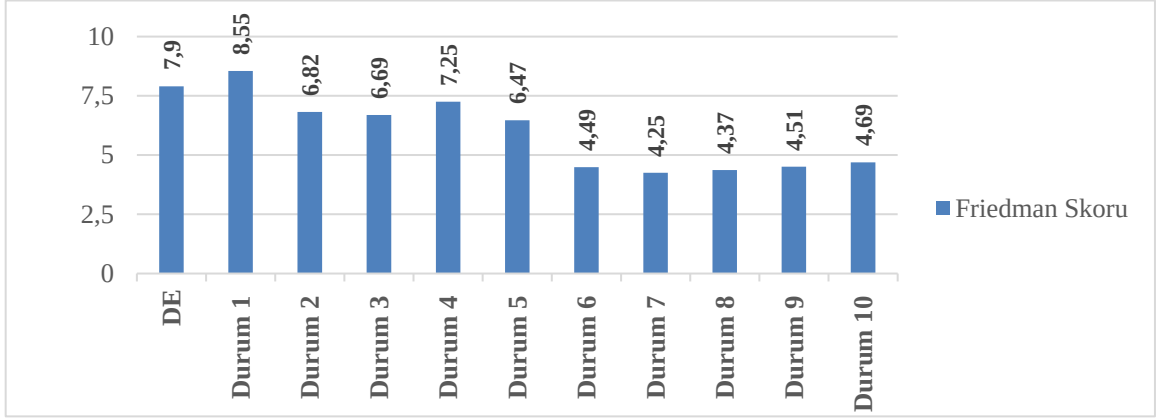
Şekil 36. CEC20 10D 8.problem çok halkalı durumlar için Friedman analiz sonuçları



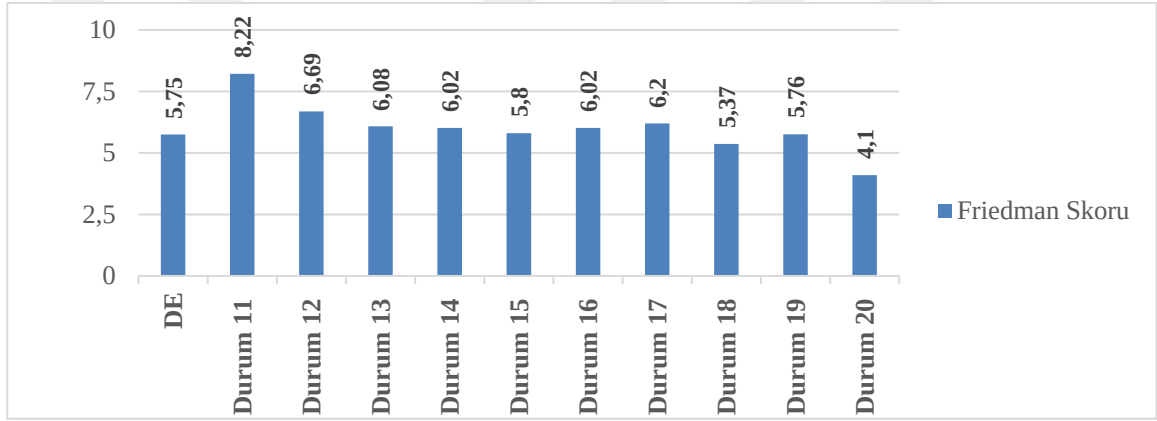
Şekil 37. CEC20 10D 9.problem çok halkalı durumlar için Friedman analiz sonuçları



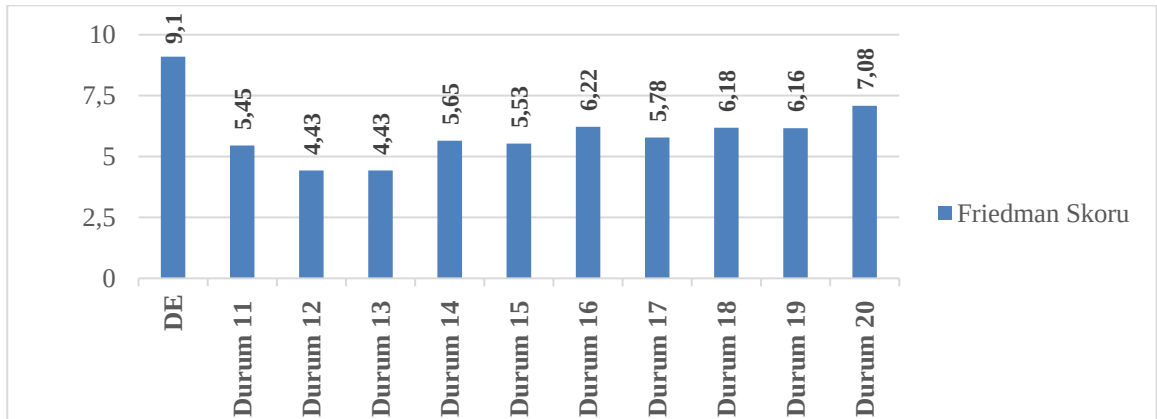
Şekil 38. CEC20 10D 10.problem çok halkalı durumlar için Friedman analiz sonuçları



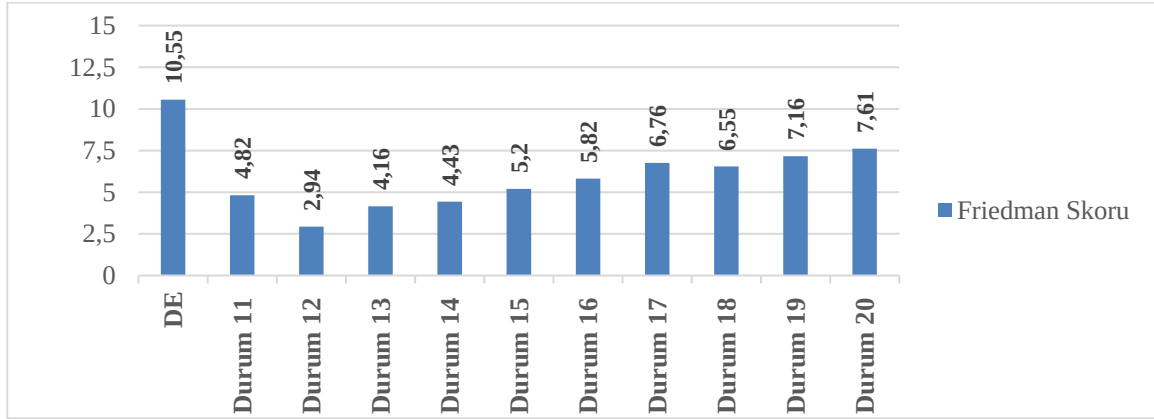
Şekil 39. CEC20 30D 1.problem çok halkalı durumlar için Friedman analiz sonuçları



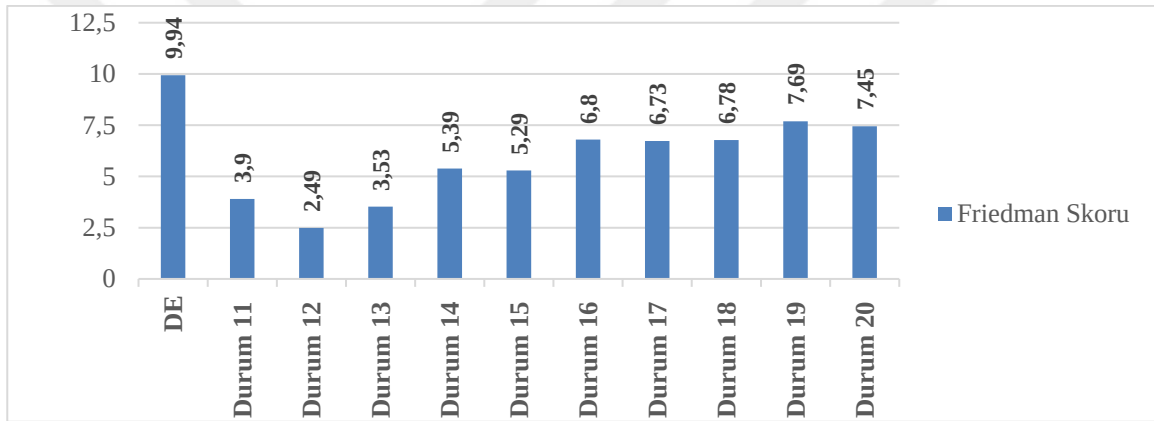
Şekil 40. CEC20 30D 2.problem çok halkalı durumlar için Friedman analiz sonuçları



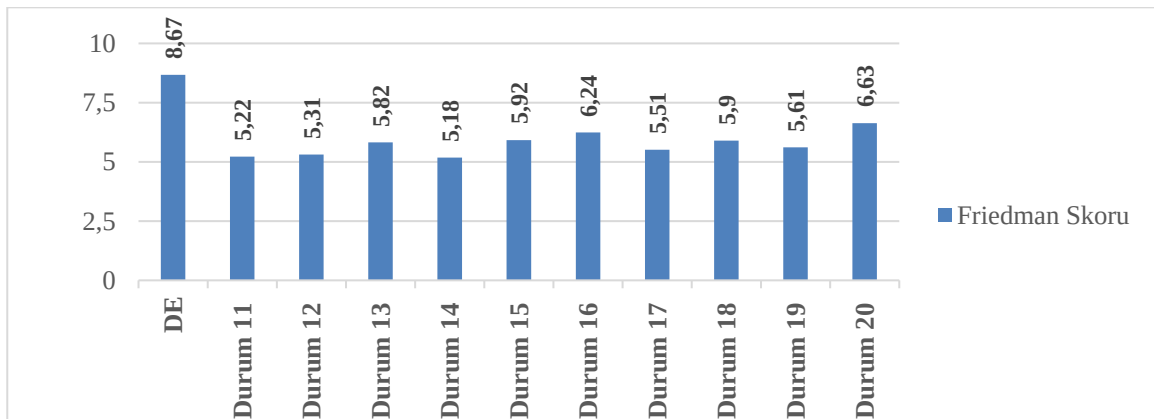
Şekil 41. CEC20 30D 3.problem çok halkalı durumlar için Friedman analiz sonuçları



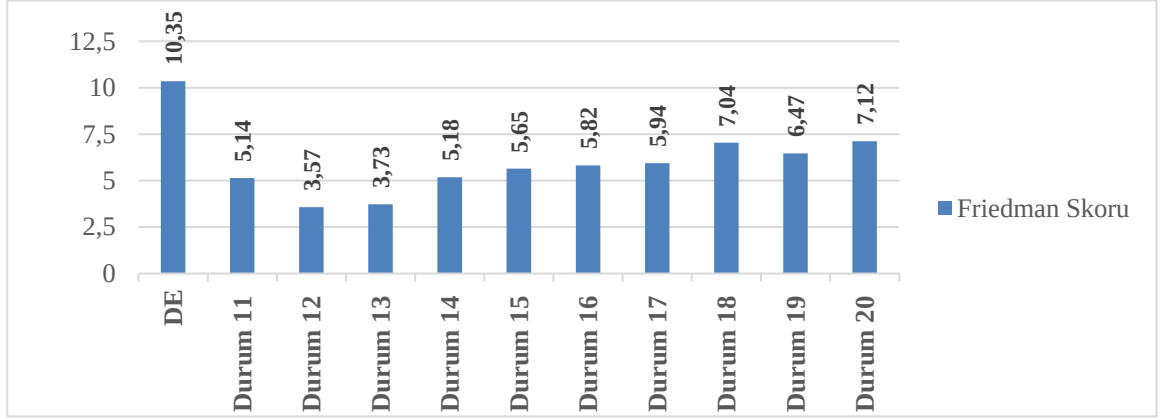
Şekil 42. CEC20 30D 4.problem çok halkalı durumlar için Friedman analiz sonuçları



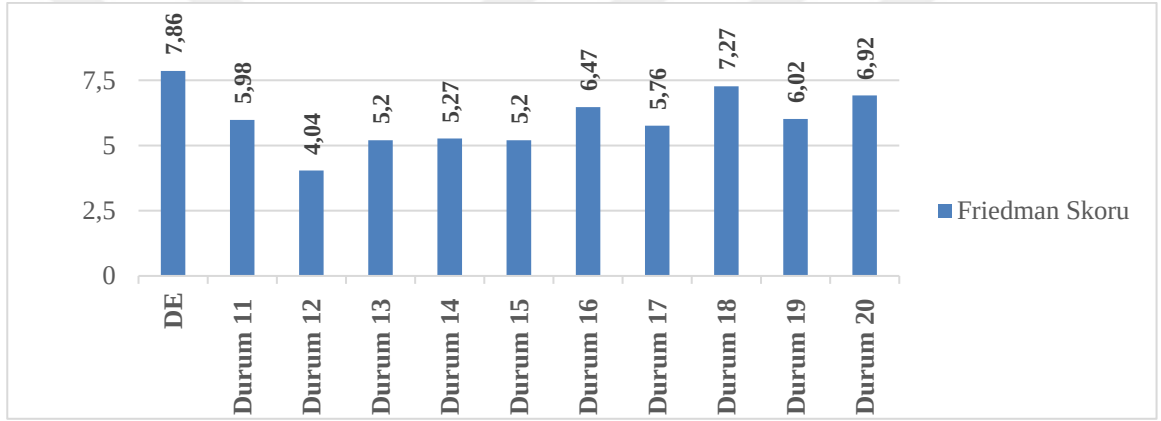
Şekil 43. CEC20 30D 5.problem çok halkalı durumlar için Friedman analiz sonuçları



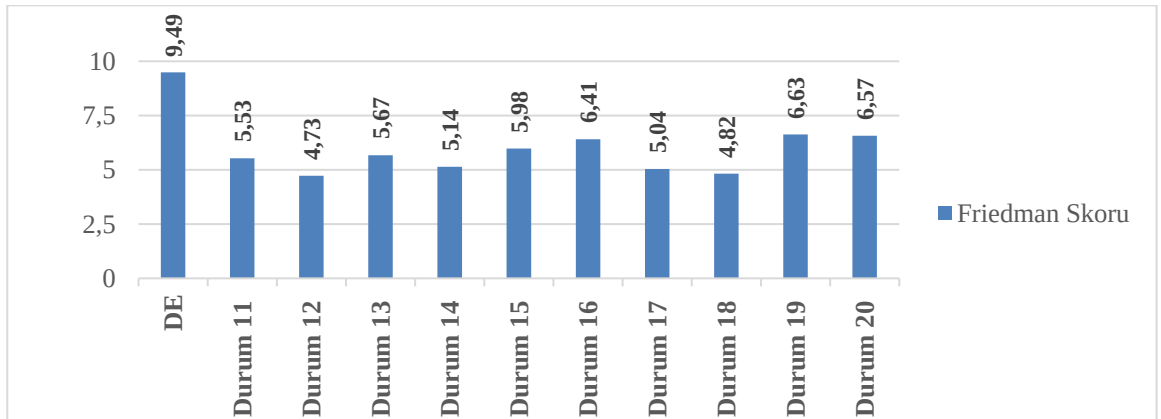
Şekil 44. CEC20 30D 6.problem çok halkalı durumlar için Friedman analiz sonuçları



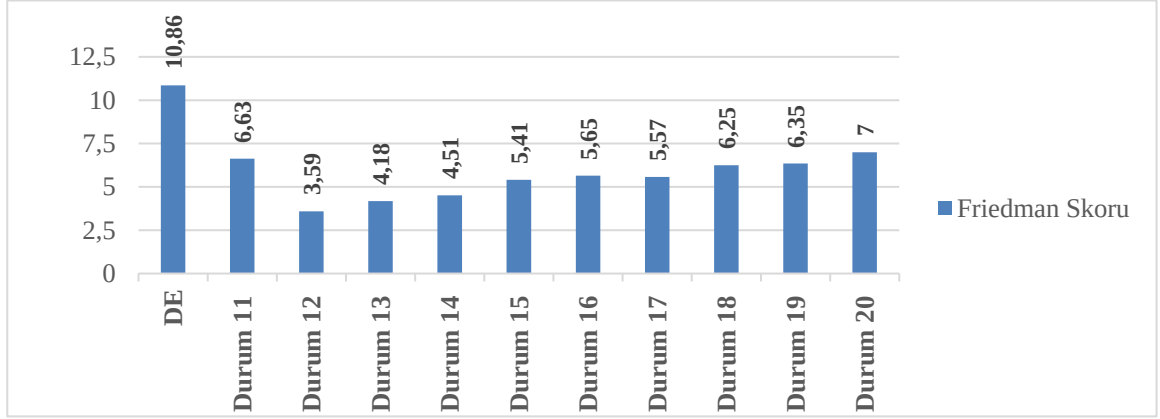
Şekil 45. CEC20 30D 7.problem çok halkalı durumlar için Friedman analiz sonuçları



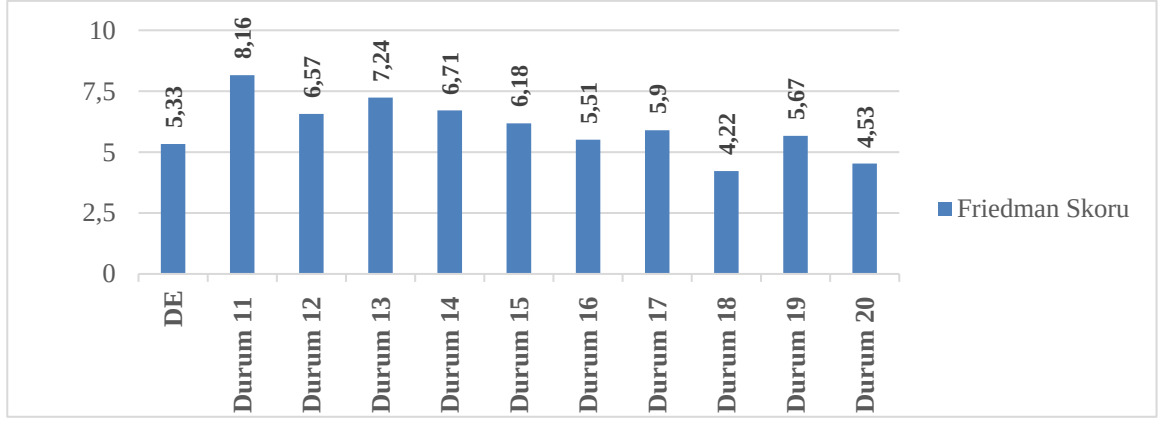
Şekil 46. CEC20 30D 8.problem çok halkalı durumlar için Friedman analiz sonuçları



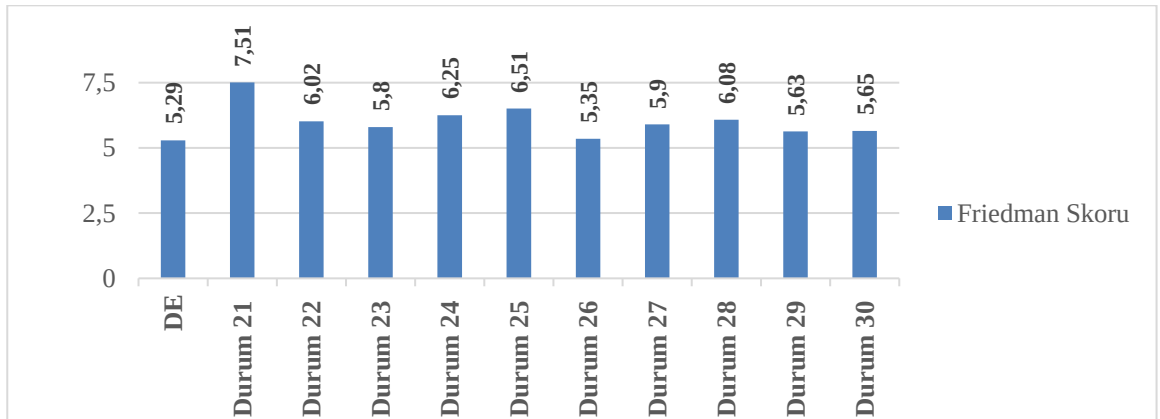
Şekil 47. CEC20 30D 9.problem çok halkalı durumlar için Friedman analiz sonuçları



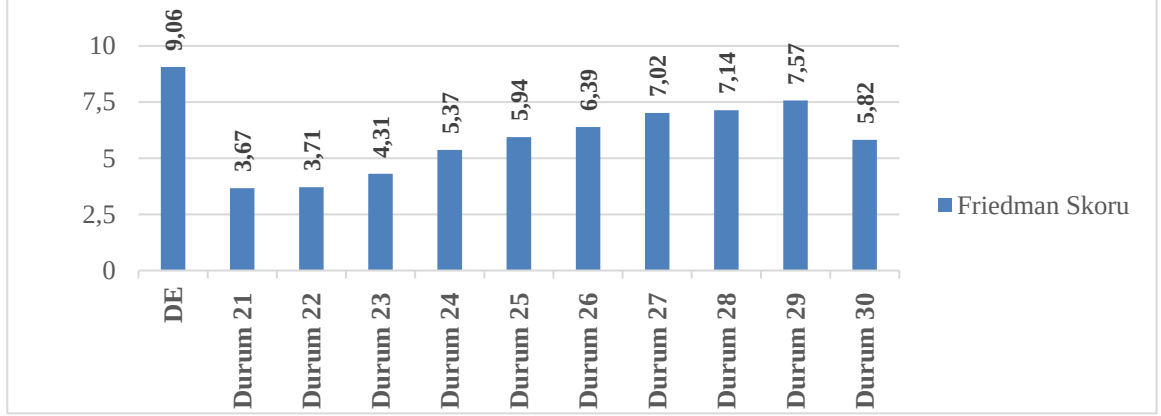
Şekil 48. CEC30 10D 10.problem çok halkalı durumlar için Friedman analiz sonuçları



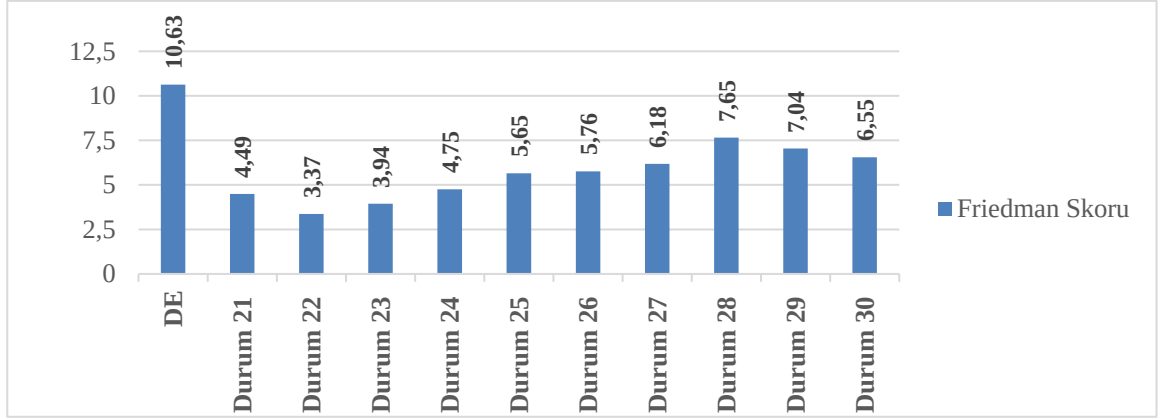
Şekil 49. CEC20 50D 1.problem çok halkalı durumlar için Friedman analiz sonuçları



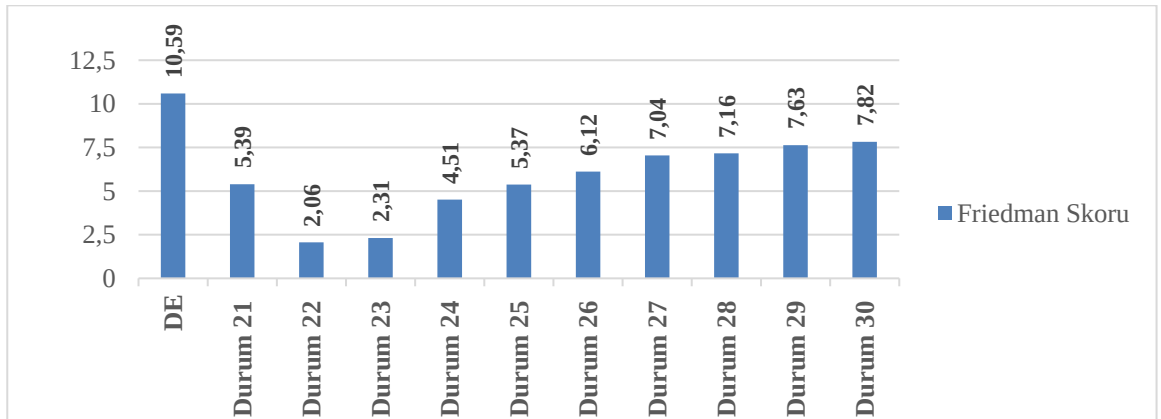
Şekil 50. CEC20 50D 2.problem çok halkalı durumlar için Friedman analiz sonuçları



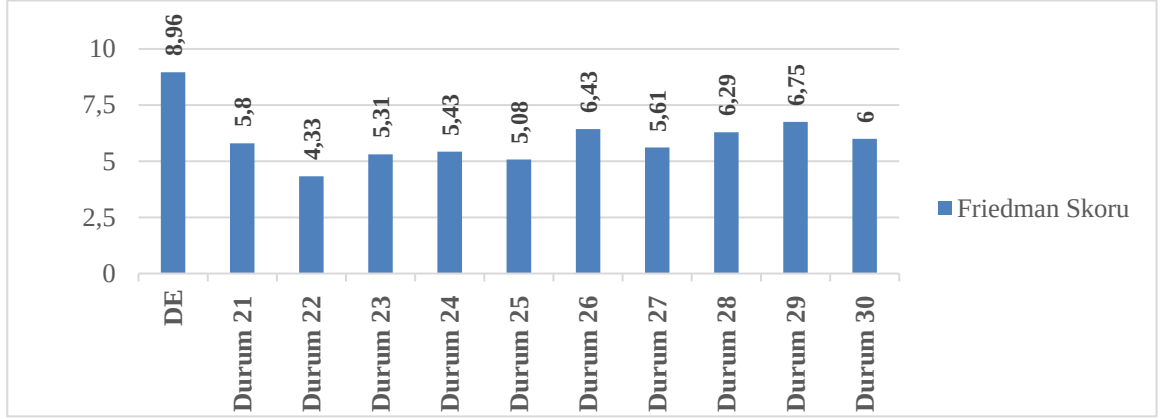
Şekil 51. CEC20 50D 3.problem çok halkalı durumlar için Friedman analiz sonuçları



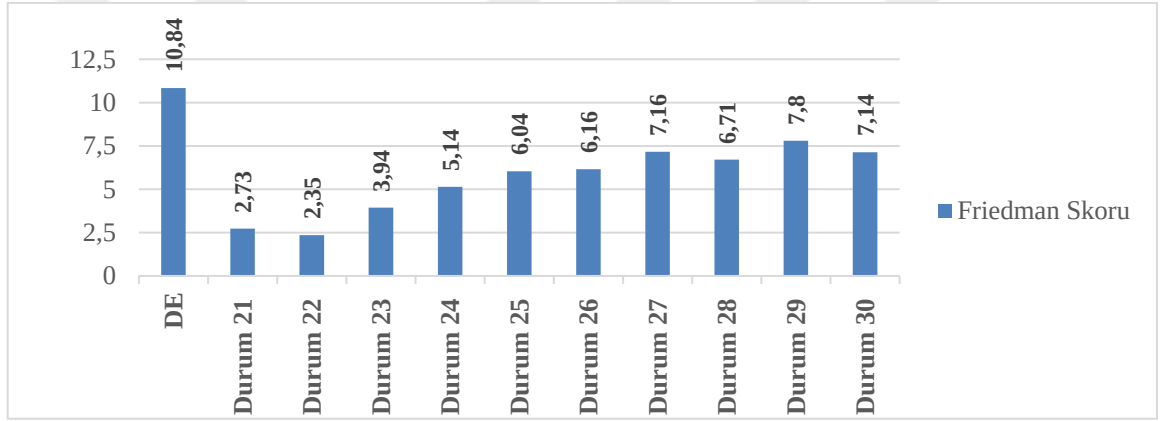
Şekil 52. CEC20 50D 4.problem çok halkalı durumlar için Friedman analiz sonuçları



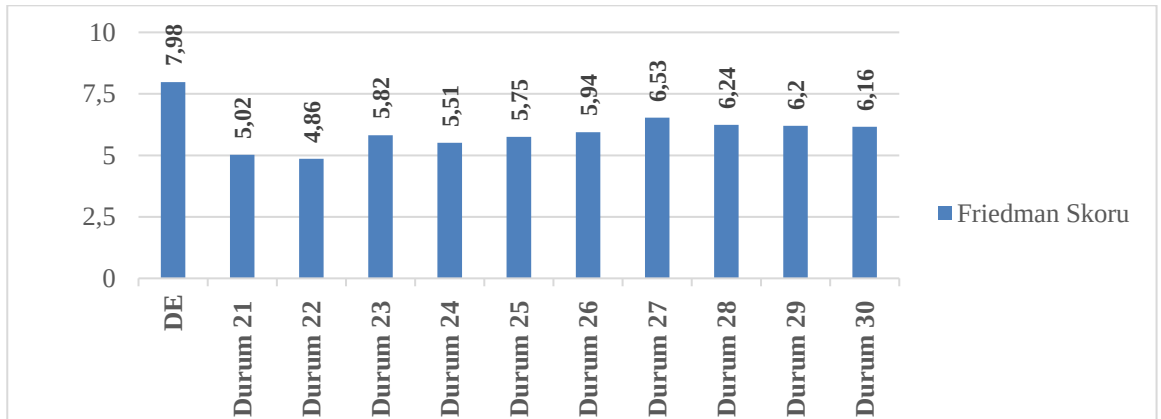
Şekil 53. CEC50 10D 5.problem çok halkalı durumlar için Friedman analiz sonuçları



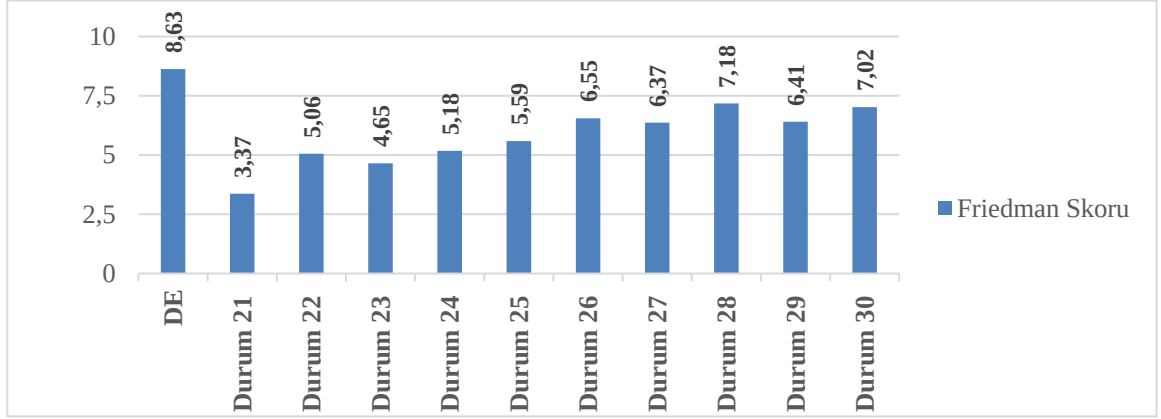
Şekil 54. CEC20 50D 6.problem çok halkalı durumlar için Friedman analiz sonuçları



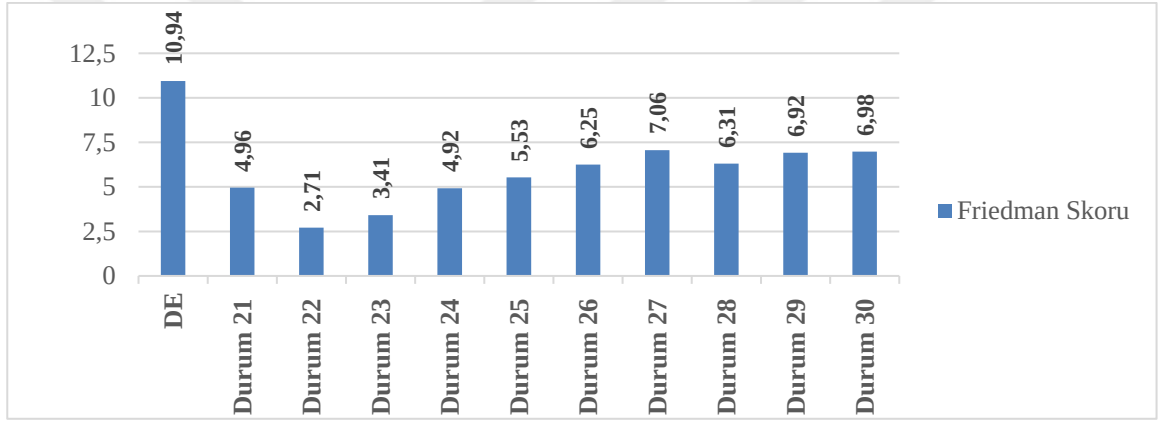
Şekil 55. CEC20 50D 7.problem çok halkalı durumlar için Friedman analiz sonuçları



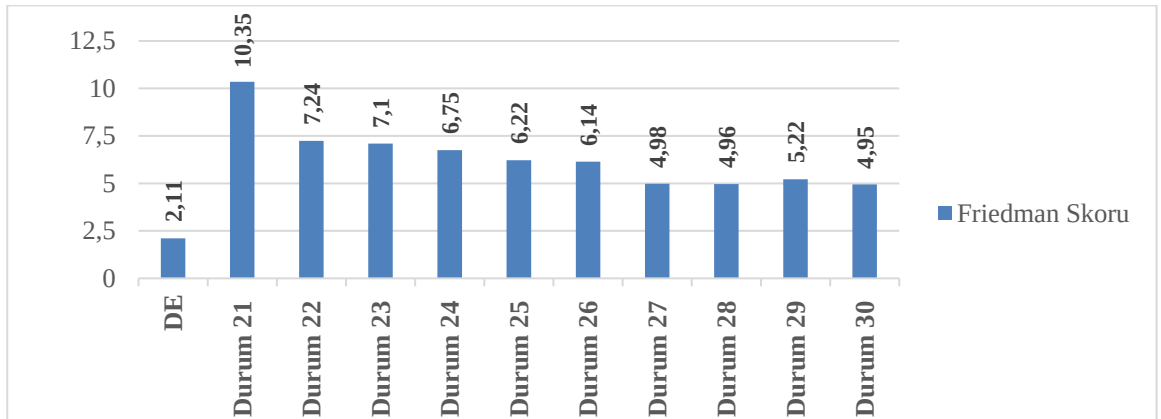
Şekil 56. CEC50 10D 8.problem çok halkalı durumlar için Friedman analiz sonuçları



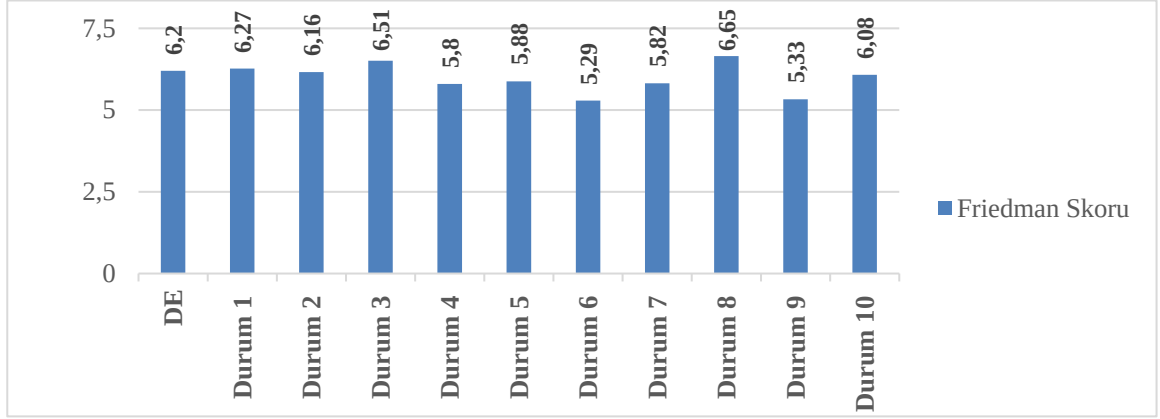
Şekil 57. CEC20 50D 9.problem çok halkalı durumlar için Friedman analiz sonuçları



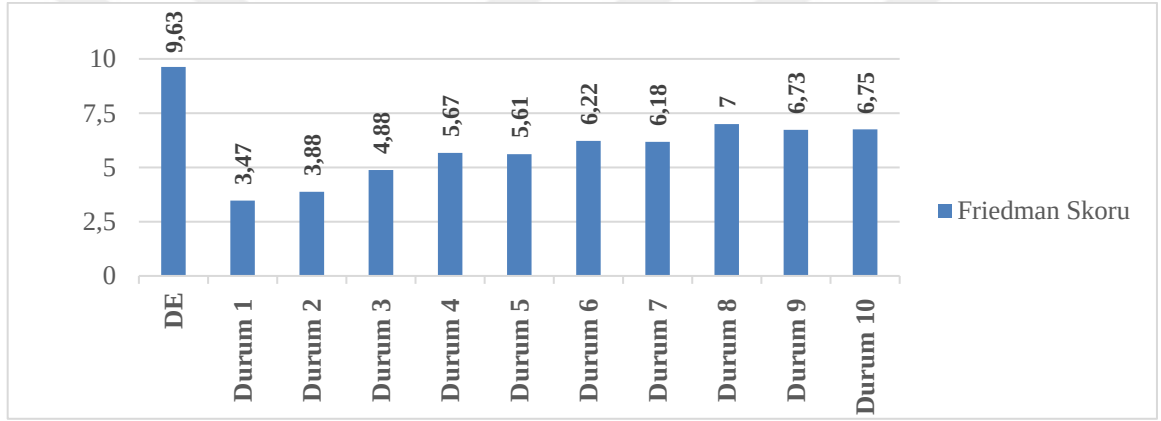
Şekil 58. CEC20 50D 10.problem çok halkalı durumlar için Friedman analiz sonuçları



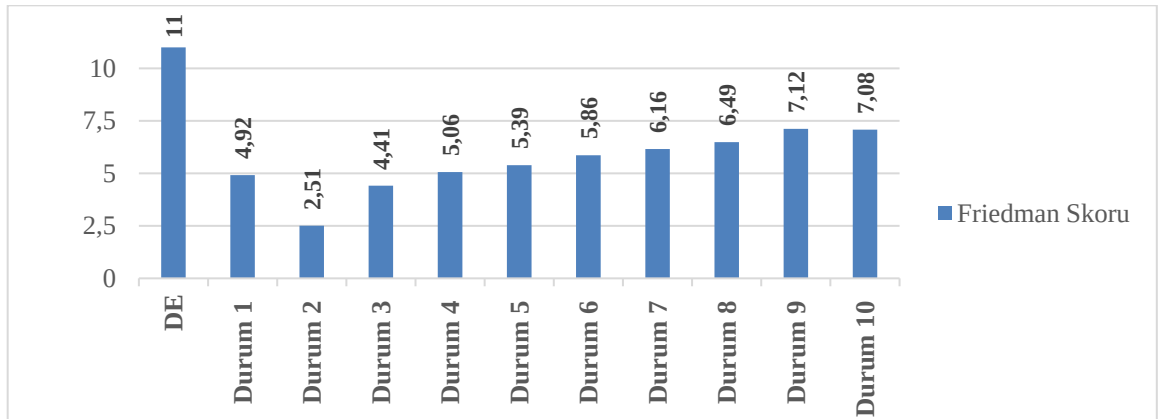
Şekil 59. CEC20 100D 1.problem çok halkalı durumlar için Friedman analiz sonuçları



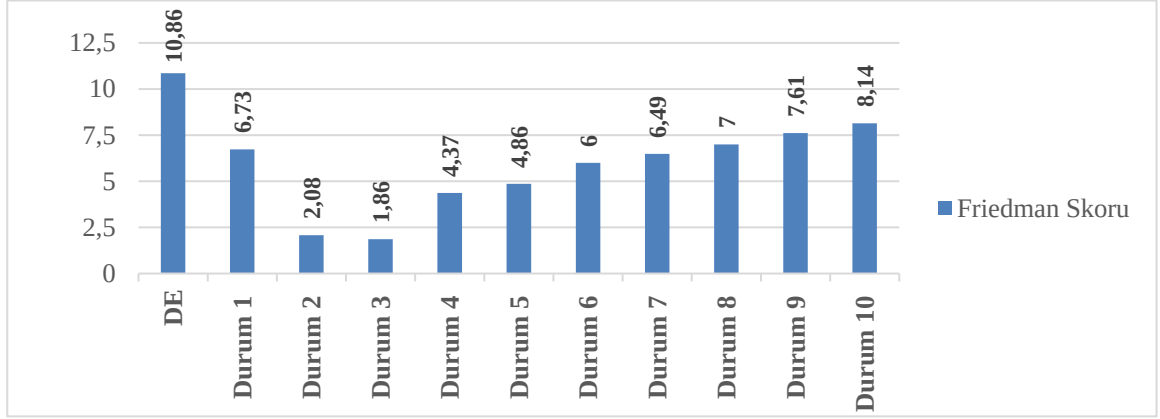
Şekil 60. CEC20 100D 2.problem çok halkalı durumlar için Friedman analiz sonuçları



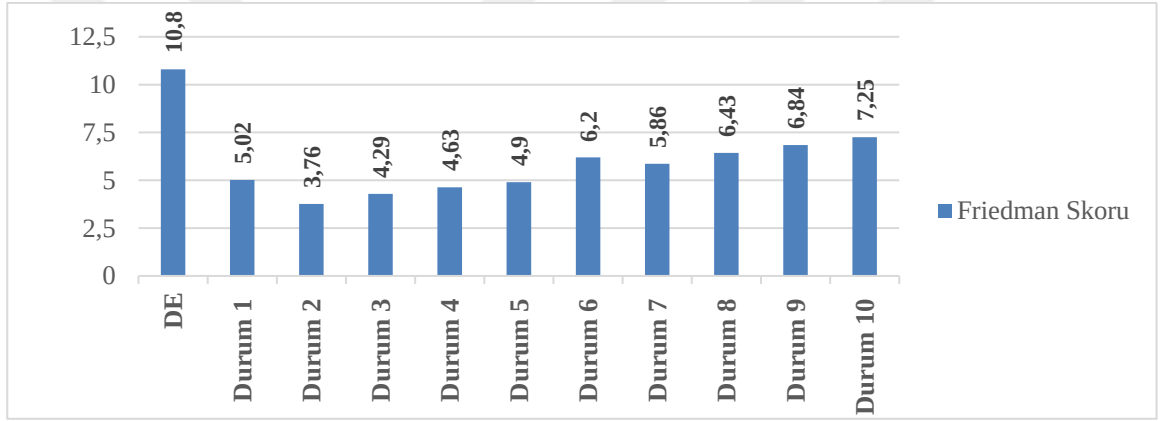
Şekil 61. CEC20 100D 3.problem çok halkalı durumlar için Friedman analiz sonuçları



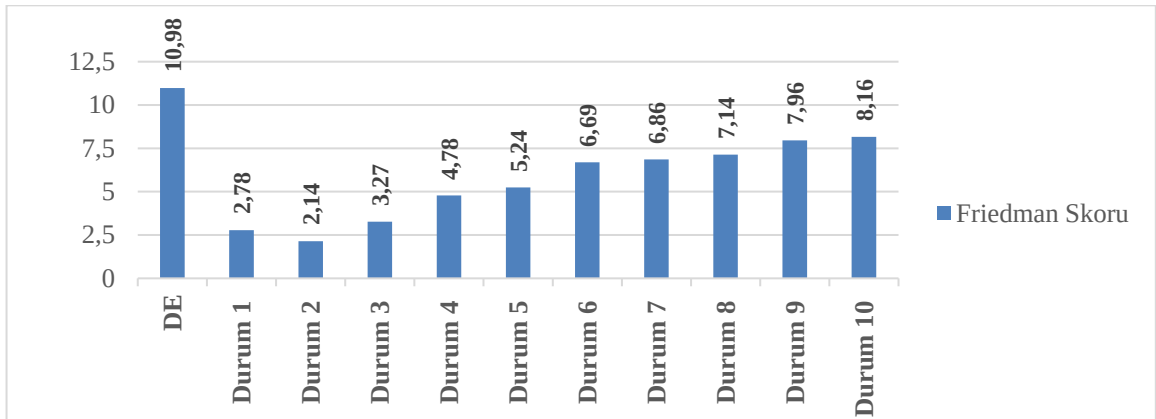
Şekil 62. CEC20 100D 4.problem çok halkalı durumlar için Friedman analiz sonuçları



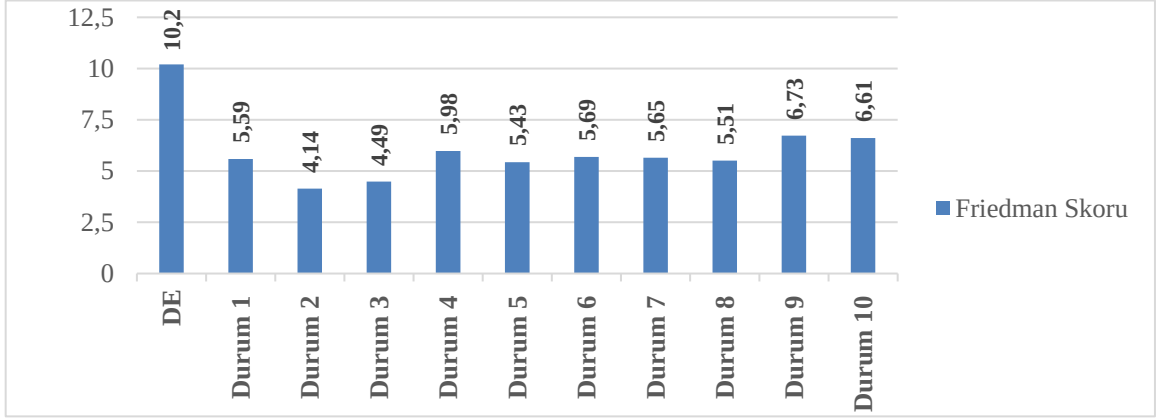
Şekil 63. CEC20 100D 5.problem çok halkalı durumlar için Friedman analiz sonuçları



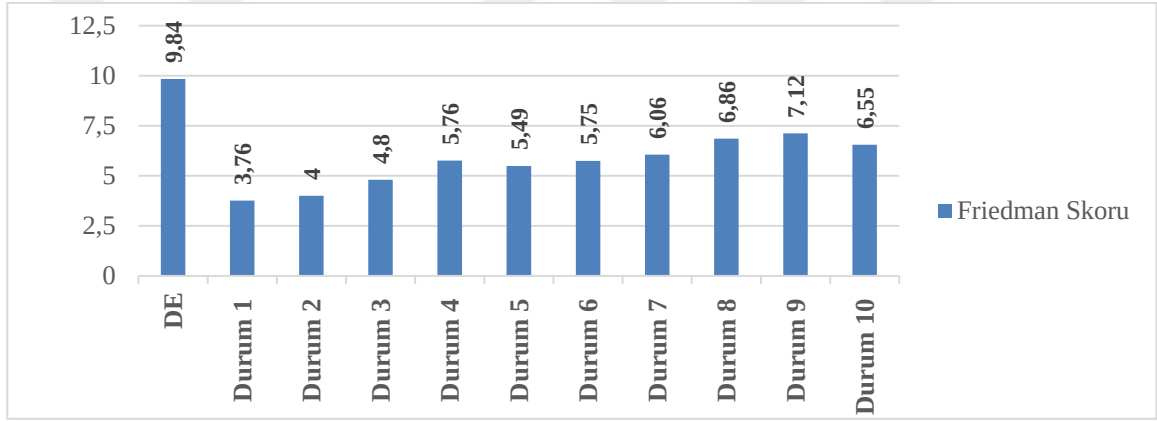
Şekil 64. CEC20 100D 6.problem çok halkalı durumlar için Friedman analiz sonuçları



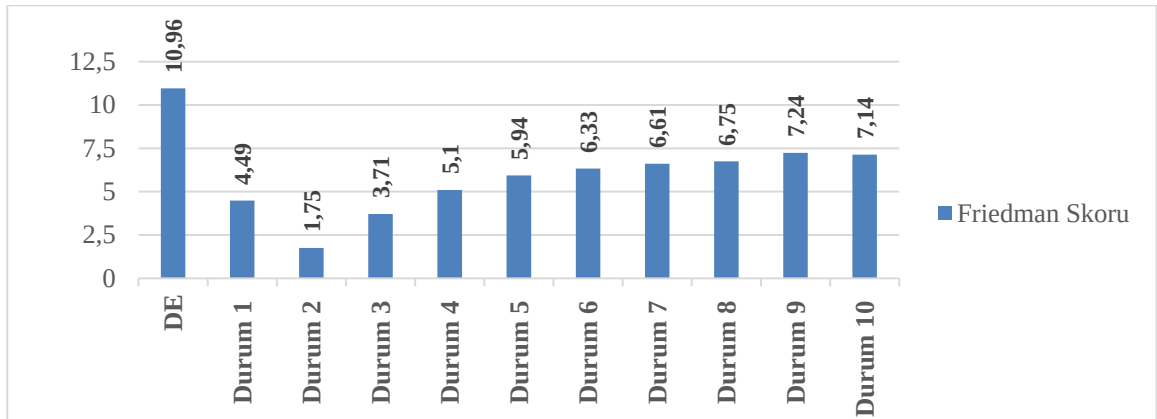
Şekil 65. CEC20 100D 7.problem çok halkalı durumlar için Friedman analiz sonuçları



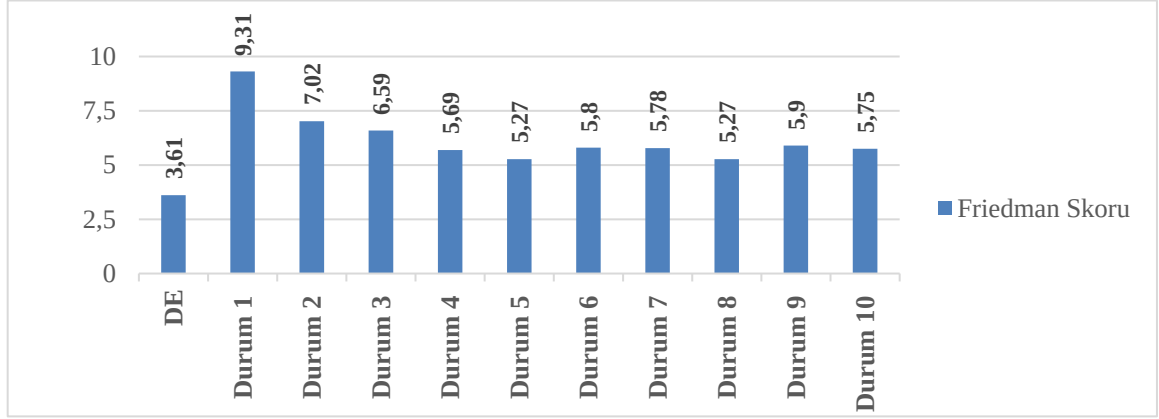
Şekil 66. CEC20 100D 8.problem çok halkalı durumlar için Friedman analiz sonuçları



Şekil 67. CEC20 100D 9.problem çok halkalı durumlar için Friedman analiz sonuçları



Şekil 68. CEC20 100D 10.problem çok halkalı durumlar için Friedman analiz sonuçları



Çok halkalı zincir için oluşturulan durumların sonuçları analiz edilmiştir. Oluşturulacak optimum zincir için problem ve boyut bazlı sonuçlardaki en iyi durumların zincirleri belirlenmiştir. Durumlar kısaltma için “D” olarak ifade edilmiştir. Bu durumda problem ve boyut bazlı olarak ayarlanacak optimum zincir Tablo 12’deki gibi olacaktır.

Tablo 12. Boyut ve problem bazlı olarak ayarlanan çok halkalı optimum zincir

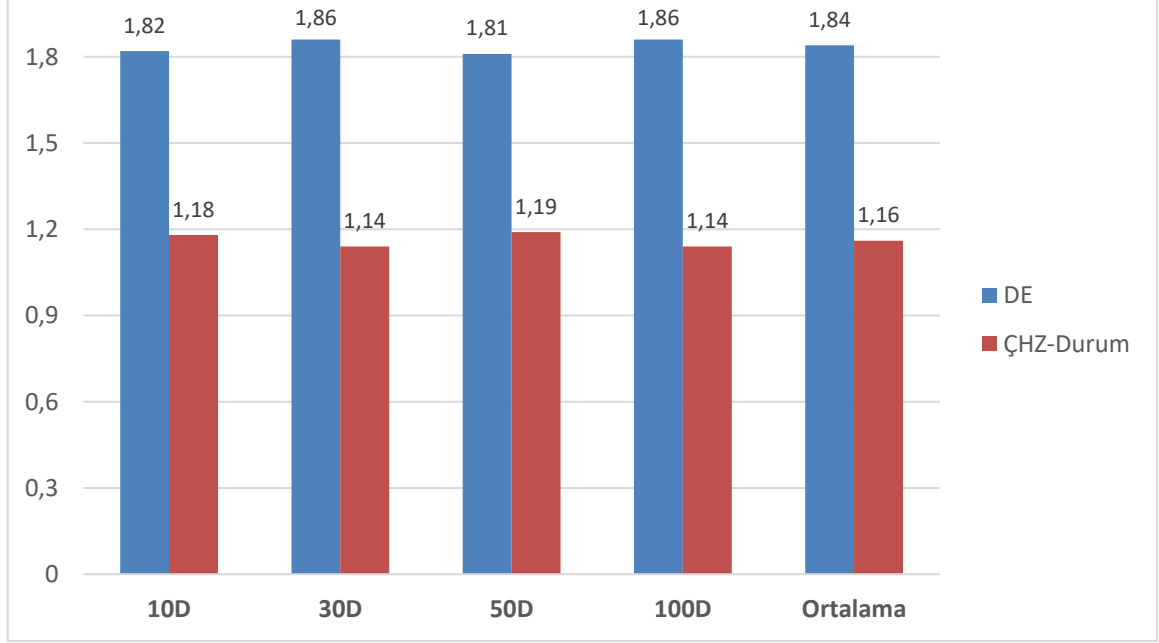
Problem	Tek Mod	Temel			Hibrit			Kompozisyon		
Boyutu	1	2	3	4	5	6	7	8	9	10
10D	D 2	D 3	D 3	D 3	D 4	D 6	D 3	D 8	D 10	D 7
30D	D 20	D 12	D 12	D 12	D 14	D 12	D 12	D 12	D 12	D1 8
50D	D 6	D 21	D 22	D 22	D 22	D 22	D 22	D 21	D 22	D 30
100D	D 6	D 1	D 2	D 3	D 2	D 2	D 2	D 1	D 2	D 5

Tablo 12’de verilen optimum zincirler ile oluşturulan durumun adı “Çok Halkalı Zincir Durum (ÇHZ-Durum)” olarak belirlenmiştir. Oluşturulan ÇHZ-Durum, CEC20 problem setinde problem ve boyut bazlı olarak algoritma içerisinde ilgili zincir seçilmektedir. Zincir seçim kuralı Tablo 12’de görülmektedir. Oluşturulan ÇHZ-Durum’un performansını görmek için DE algoritmasıyla Wilcoxon ve Friedman analizleri yapılmıştır. Yapılan analiz sonuçları Tablo 13 ve Şekil 69’de görülmektedir.

Tablo 13. DE algoritması ve ÇHZ-Durum Wilcoxon analiz sonucu

DE vs.+/-				
	D=10	D=30	D=50	D=100
ÇHZ-Durum	9/1/0	10/0/0	8/1/1	8/1/1

Şekil 69. ÇHZ-Durum için Friedman analiz sonucu



3.3.2. İki Halkalı Optimum Zincir Araştırılması

İki halkalı optimum zincir araştırmasında, DE algoritmasının baz arama denklemine en iyi uyum sağlayacak olan halkanın belirlenmesi amaçlanmıştır. Bundan dolayı oluşturulacak tüm zincirlerde 2 adet halka bulunacaktır. Bunlardan birisi DE algoritmasının baz arama denklemi olan 4 numaralı halka ile, diğer oluşturulan halkalardır. Oluşturulan durumlar Tablo 14’te görülmektedir. Oluşturulan durumlar belirli HED değerleri ve CEC20 problem boyutlarında çalıştırılacaktır.

Tablo 14. İki halkalı zincir için oluşturulan durumlar

Durum Adları	HED Değerleri	Problem Boyutları	Zincirdeki Halka Numaraları
Durum 31	0.01	10 ve 100	1-4
Durum 57	0.02	30	
Durum 83	5	50	
Durum 32	0.01	10 ve 100	2-4
Durum 58	0.02	30	
Durum 84	5	50	
Durum 33	0.01	10 ve 100	3-4
Durum 59	0.02	30	
Durum 85	5	50	
Durum 34	0.01	10 ve 100	5-4
Durum 60	0.02	30	
Durum 86	5	50	
Durum 35	0.01	10 ve 100	6-4
Durum 61	0.02	30	
Durum 87	5	50	
Durum 36	0.01	10 ve 100	7-4
Durum 62	0.02	30	
Durum 88	5	50	
Durum 37	0.01	10 ve 100	8-4
Durum 63	0.02	30	
Durum 89	5	50	
Durum 38	0.01	10 ve 100	9-4
Durum 64	0.02	30	
Durum 90	5	50	
Durum 39	0.01	10 ve 100	10-4
Durum 65	0.02	30	
Durum 91	5	50	
Durum 40	0.01	10 ve 100	11-4
Durum 66	0.02	30	
Durum 92	5	50	
Durum 78	0.02	30	
Durum 104	5	50	

Tablo 14'ün devamı

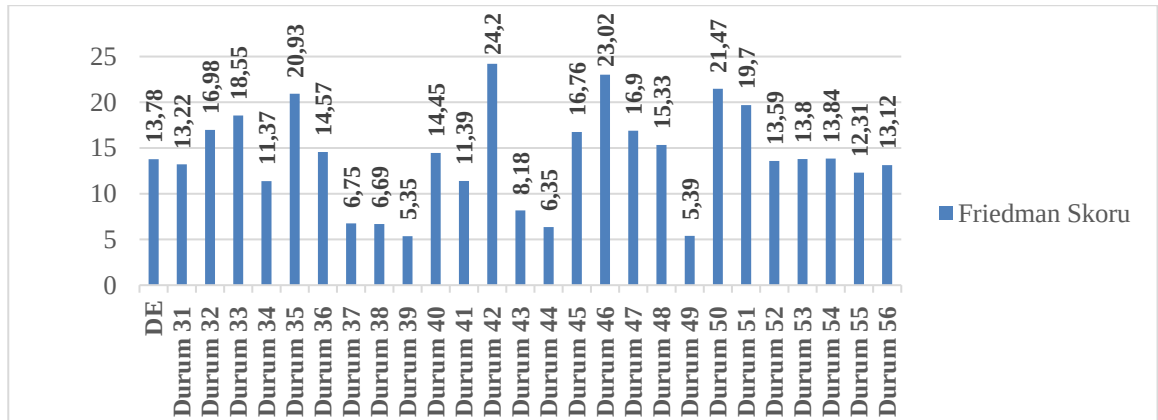
Durum 41	0.01	10 ve 100	12-4
Durum 67	0.02	30	
Durum 93	5	50	
Durum 42	0.01	10 ve 100	13-4
Durum 68	0.02	30	
Durum 94	5	50	
Durum 43	0.01	10 ve 100	14-4
Durum 69	0.02	30	
Durum 95	5	50	
Durum 44	0.01	10 ve 100	15-4
Durum 70	0.02	30	
Durum 96	5	50	
Durum 45	0.01	10 ve 100	16-4
Durum 71	0.02	30	
Durum 97	5	50	
Durum 46	0.01	10 ve 100	17-4
Durum 72	0.02	30	
Durum 98	5	50	
Durum 47	0.01	10 ve 100	18-4
Durum 73	0.02	30	
Durum 99	5	50	
Durum 48	0.01	10 ve 100	19-4
Durum 74	0.02	30	
Durum 100	5	50	
Durum 49	0.01	10 ve 100	20-4
Durum 75	0.02	30	
Durum 101	5	50	
Durum 50	0.01	10 ve 100	21-4
Durum 76	0.02	30	
Durum 102	5	50	
Durum 51	0.01	10 ve 100	22-4
Durum 77	0.02	30	
Durum 103	5	50	

Tablo 14'ün devamı

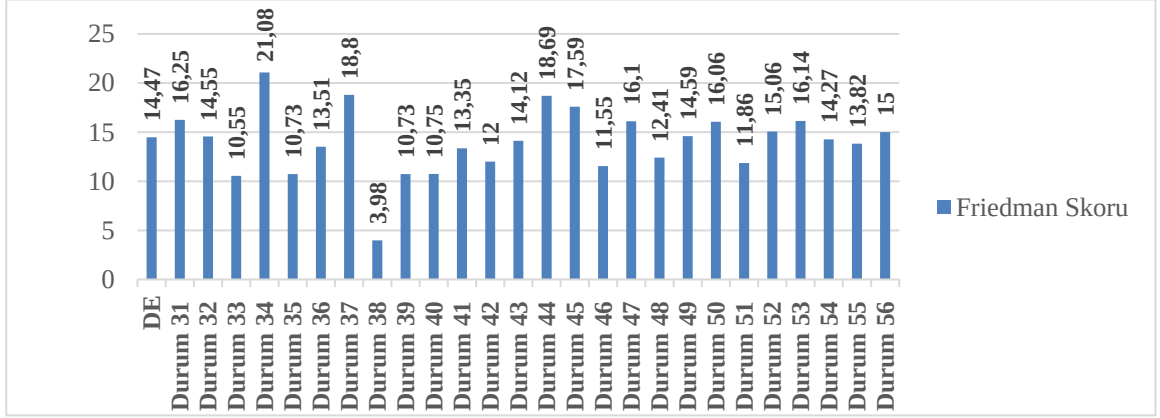
Durum 52	0.01	10 ve 100	23-4
Durum 78	0.02	30	
Durum 104	5	50	
Durum 53	0.01	10 ve 100	24-4
Durum 79	0.02	30	
Durum 105	5	50	
Durum 54	0.01	10 ve 100	25-4
Durum 80	0.02	30	
Durum 106	5	50	
Durum 55	0.01	10 ve 100	26-4
Durum 81	0.02	30	
Durum 107	5	50	
Durum 56	0.01	10 ve 100	27-4
Durum 82	0.02	30	
Durum 108	5	50	

İki halkalı oluşturulan durumlar, çok halkalı durum çalışmasında yapılan çalışmada olduğu gibi CEC20 problem setinde boyut ve problem bazlı olarak çalıştırılarak analiz edilmiştir. Yapılan çalışmaların analiz sonuçları Şekil 70 ve Şekil 109 arasındaki şekillerde verilmiştir.

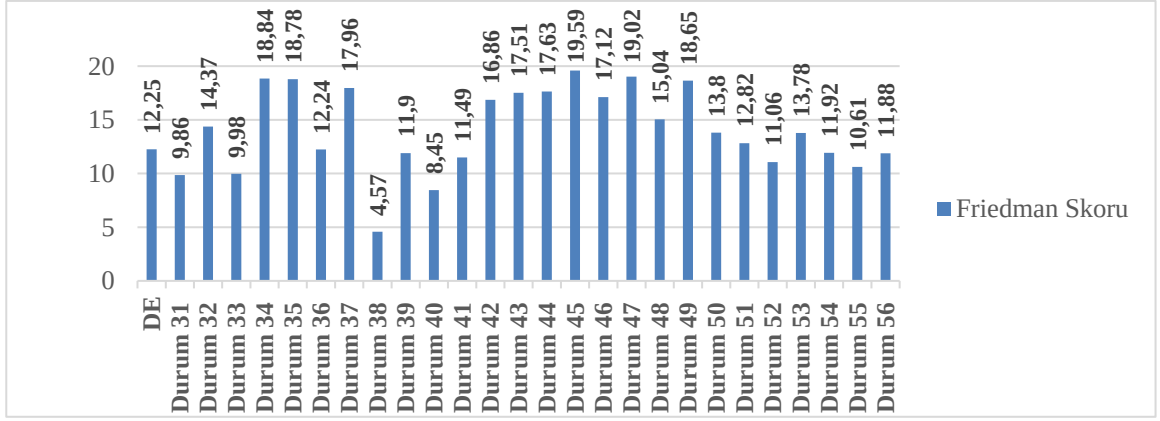
Şekil 70. CEC20 10D 1.problem iki halkalı durumlar için Friedman analiz sonuçları



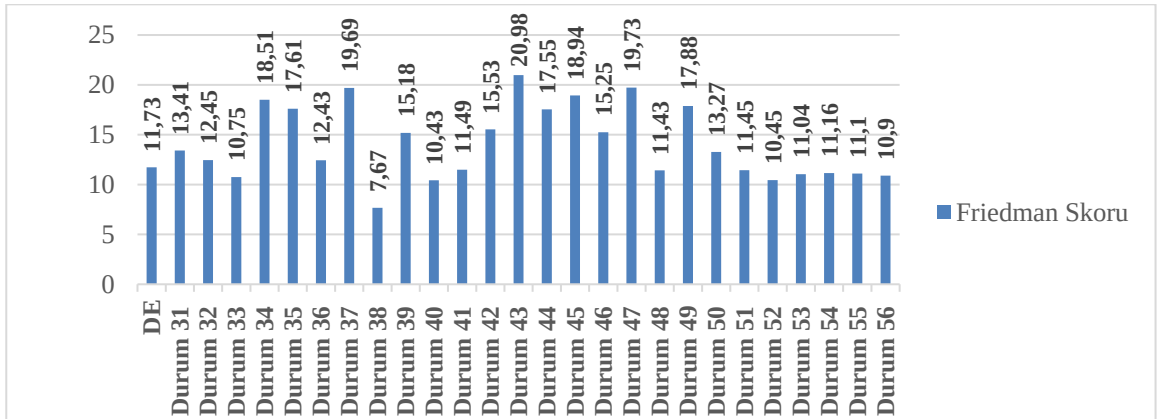
Şekil 71. CEC20 10D 2.problem iki halkalı durumlar için Friedman analiz sonuçları



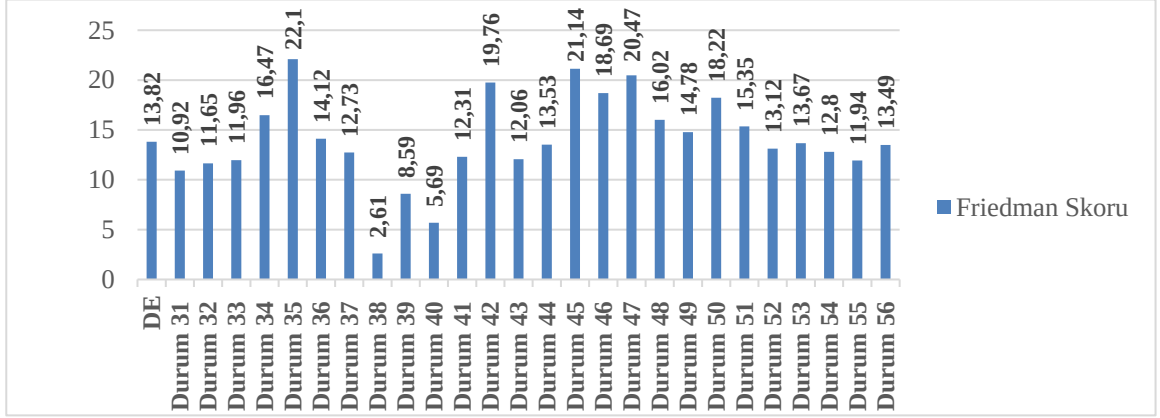
Şekil 72. CEC20 10D 3.problem iki halkalı durumlar için Friedman analiz sonuçları



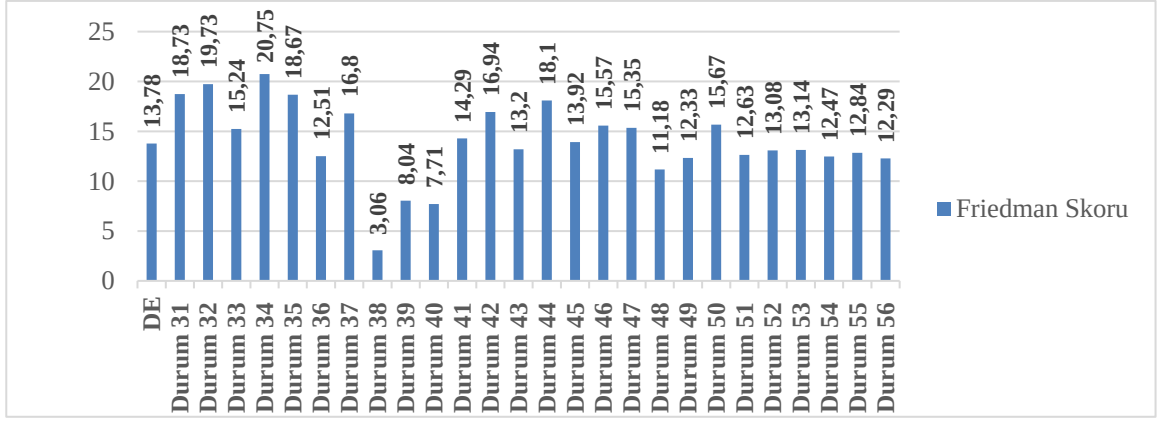
Şekil 73. CEC20 10D 4.problem iki halkalı durumlar için Friedman analiz sonuçları



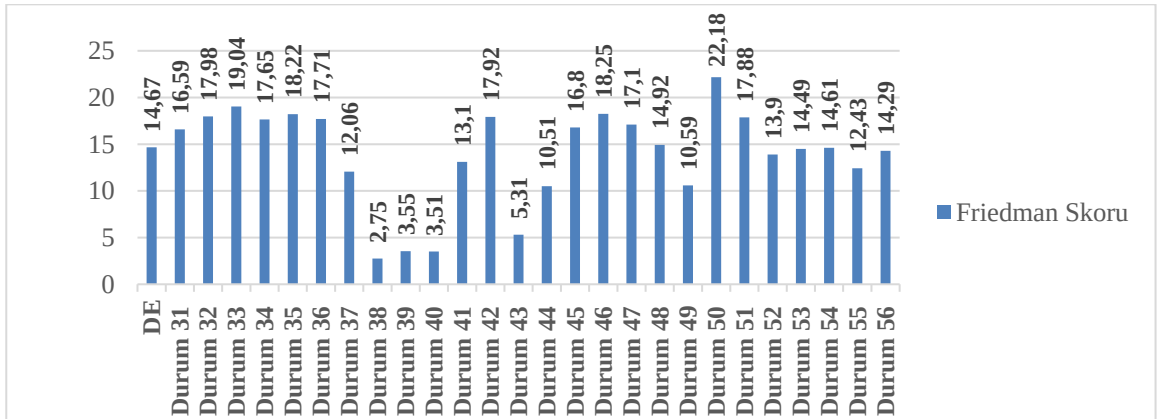
Şekil 74. CEC20 10D 5.problem iki halkalı durumlar için Friedman analiz sonuçları



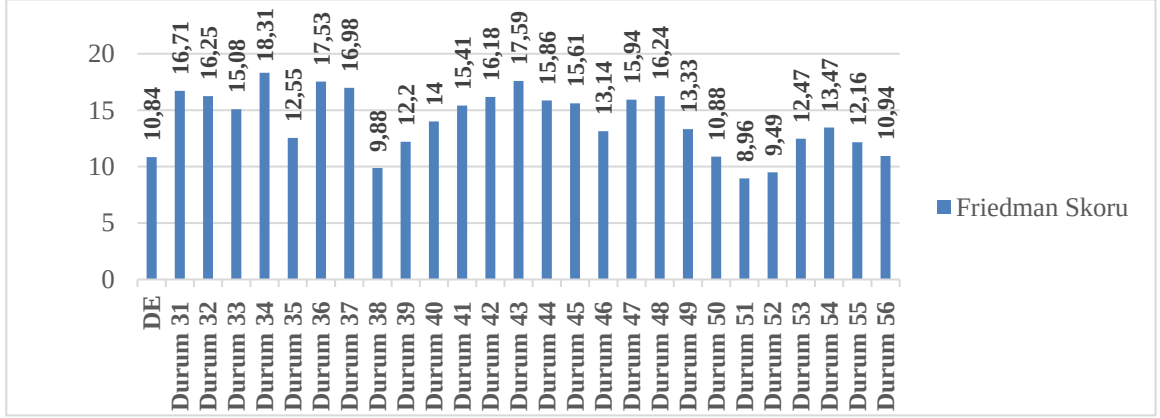
Şekil 75. CEC20 10D 6.problem iki halkalı durumlar için Friedman analiz sonuçları



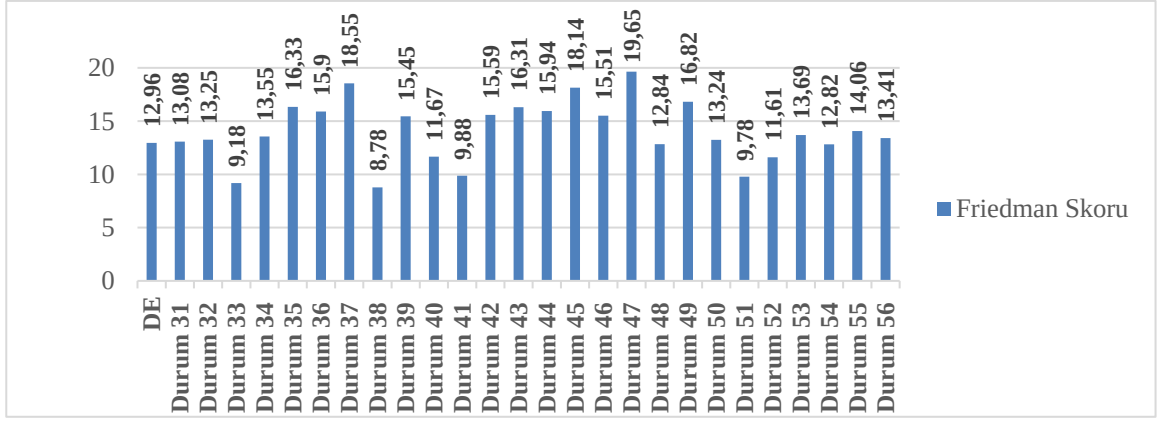
Şekil 76. CEC20 10D 7.problem iki halkalı durumlar için Friedman analiz sonuçları



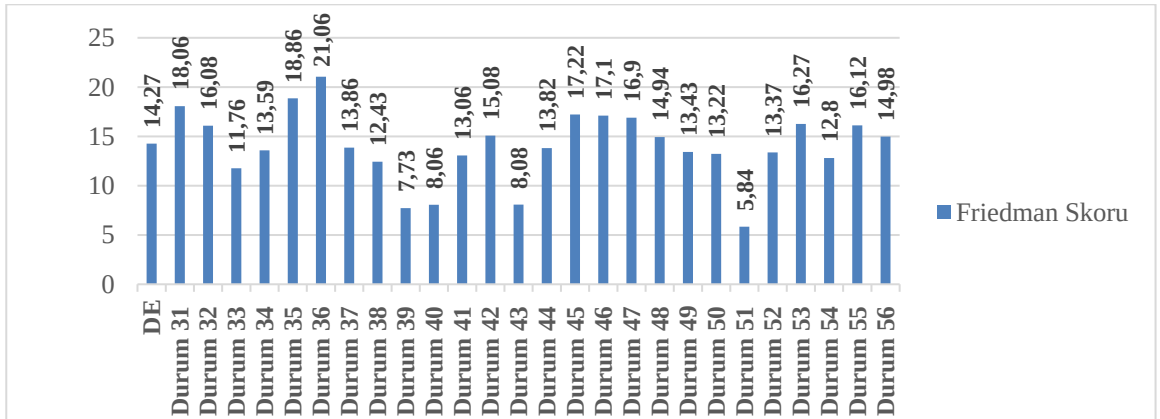
Şekil 77. CEC20 10D 8.problem iki halkalı durumlar için Friedman analiz sonuçları



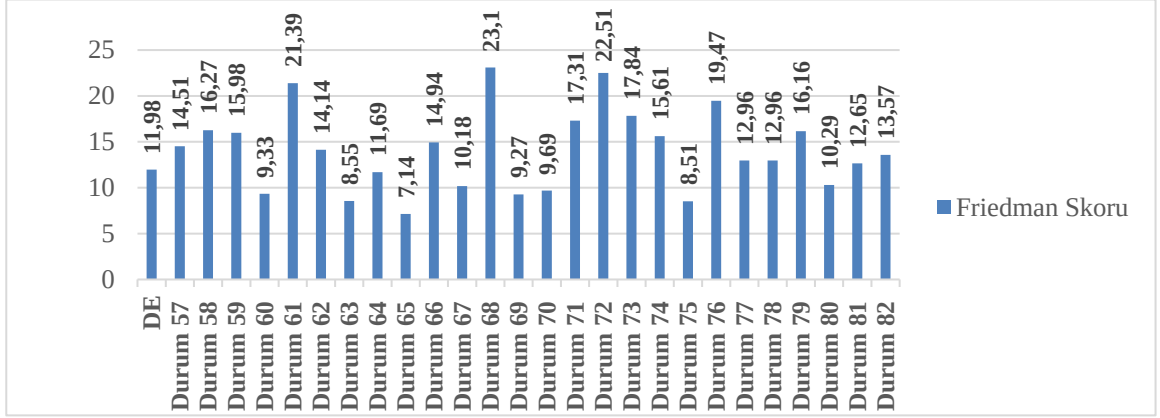
Şekil 78. CEC20 10D 9.problem iki halkalı durumlar için Friedman analiz sonuçları



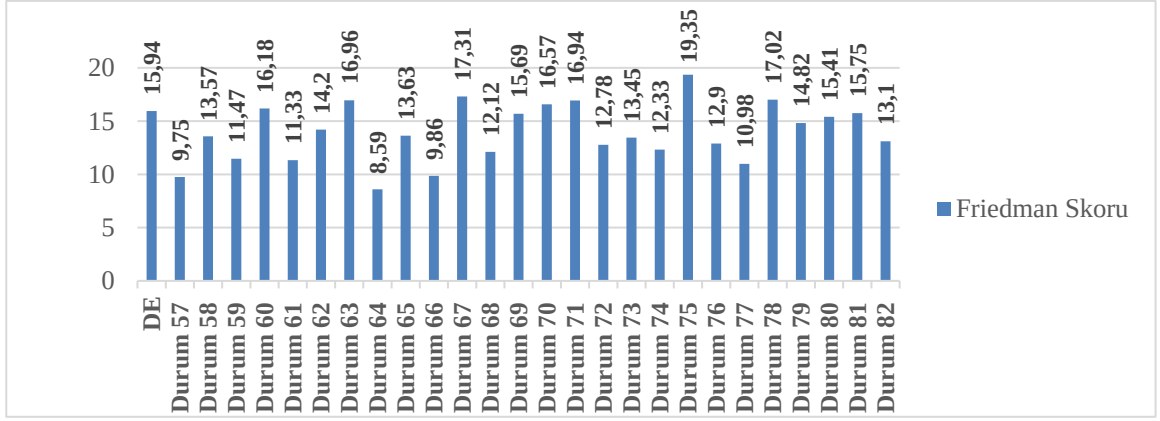
Şekil 79. CEC20 10D 10.problem iki halkalı durumlar için Friedman analiz sonuçları



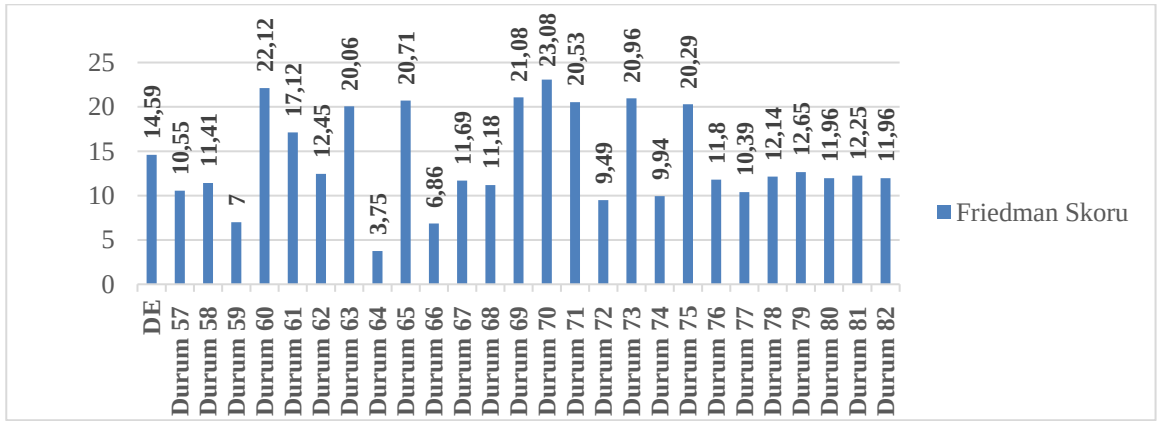
Şekil 80. CEC20 30D 1.problem iki halkalı durumlar için Friedman analiz sonuçları



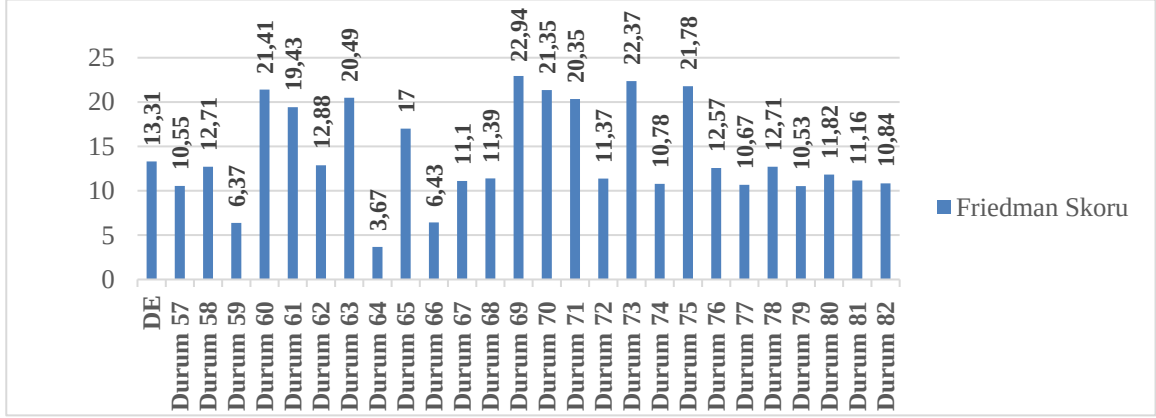
Şekil 81. CEC20 30D 2.problem iki halkalı durumlar için Friedman analiz sonuçları



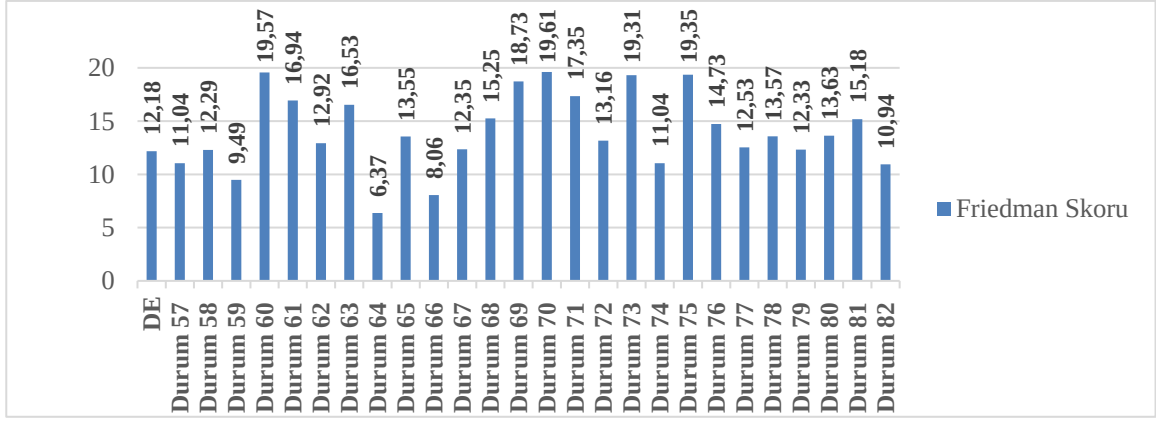
Şekil 82. CEC20 30D 3.problem iki halkalı durumlar için Friedman analiz sonuçları



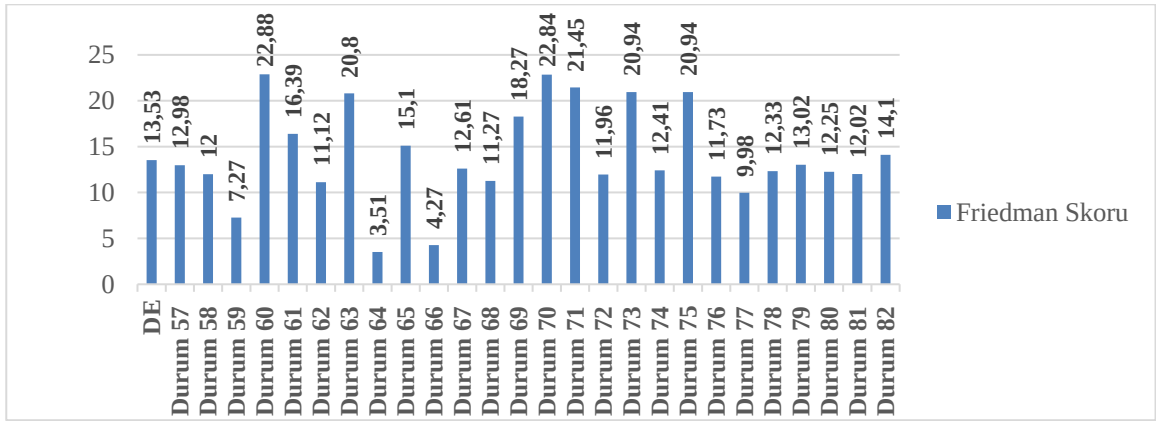
Şekil 83. CEC20 30D 4.problem iki halkalı durumlar için Friedman analiz sonuçları



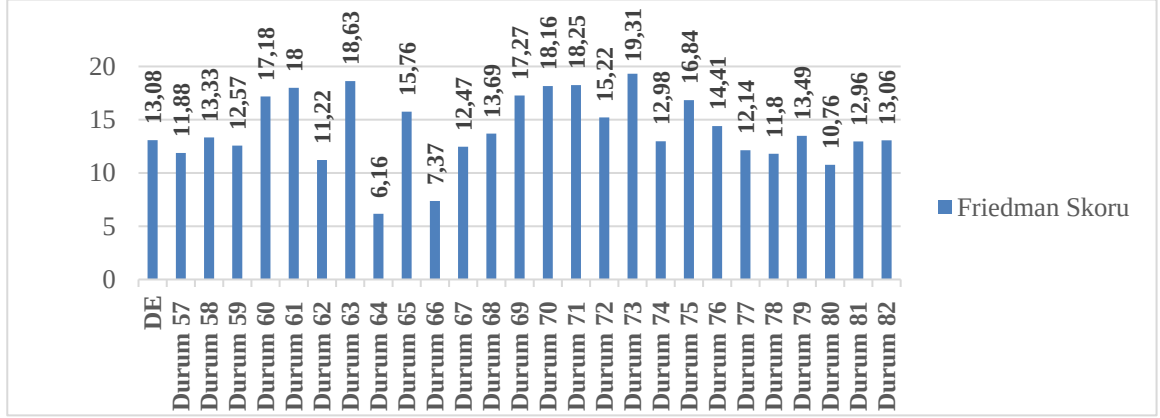
Şekil 84. CEC20 30D 5.problem iki halkalı durumlar için Friedman analiz sonuçları



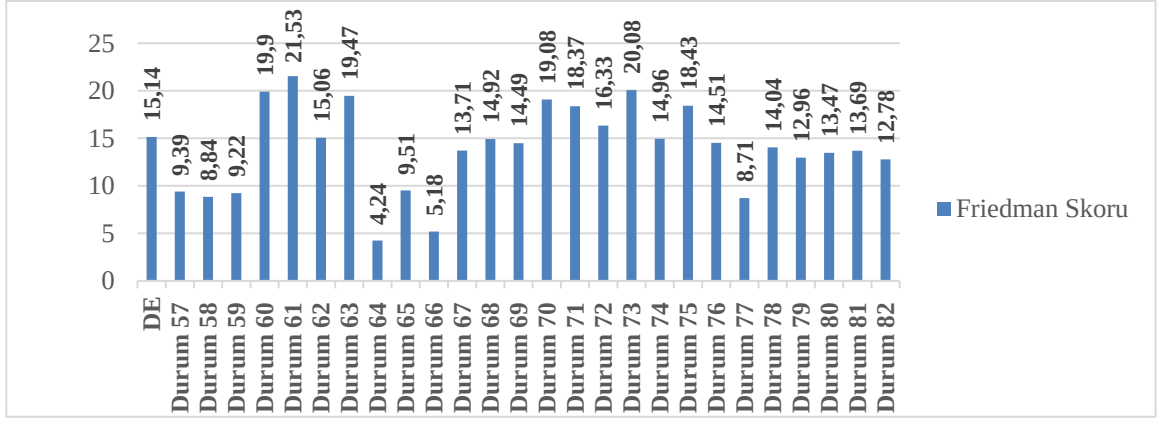
Şekil 85. CEC20 30D 6.problem iki halkalı durumlar için Friedman analiz sonuçları



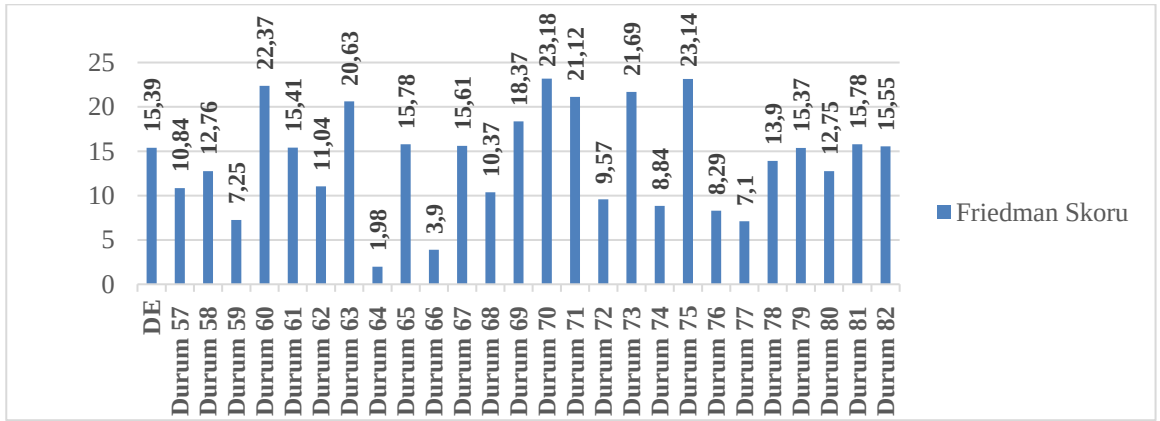
Şekil 86. CEC20 30D 7.problem iki halkalı durumlar için Friedman analiz sonuçları



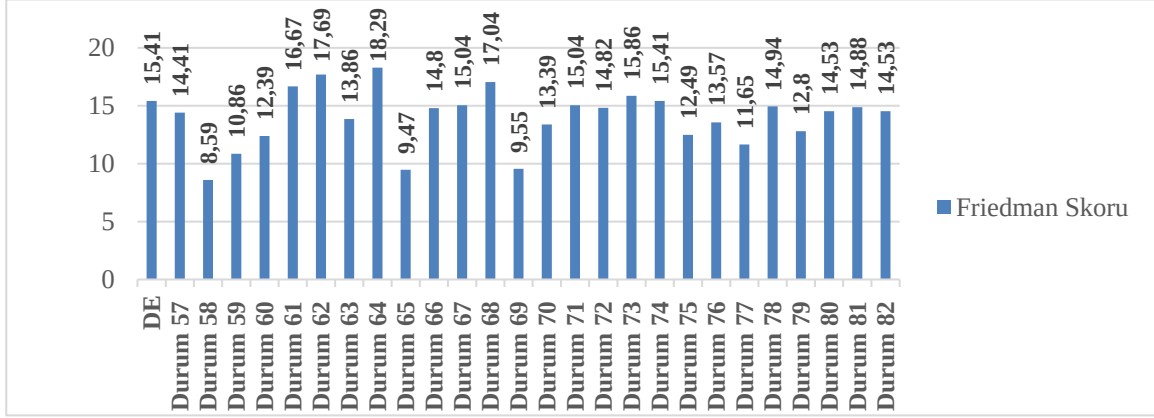
Şekil 87. CEC20 30D 8.problem iki halkalı durumlar için Friedman analiz sonuçları



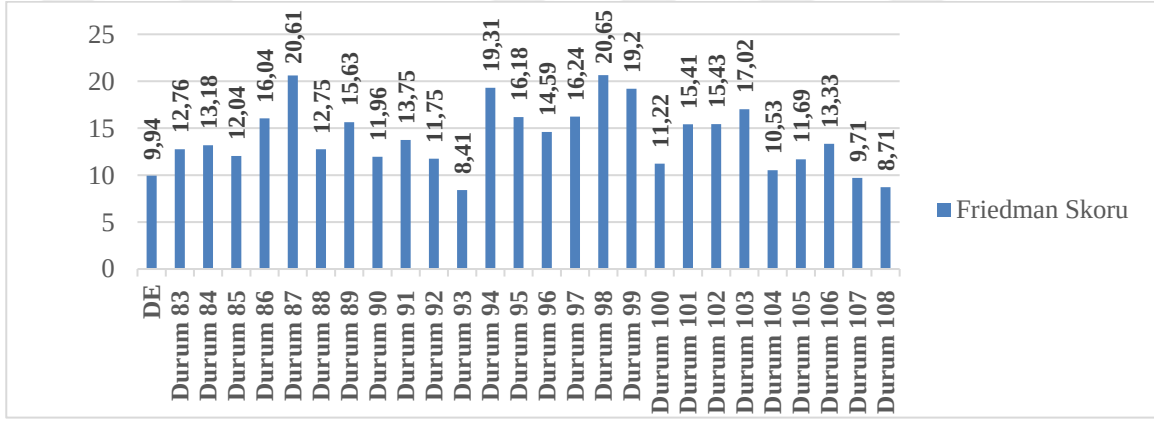
Şekil 88. CEC20 30D 9.problem iki halkalı durumlar için Friedman analiz sonuçları



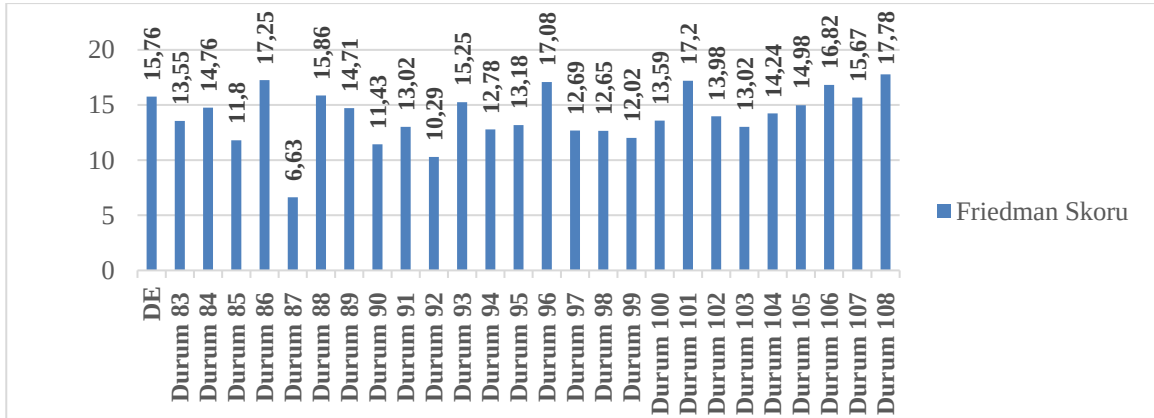
Şekil 89. CEC20 30D 10.problem iki halkalı durumlar için Friedman analiz sonuçları



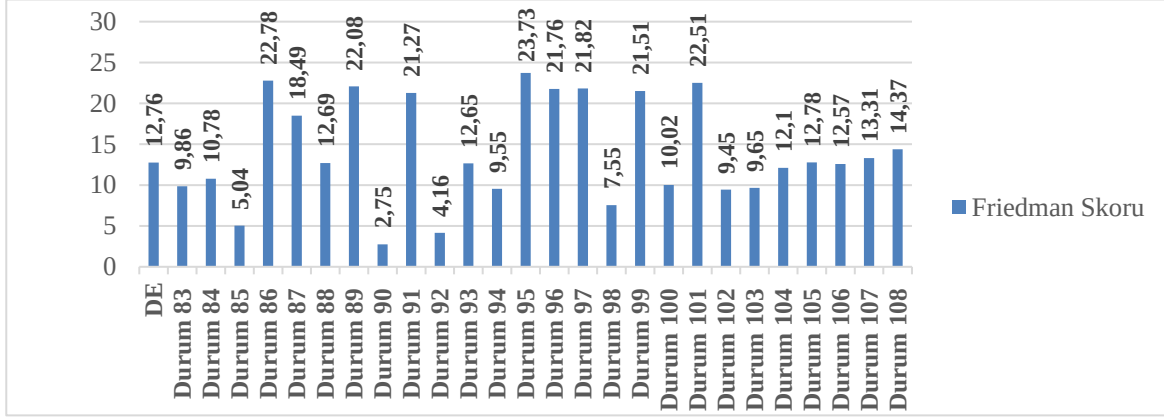
Şekil 90. CEC20 50D 1.problem iki halkalı durumlar için Friedman analiz sonuçları



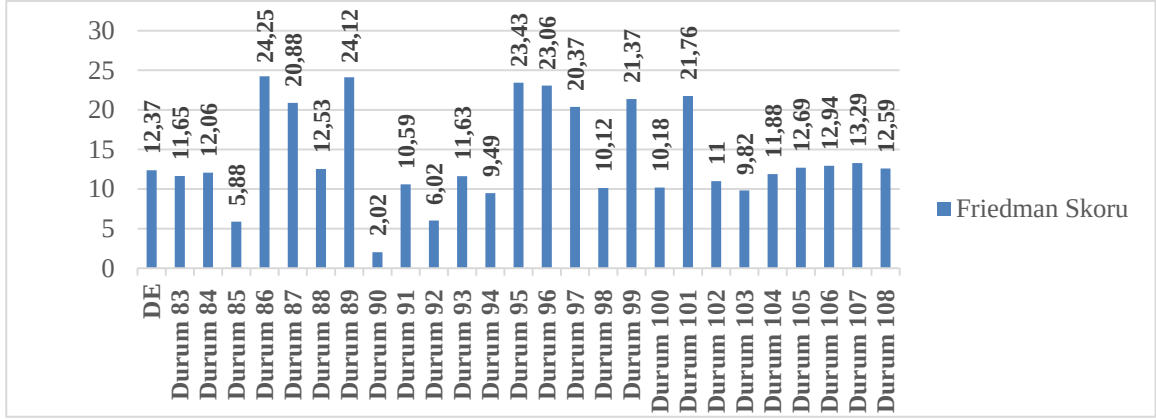
Şekil 91. CEC20 50D 2.problem iki halkalı durumlar için Friedman analiz sonuçları



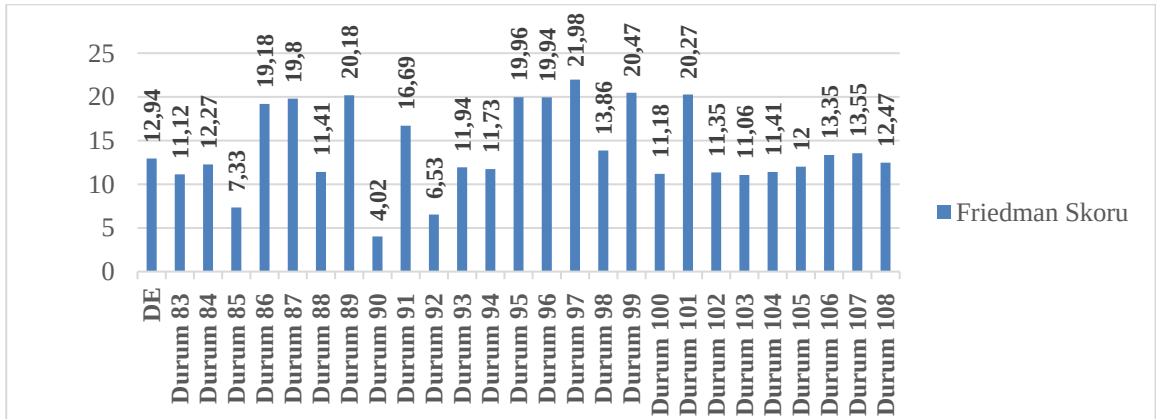
Şekil 92. CEC20 50D 3.problem iki halkalı durumlar için Friedman analiz sonuçları



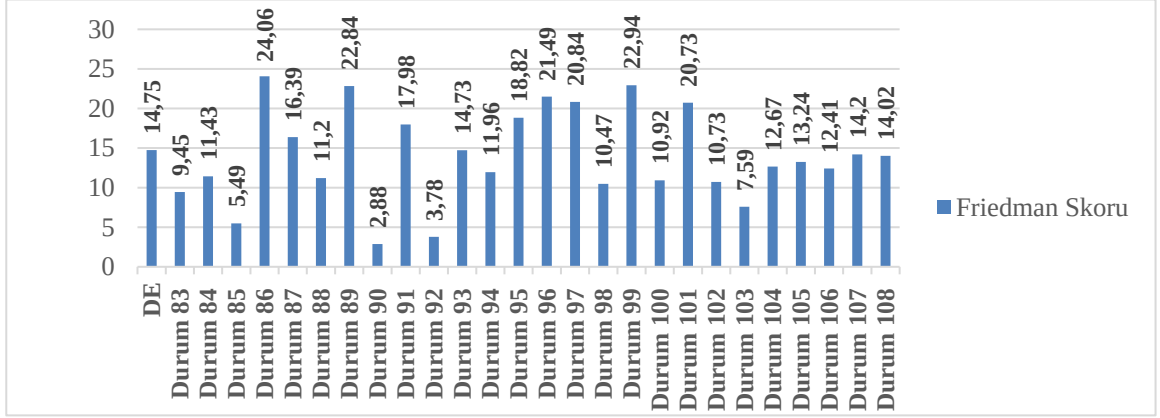
Şekil 93. CEC20 50D 4.problem iki halkalı durumlar için Friedman analiz sonuçları



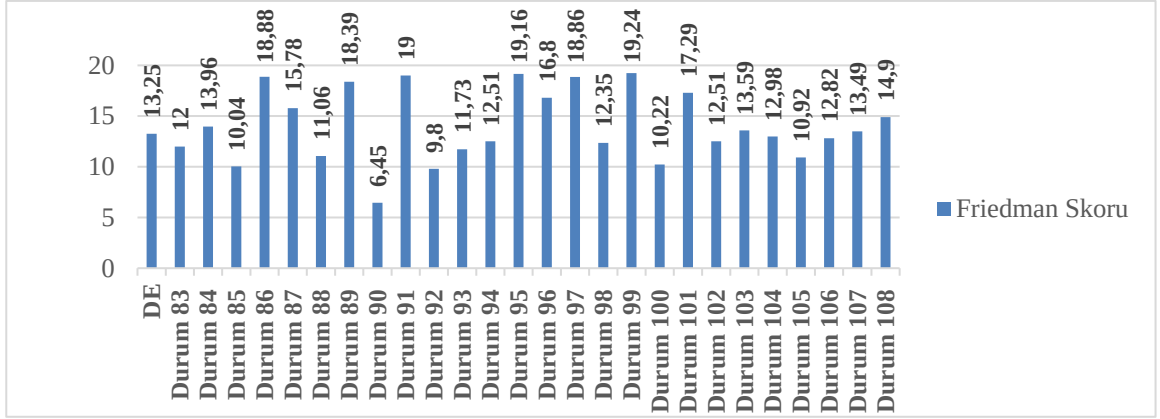
Şekil 94. CEC20 50D 5.problem iki halkalı durumlar için Friedman analiz sonuçları



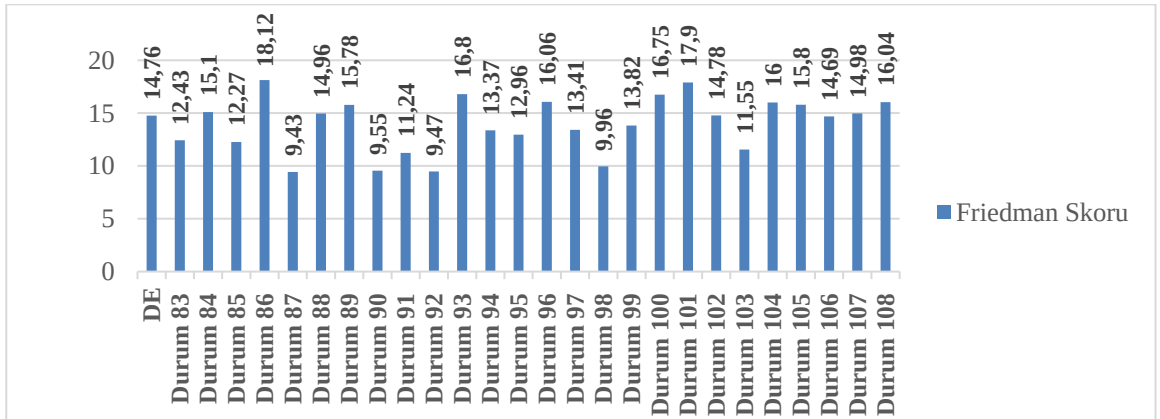
Şekil 95. CEC20 50D 6.problem iki halkalı durumlar için Friedman analiz sonuçları



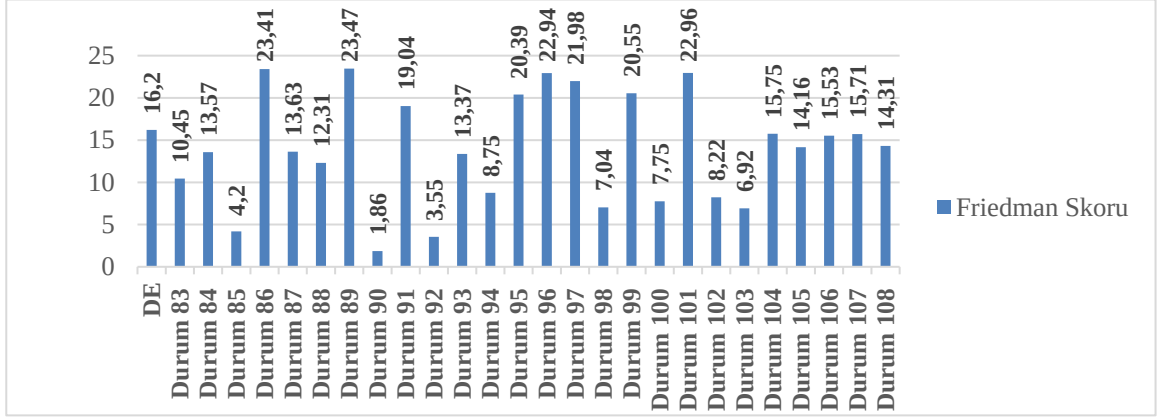
Şekil 96. CEC20 50D 7.problem iki halkalı durumlar için Friedman analiz sonuçları



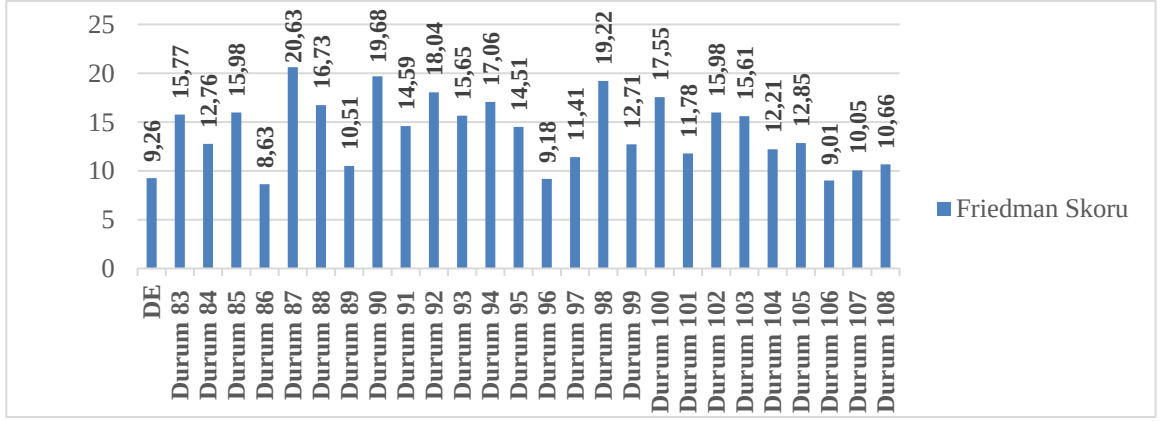
Şekil 97. CEC20 50D 8.problem iki halkalı durumlar için Friedman analiz sonuçları



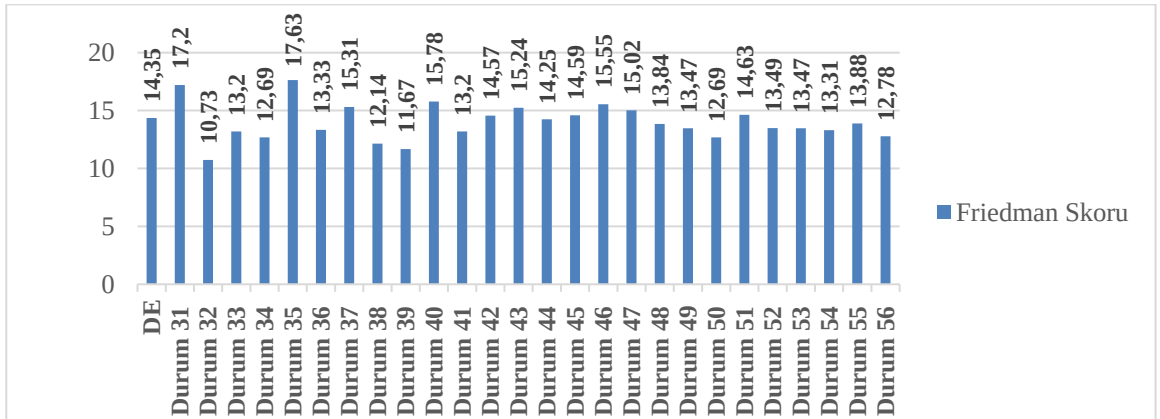
Şekil 98. CEC20 50D 9.problem iki halkalı durumlar için Friedman analiz sonuçları



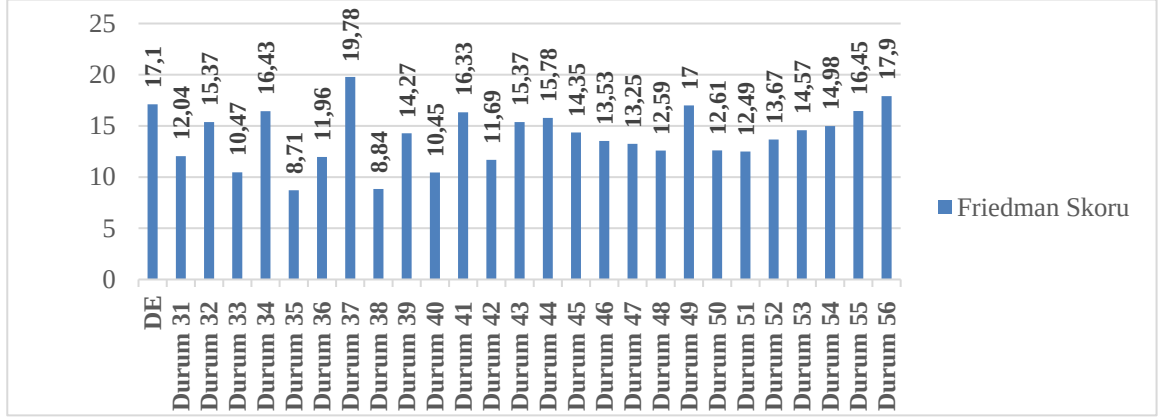
Şekil 99. CEC20 50D 10.problem iki halkalı durumlar için Friedman analiz sonuçları



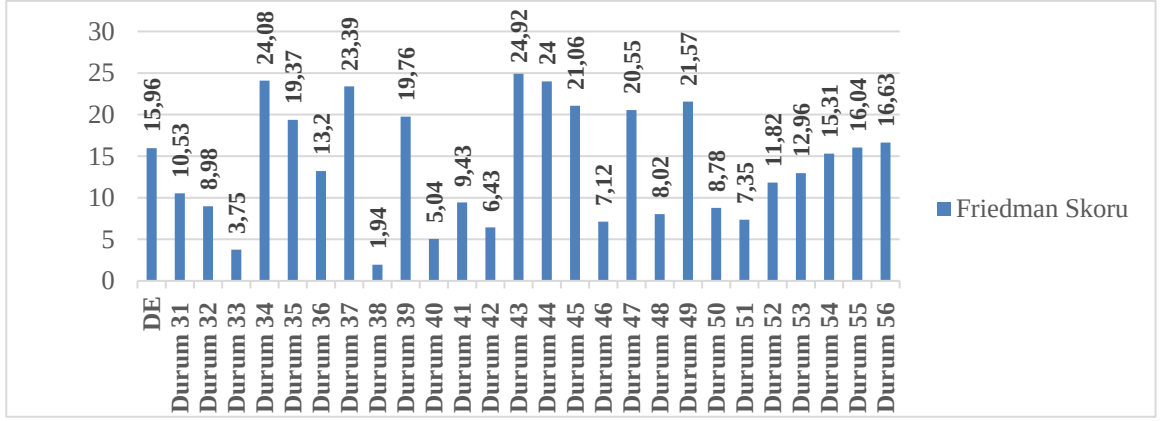
Şekil 100. CEC20 100D 1.problem iki halkalı durumlar için Friedman analiz sonuçları



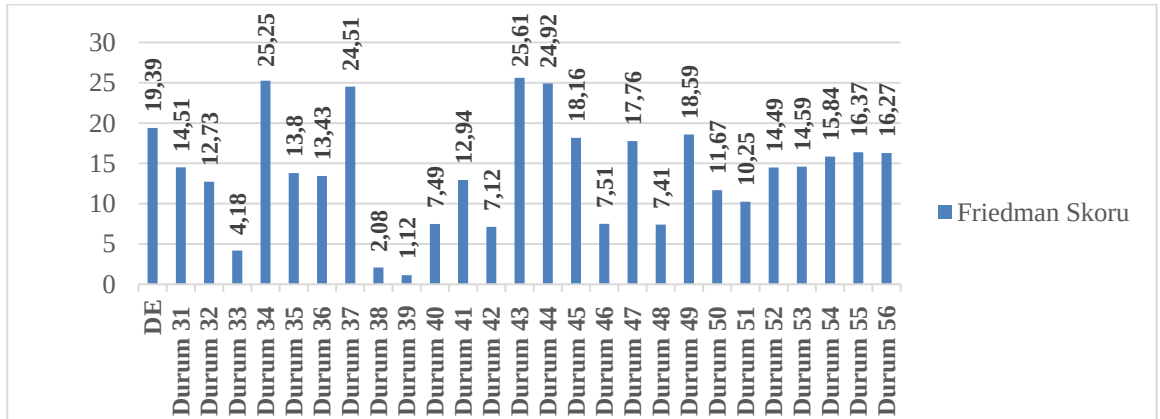
Şekil 101. CEC20 100D 2.problem iki halkalı durumlar için Friedman analiz sonuçları



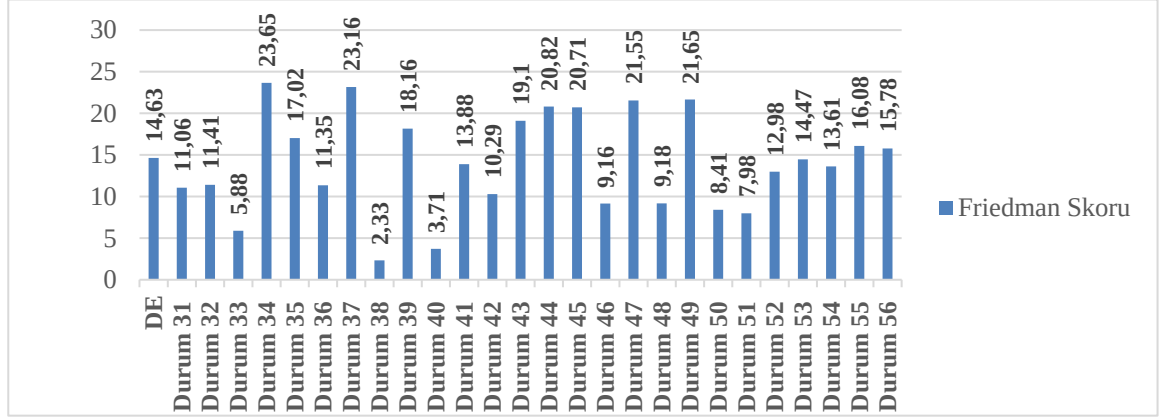
Şekil 102. CEC20 100D 3.problem iki halkalı durumlar için Friedman analiz sonuçları



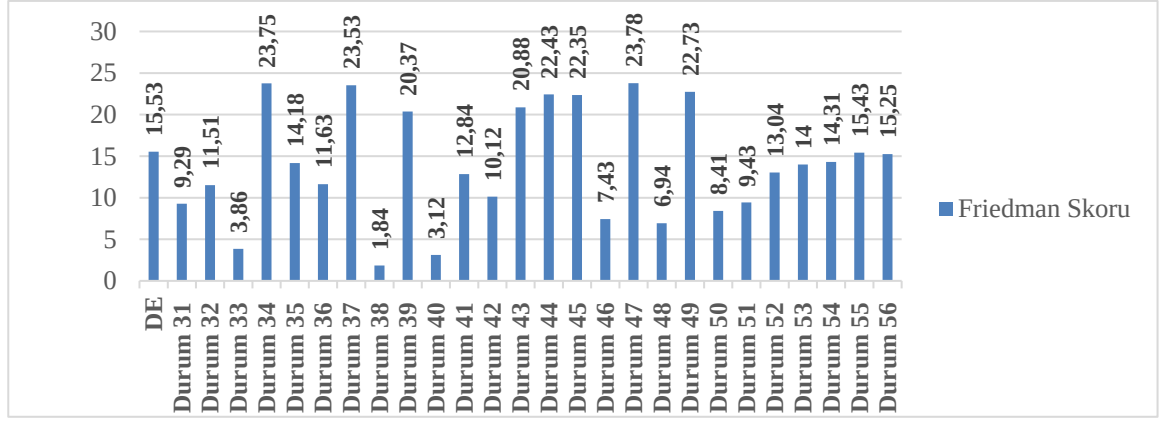
Şekil 103. CEC20 100D 4.problem iki halkalı durumlar için Friedman analiz sonuçları



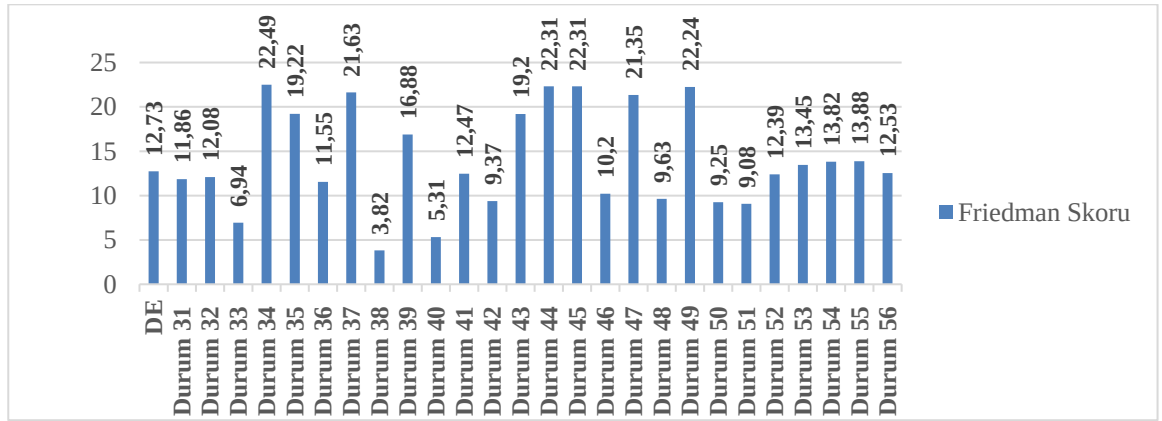
Şekil 104. CEC20 100D 5.problem iki halkalı durumlar için Friedman analiz sonuçları



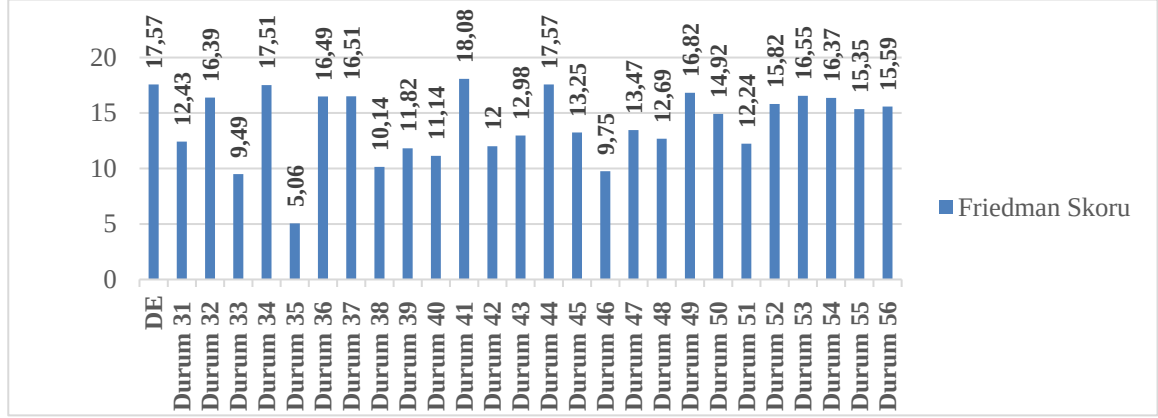
Şekil 105. CEC20 100D 6.problem iki halkalı durumlar için Friedman analiz sonuçları



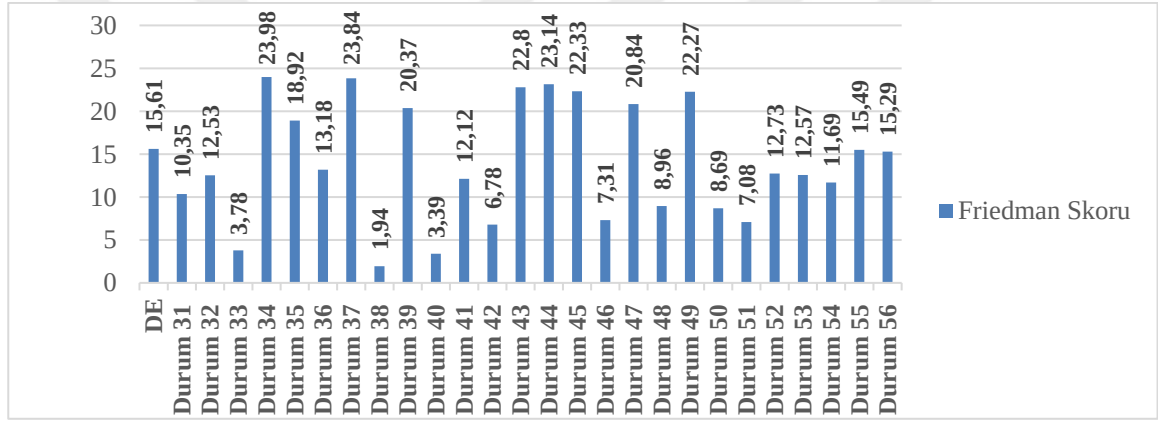
Şekil 106. CEC20 100D 7.problem iki halkalı durumlar için Friedman analiz sonuçları



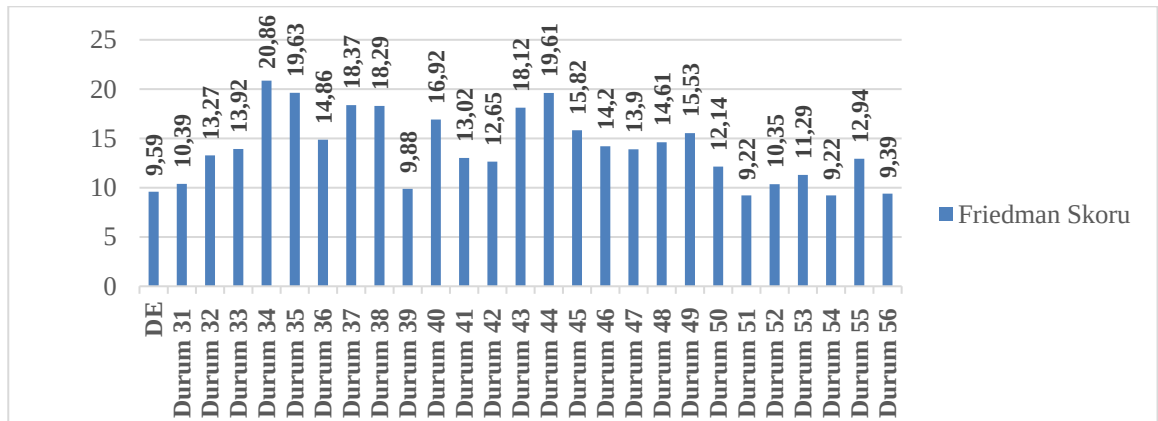
Şekil 107. CEC20 100D 8.problem iki halkalı durumlar için Friedman analiz sonuçları



Şekil 108. CEC20 100D 9.problem iki halkalı durumlar için Friedman analiz sonuçları



Şekil 109. CEC20 100D 10.problem iki halkalı durumlar için Friedman analiz sonuçları



İki halkalı zincir için oluşturulan durumların sonuçları analiz edilmiştir. Oluşturulacak optimum zincir için problem ve boyut bazlı sonuçlardaki en iyi durumların zincirleri belirlenmiştir. Durumlar kısaltma için “D” olarak ifade edilmiştir. Bu durumda problem ve boyut bazlı olarak ayarlanacak optimum zincir Tablo 15’deki gibi olacaktır.

Tablo 15. Boyut ve problem bazlı olarak ayarlanan iki halkalı optimum zincir

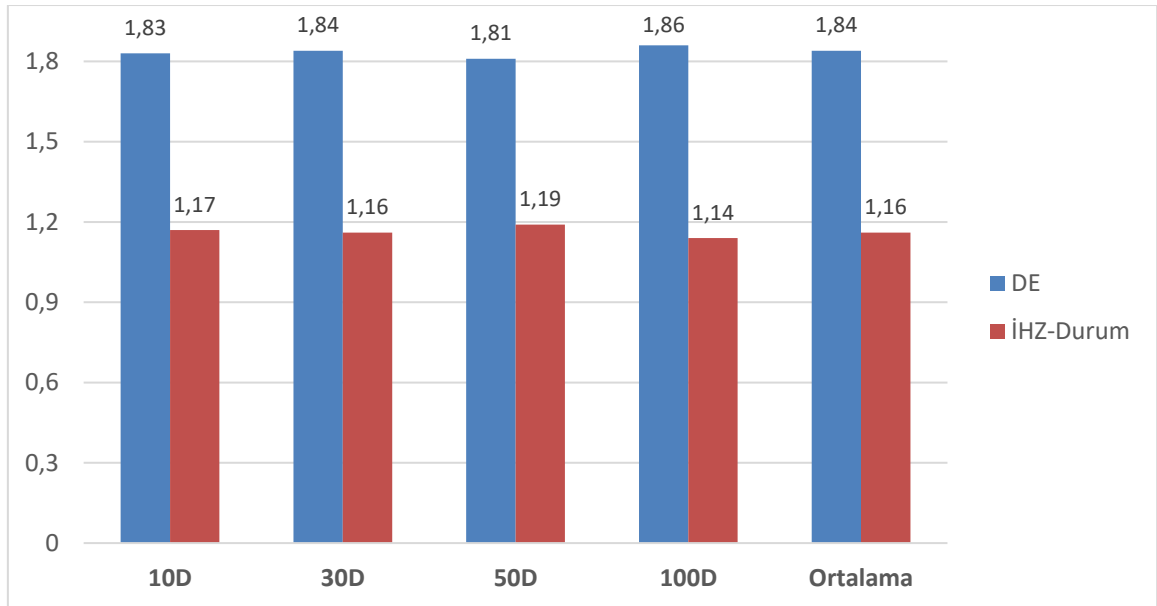
Problem Boyutu	Tek Mod	Temel			Hibrit			Kompozisyon		
	1	2	3	4	5	6	7	8	9	10
10D	D 39	D 38	D 38	D 38	D 38	D 38	D 38	D 51	D 38	D 51
30D	D 65	D 64	D 64	D 64	D 64	D 64	D 64	D 64	D 64	D 58
50D	D 93	D 87	D 90	D 90	D 90	D 90	D 90	D 87	D 90	D 86
100D	D 32	D 35	D 38	D 39	D 38	D 38	D 38	D 35	D 38	D 51

Tablo 15’te verilen optimum zincirler ile oluşturulan durumun adı “İki Halkalı Zincir Durum (İHZ-Durum)” olarak belirlenmiştir. Oluşturulan İHZ-Durum, CEC20 problem setinde problem ve boyut bazlı olarak algoritma içerisinde ilgili zincir seçilmektedir. Zincir seçim kuralı Tablo 15’te görülmektedir. Oluşturulan İHZ-Durum’un performansını görmek için DE algoritmasıyla Wilcoxon ve Friedman analizleri yapılmıştır. Yapılan analiz sonuçları Tablo 16 ve Şekil 110’da görülmektedir.

Tablo 16. DE algoritması ve İHZ-Durum Wilcoxon analiz sonucu

DE vs. +/-				
	D=10	D=30	D=50	D=100
İHZ-Durum	10/0/0	10/0/0	8/2/0	9/1/0

Şekil 110. İHZ-Durum için Friedman analiz sonucu



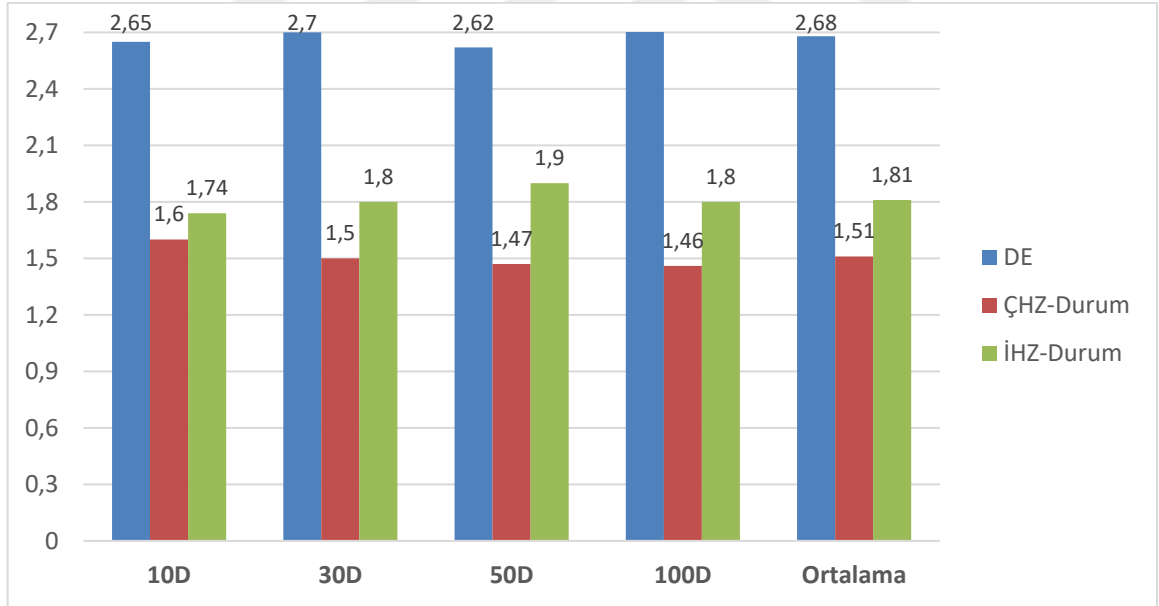
3.3.3. İkili ve Çok Halkalı En İyi Durumların Karşılaştırılması

Bu bölümde ikili ve çok halkalı çalışmalarında belirlenen Cec20 problem seti için problem ve boyut bazlı optimum zincirlerin performans karşılaştırması yapılmıştır. Yapılan karşılaştırma sonucunda hangi çalışmanın DE algoritması üzerinde daha olumlu etki yarattığı belirlenmiş olacaktır. Yapılan performans karşılaştırma analiz sonuçları Tablo 17 ve Şekil 111’de verilmiştir. Yapılan analiz sonuçları incelendiğinde en iyi iyileşme ÇHZ-Durum çalışmasında elde edilmiştir.

Tablo 17. ÇHZ-Durum ve İHZ-Durum çalışmalarının Wilcoxon analiz sonuçları

DE vs. +/-				
	D=10	D=30	D=50	D=100
ÇHZ-Durum	9/1/0	10/0/0	8/1/1	8/1/1
İHZ-Durum	10/0/0	10/0/0	8/2/0	9/1/0

Şekil 111. ÇHZ-Durum ve İHZ-Durum çalışmalarının Friedman analiz sonuçları



3.3.4. İkili ve Çok Halkalı En İyi Durumların Rakip Algoritmalarla Karşılaştırılması

Bu bölümde, bir önceki bölümlerde oluşturulmuş ÇHZ-Durum ve İHZ-Durum çalışmalarının performanslarını ölçebilmek için literatürden seçilen 51 algoritma ile karşılaştırma yapılmıştır. Kullanılan algoritmalar:

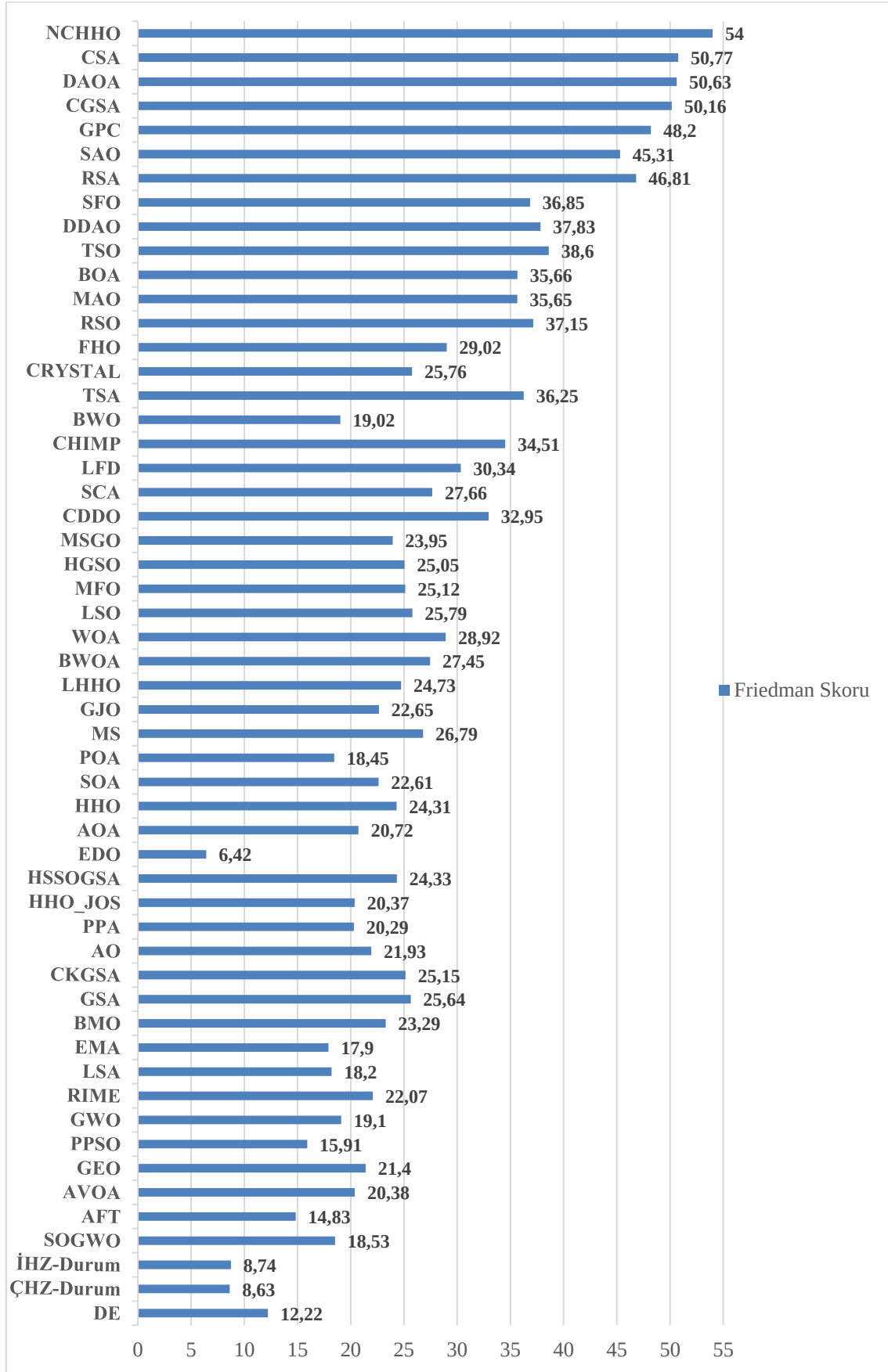
- Seçmeli Karşıtlık Temelli Gri Kurt Optimizasyon (SOGWO) (Dhargupta, Ghosh, Mirjalili, & Sarkar, 2020)
- Ali Baba ve Kırk Haramiler (Ali Baba and the Forty Thieves) (AFT) (Braik, Ryalat, & Al-Zoubi, 2022)
- Afrika Akbabaları Optimizasyon Algoritması (AVOA) (Abdollahzadeh, Gharehchopogh, & Mirjalili, 2021)
- Altın Kartal Optimizasyonu (GEO) (Mohammadi-Balani, Nayeri, Azar, & Taghizadeh-Yazdi, 2021)
- Fazör Parçacık Sürü Optimizasyonu (PPSO) (Ghasemi, ve diğerleri, 2019)
- Gri Kurt Optimizasyon (GWO)
- Kırağı-buz Fenomeni Optimizasyonu (RIME) (Su, ve diğerleri, 2023)
- Yıldırım Arama Algoritması (LSA) (Shareef, Ibrahim, & Mutlag, 2015)
- Döviz Piyasası Algoritması (EMA) (Ghorbani & Babaei, 2014)
- Denizkestaneleri Çiftleşme Optimizasyonu (BMO) (Sulaiman, Mustaffa, Saari, & Daniyal, 2020)
- Yerçekimi Arama Algoritması (GSA)
- Kaotik Kbest Yerçekimi Arama Algoritması (CKGSA) (Mittal, Pal, Kulhari, & Saraswat, 2016)
- Şahin Optimizasyonu (AO) (Abualigah, ve diğerleri, 2021)
- Parazitizm-predasyon Algoritması (PPA) (Mohamed, ve diğerleri, 2020)
- Diferansiyel Evrim (DE)
- Harris Şahinleri Ortak Karşı Seçim Optimizasyonu (HHO_JOS) (Arini, Chiewchanwattana, Soomlek, & Sunat, 2022)
- Hibrid Sperm Sürü ve Yerçekimi Arama Optimizasyonu (HSSOGSA) (Shehadeh, 2021)
- Üstel Dağıtım Optimizasyonu (EDO) (Abdel-Basset, El-Shahat, Jameel, & Abouhawwash, 2023)
- Arşimet Optimizasyon Algoritması (AOA) (Hashim, Hussain, Houssein, Mabrouk, & Al-Atabany, 2021)
- Harris Şahinleri Optimizasyonu (HHO) (Heidari, ve diğerleri, 2019)
- Martı Optimizasyon Algoritması (SOA) (Dhiman & Kumar, 2019)
- Bulmaca Optimizasyon Algoritması (POA) (Zeidabadi & Dehghani, 2022)
- Güve Arama Algoritması (MS) (Wang, 2018)

- Altın Çakal Optimizasyonu (GJO) (Chopra & Ansari, 2022)
- Lider Harris Şahinleri Optimizasyonu (LHHO) (Naik, Panda, Wunnava, Jena, & Abraham, 2021)
- Karadul Optimizasyon Algoritması (BWOA) (Peña-Delgado, ve diğerleri, 2020)
- Balina Optimizasyon Algoritması (WOA) (Mirjalili & Lewis, 2016)
- Işık Spektrumu Optimizasyonu (LSO) (Abdel-Basset, Mohamed, Sallam, & Chakraborty, 2022)
- Ateşböceği Optimizasyonu (MFO) (Mirjalili, 2015)
- Henry Gazı Çözünürlüğü Optimizasyonu (HGSO) (Hashim, Essam, Mabrouk, Al-Atabany, & Mirjalili, 2019)
- Modifiye Edilmiş Sosyal Grup Optimizasyonu (MSGO) (Naik, Satapathy, & Abraham, 2020)
- Çocuk Çizim Gelişimi Optimizasyonu (CDDO) (Abdulhameed & Rashid, 2022)
- Sinüs Kosinüs Algoritması (SCA) (Mirjalili, 2016)
- Levy Uçuş Dağılımı Algoritması (LFD) (Houssein, Saad, Hashim, Shaban, & Hassaballah, 2020)
- Şempanze Optimizasyon Algoritması (CHIMP) (Khishe & Mosavi, 2020)
- Karadul optimizasyonu (BWO) (Hayyolalam & Kazem, 2020)
- Ağaç Tohum Algoritması (TSA) (Kiran, 2015)
- Kristal Yapı Algoritması (CRYSTAL) (Talatahari, Azizi, Tolouei, Talatahari, & Sareh, 2021)
- Ateş Şahini Optimizasyonu (FHO) (Azizi, Talatahari, & Gandomi, 2023)
- Sıçan Sürüsü Optimizasyonu (RSO) (Dhiman, Garg, Nagar, Kumar, & Dehghani, 2021)
- Meksika Axolotl Optimizasyonu (MAO) (Villuendas-Rey, Velázquez-Rodríguez, Alanis-Tamez, Moreno-Ibarra, & Yáñez-Márquez, 2021)
- Kelebek Optimizasyon Algoritması (BOA) (Arora & Singh, 2019)
- Geçici Arama Optimizasyonu (TSO) (Qais, Hasanien, & Alghuwainem, 2020)
- Dinamik Diferansiyel Dinlendirilmiş Optimizasyonu (DDAO) (Ghafil & Jármai, 2020)
- Ayçiçeği Optimizasyonu (SFO) (Gomes, Cunha, & Ancelotti, 2019)
- Sürünge Arama Algoritması (RSA) (Abualigah, Elaziz, Sumari, Geem, & Gandomi, 2022)

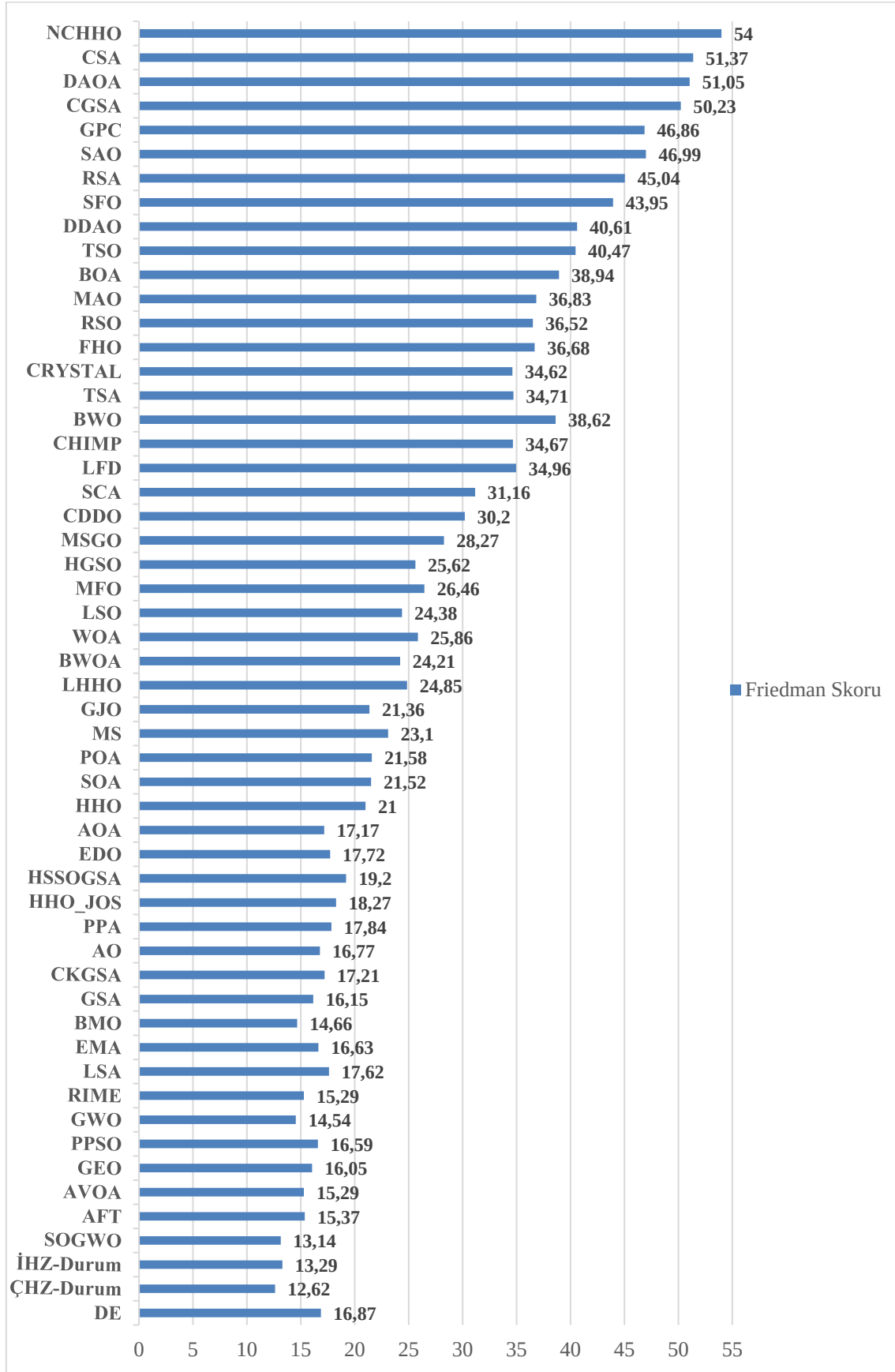
- Koku Ajanı Optimizasyonu (SAO) (Meadows, Mu’Azu, & Salawudeen, 2022)
- Giza Piramitleri İnşası Optimizasyonu (GPC) (Harifi, Mohammadzadeh, Khalilian, & Ebrahimnejad, 2021)
- Kaotik Kütle Sabitli Yerçekimi Arama Algoritması (CGSA) (Mirjalili & Gandomi, 2017)
- Dinamik Aritmetik Optimizasyon Algoritması (DAOA) (Khodadadi, Snasel, & Mirjalili, 2022)
- Karga Arama Algoritması (CSA) (Askarzadeh, 2016)
- Doğrusal Olmayan Tabanlı Kaotik Harris Şahinleri Optimizasyonu (NCHHO) (Dehkordi, Sadiq, Mirjalili, & Ghafoor, 2021)

51 rakip algoritma CEC20 problem setinde 10, 30, 50 ve 100 boyutlarında çalıştırılarak sonuçları alınmıştır. Alınan sonuçlara İHZ-Durum ve ÇHZ-Durum çalışmaları dahil edilerek Friedman analizinde incelenmiştir. Friedman analiz sonuçları Şekil 112 ve Şekil 116 arasındaki şekillerde görülmektedir. Ortalama Friedman sonuçları incelendiğinde (Şekil 116) ÇHZ-Durum çalışması sıralamada 1., ÇHZ-Durum çalışması 5., DE algoritması ise 17. olmuştur. Geliştirilen ÇHZ-Durum çalışması, ortalama Friedman skorların DE algoritmasını geçen 14 rakip algoritmayı geçmiştir. Bu sonuçlar önerdiğimiz yöntemin DE algoritmasının arama performansını güçlü bir şekilde arttırdığını göstermektedir.

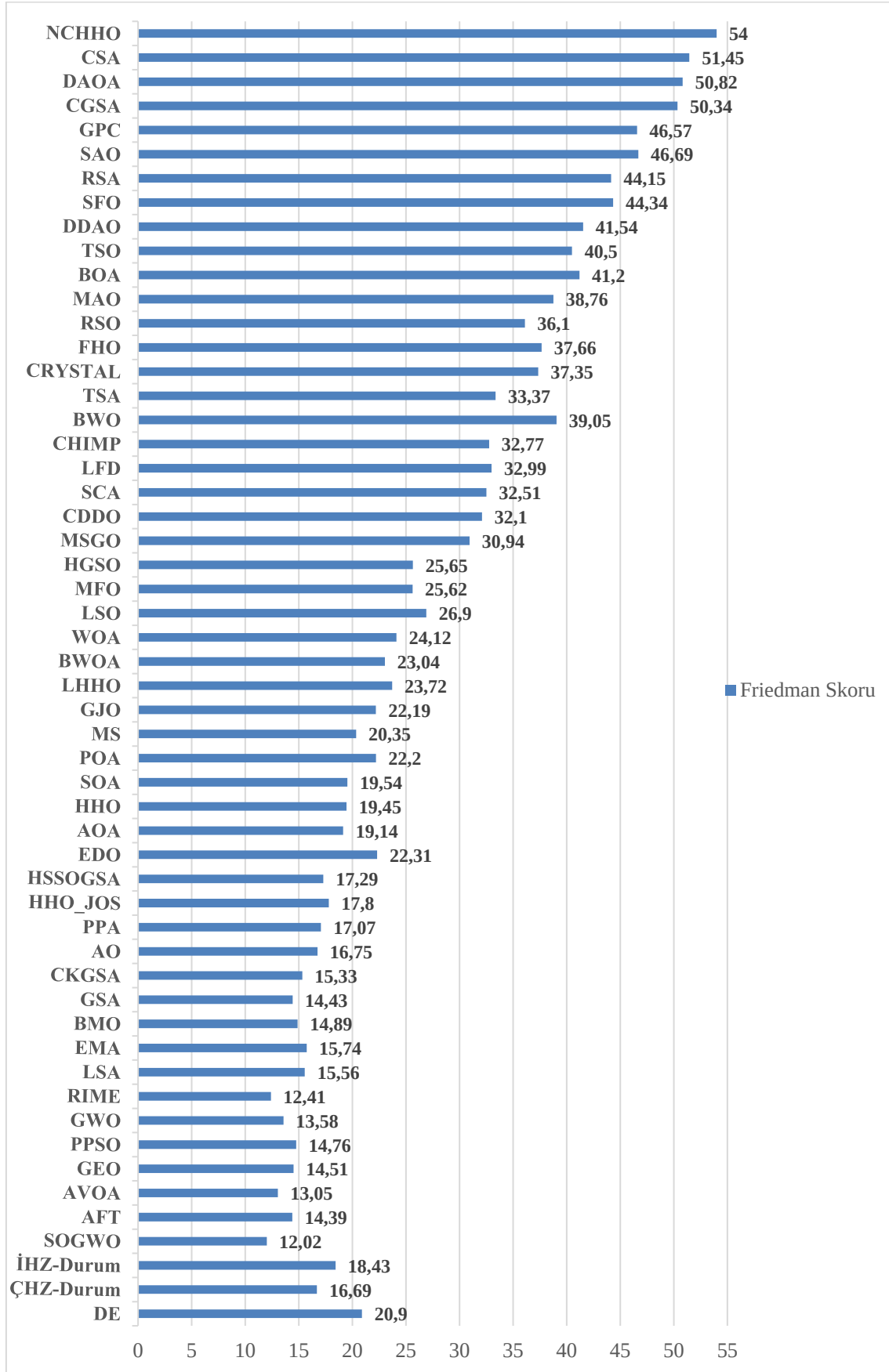
Şekil 112. CEC20 10D rakip algoritmalar Friedman analiz sonucu



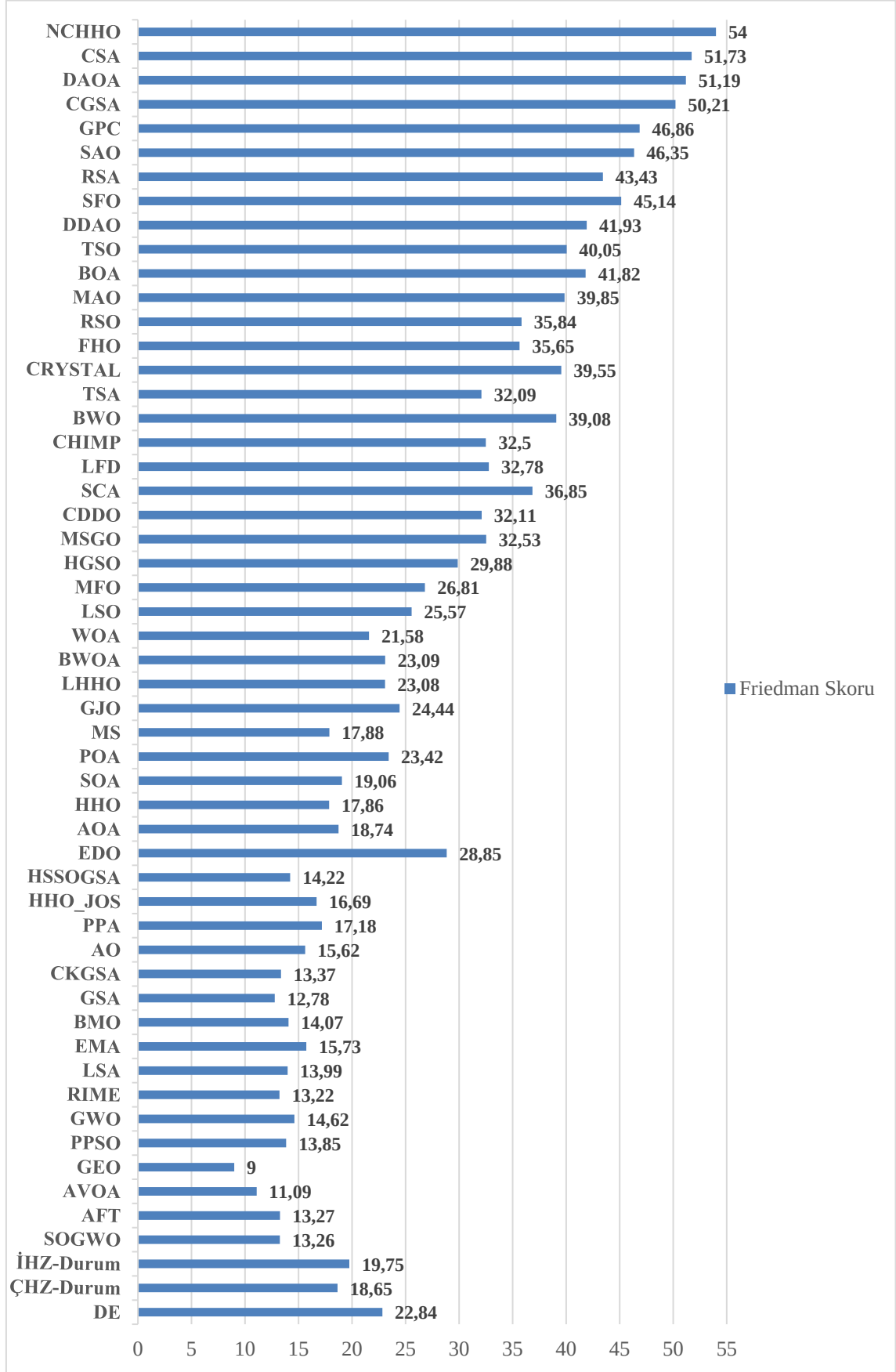
Şekil 113. CEC20 20D rakip algoritmalar Friedman analiz sonucu



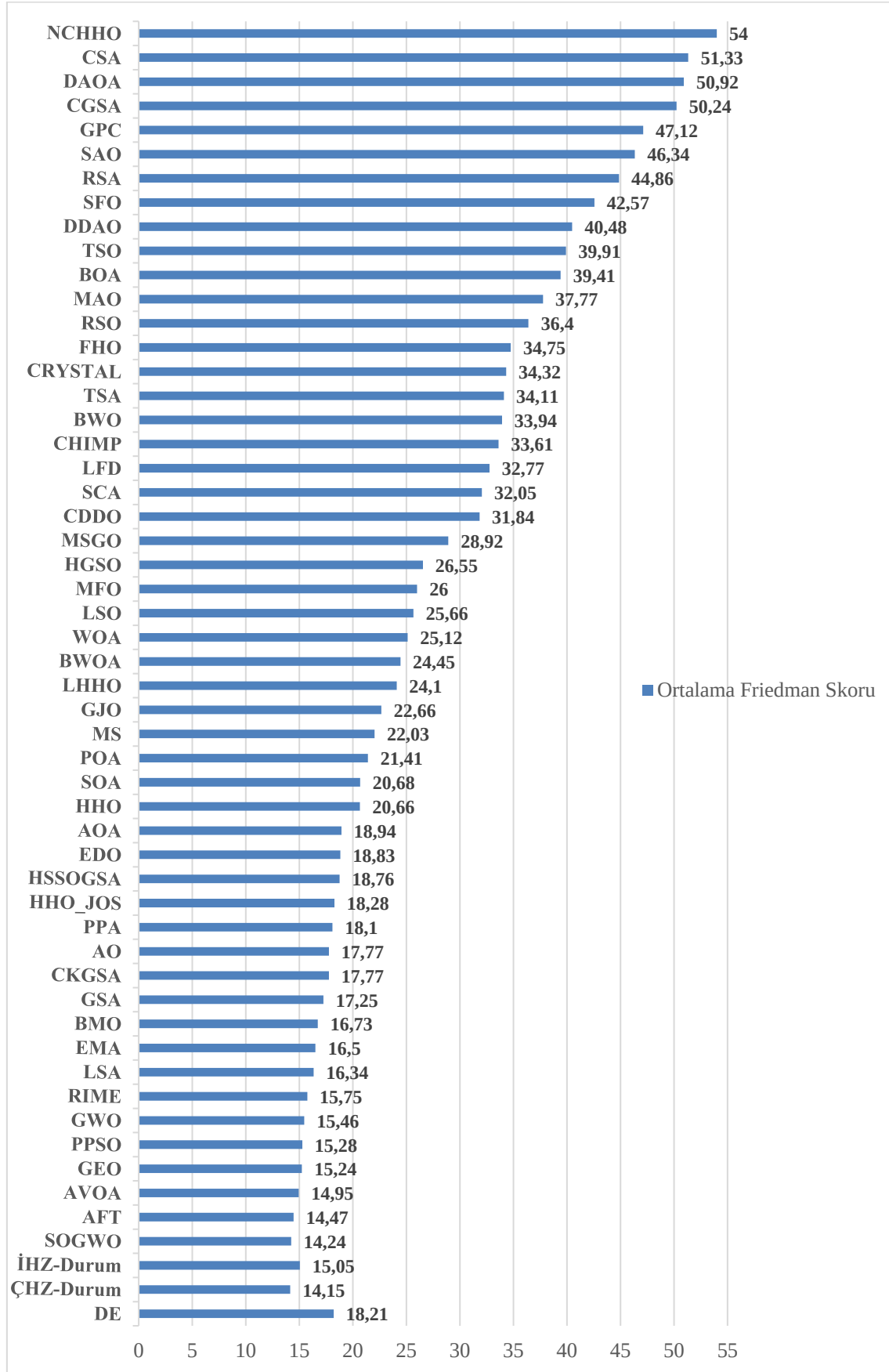
Şekil 114. CEC20 50D rakip algoritmalar Friedman analiz sonucu



Şekil 115. CEC20 100D rakip algoritmalar Friedman analiz sonucu



Şekil 116. CEC20 rakip algoritmalar ortalama boyut Friedman analiz sonucu



3.4. Gerçek Dünya Kısıtlı Mühendislik Problemlerinden Elde Edilen Bulgular

Gerçek dünya problemlerinde, problemlerin boyutu değişken olduğu için, kısıtsız çalışmasında yapılan problem boyutuna göre HED değerlerinin belirlenmesi bu çalışmada yapılamamıştır. Bundan dolayı gerçek dünya kısıtlı mühendislik problemlerinde “İki Halkalı Optimum Zincir Araştırılması” başlığı altında yapılan çalışmalar kullanılacaktır. Problem boyutuna göre HED değeri kullanılmadığından dolayı çalışmada belirlenen 9 HED değeri (0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1, 2, 5) ile iki halkalı durumlar oluşturulacaktır. “İki Halkalı Optimum Zincir Araştırılması” çalışmasında 0.01, 0.02 ve 5 HED değerleri için durumlar oluşturulmuştur. Oluşturulan bu durumlar, kısıtlı mühendislik çalışmasında kullanılacaktır. 0.05, 0.1, 0.2, 0.5, 1, 2 HED değerleri için oluşturulan durumlar Tablo 18’de görülmektedir.

Tablo 18. Kısıtlı mühendislik için oluşturulan iki halkalı durumlar

Durum Adları	HED Değerleri	Zincirdeki Halka Numaraları
Durum 109	0.05	1-4
Durum 135	0.1	
Durum 161	0.2	
Durum 187	0.5	
Durum 213	1	
Durum 239	2	
Durum 110	0.05	2-4
Durum 136	0.1	
Durum 162	0.2	
Durum 188	0.5	
Durum 214	1	
Durum 240	2	
Durum 111	0.05	3-4
Durum 137	0.1	
Durum 163	0.2	
Durum 189	0.5	
Durum 215	1	
Durum 241	2	

Tablo 18'in devamı

Durum 112	0.05	5-4
Durum 138	0.1	
Durum 164	0.2	
Durum 190	0.5	
Durum 216	1	
Durum 242	2	
Durum 113	0.05	6-4
Durum 139	0.1	
Durum 165	0.2	
Durum 191	0.5	
Durum 217	1	
Durum 243	2	
Durum 114	0.05	7-4
Durum 140	0.1	
Durum 166	0.2	
Durum 192	0.5	
Durum 218	1	
Durum 244	2	
Durum 115	0.05	8-4
Durum 141	0.1	
Durum 167	0.2	
Durum 193	0.5	
Durum 219	1	
Durum 245	2	
Durum 116	0.05	9-4
Durum 142	0.1	
Durum 168	0.2	
Durum 194	0.5	
Durum 220	1	
Durum 246	2	

Tablo 18'in devamı

Durum 117	0.05	10-4
Durum 143	0.1	
Durum 169	0.2	
Durum 195	0.5	
Durum 221	1	
Durum 247	2	
Durum 118	0.05	11-4
Durum 144	0.1	
Durum 170	0.2	
Durum 196	0.5	
Durum 222	1	
Durum 248	2	
Durum 119	0.05	12-4
Durum 145	0.1	
Durum 171	0.2	
Durum 197	0.5	
Durum 223	1	
Durum 249	2	
Durum 120	0.05	13-4
Durum 146	0.1	
Durum 172	0.2	
Durum 198	0.5	
Durum 224	1	
Durum 250	2	
Durum 121	0.05	14-4
Durum 147	0.1	
Durum 173	0.2	
Durum 199	0.5	
Durum 225	1	
Durum 251	2	

Tablo 18'in devamı

Durum 122	0.05	15-4
Durum 148	0.1	
Durum 174	0.2	
Durum 200	0.5	
Durum 226	1	
Durum 252	2	
Durum 123	0.05	16-4
Durum 149	0.1	
Durum 175	0.2	
Durum 201	0.5	
Durum 227	1	
Durum 253	2	
Durum 124	0.05	17-4
Durum 150	0.1	
Durum 176	0.2	
Durum 202	0.5	
Durum 228	1	
Durum 254	2	
Durum 125	0.05	18-4
Durum 151	0.1	
Durum 177	0.2	
Durum 203	0.5	
Durum 229	1	
Durum 255	2	
Durum 126	0.05	19-4
Durum 152	0.1	
Durum 178	0.2	
Durum 204	0.5	
Durum 230	1	
Durum 256	2	

Tablo 18'in devamı

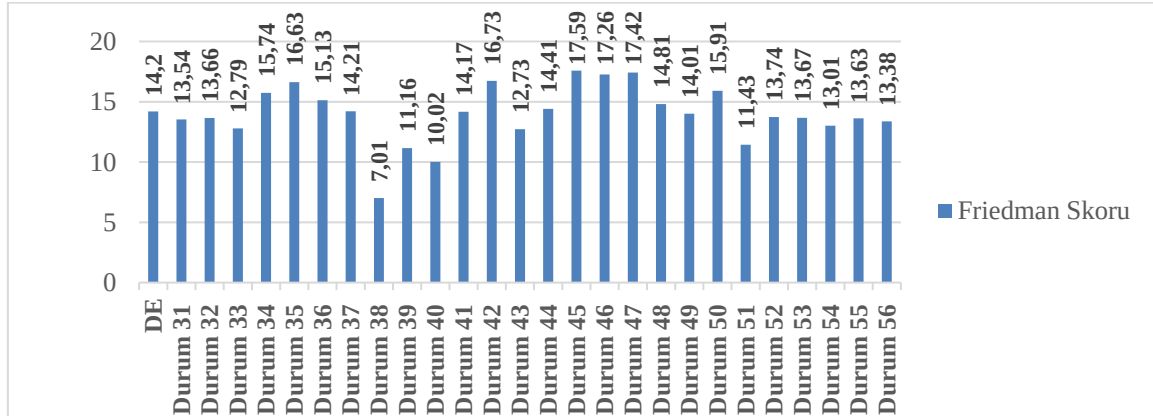
Durum 127	0.05	20-4
Durum 153	0.1	
Durum 179	0.2	
Durum 205	0.5	
Durum 231	1	
Durum 257	2	
Durum 128	0.05	21-4
Durum 154	0.1	
Durum 180	0.2	
Durum 206	0.5	
Durum 232	1	
Durum 258	2	
Durum 129	0.05	22-4
Durum 155	0.1	
Durum 181	0.2	
Durum 207	0.5	
Durum 233	1	
Durum 259	2	
Durum 130	0.05	23-4
Durum 156	0.1	
Durum 182	0.2	
Durum 208	0.5	
Durum 234	1	
Durum 260	2	
Durum 131	0.05	24-4
Durum 157	0.1	
Durum 183	0.2	
Durum 209	0.5	
Durum 235	1	
Durum 261	2	

Tablo 18'in devamı

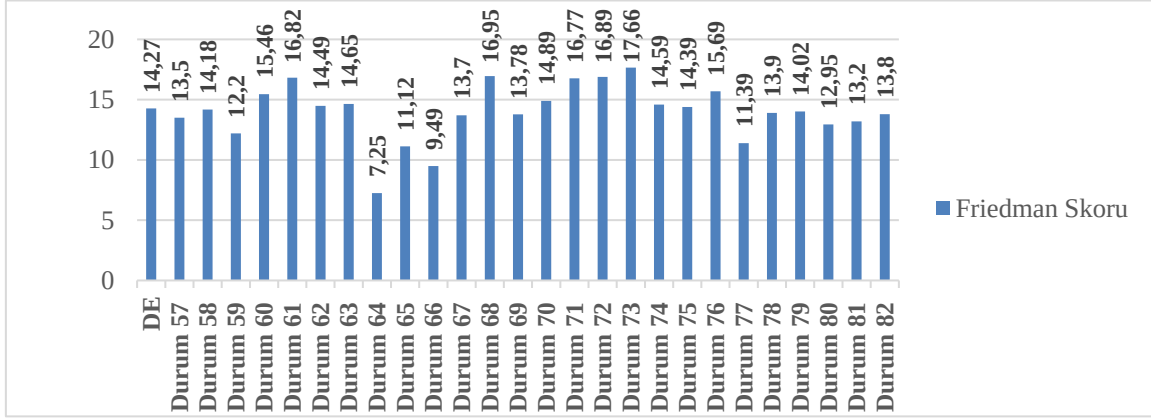
Durum 132	0.05	25-4
Durum 158	0.1	
Durum 184	0.2	
Durum 210	0.5	
Durum 236	1	
Durum 262	2	
Durum 133	0.05	26-4
Durum 159	0.1	
Durum 185	0.2	
Durum 211	0.5	
Durum 237	1	
Durum 263	2	
Durum 134	0.05	27-4
Durum 160	0.1	
Durum 186	0.2	
Durum 212	0.5	
Durum 238	1	
Durum 264	2	

Oluşturulan durumlar 10 adet kısıtlı mühendislik problemlerinde çalıştırılmıştır. Alınan sonuçlar Friedman yöntemiyle analiz edilmiştir. Analiz sonuçları Şekil 117 ve Şekil 125 arasındaki şekillerde verilmiştir.

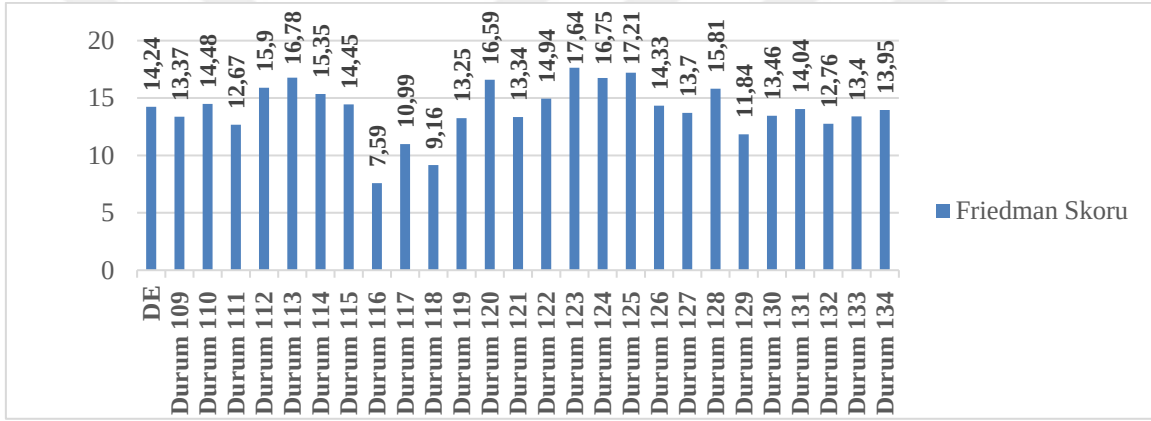
Şekil 117. Kısıtlı mühendislik 0.01 HED değeri için Friedman analiz sonucu



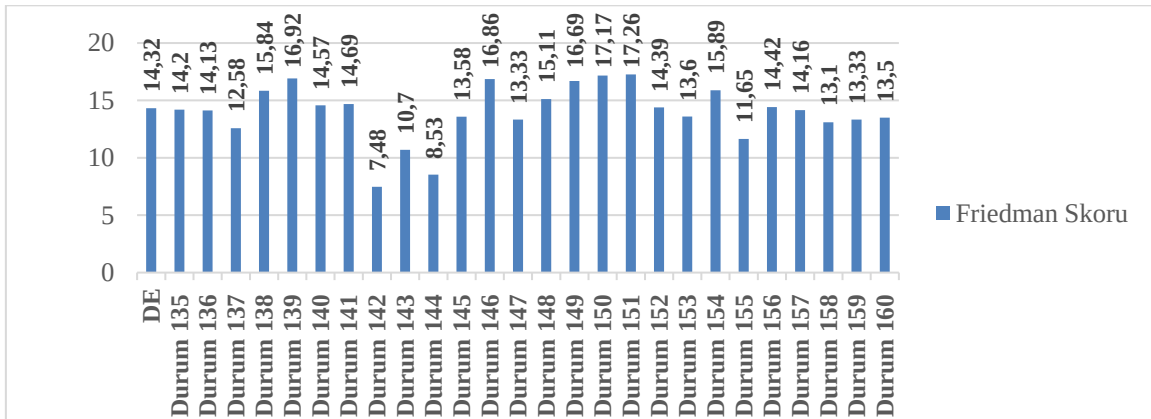
Şekil 118. Kısıtlı mühendislik 0.02 HED değeri için Friedman analiz sonucu



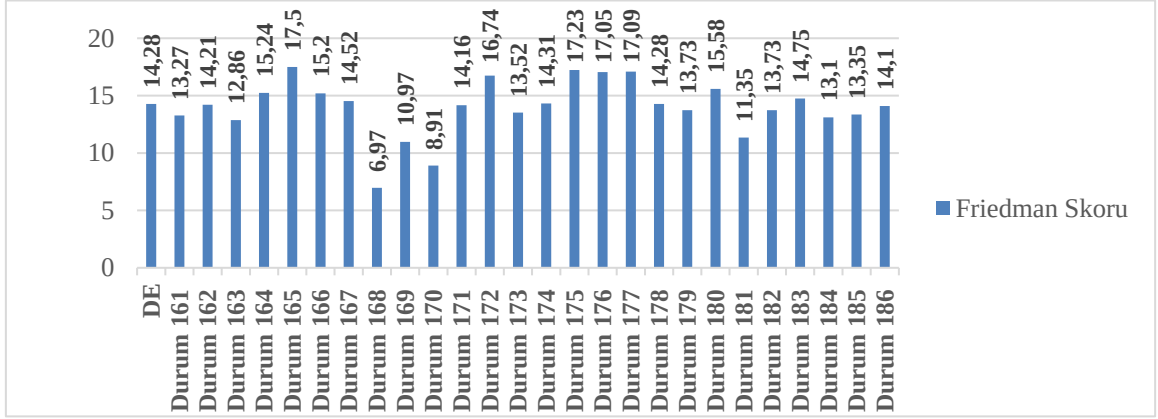
Şekil 119. Kısıtlı mühendislik 0.05 HED değeri için Friedman analiz sonucu



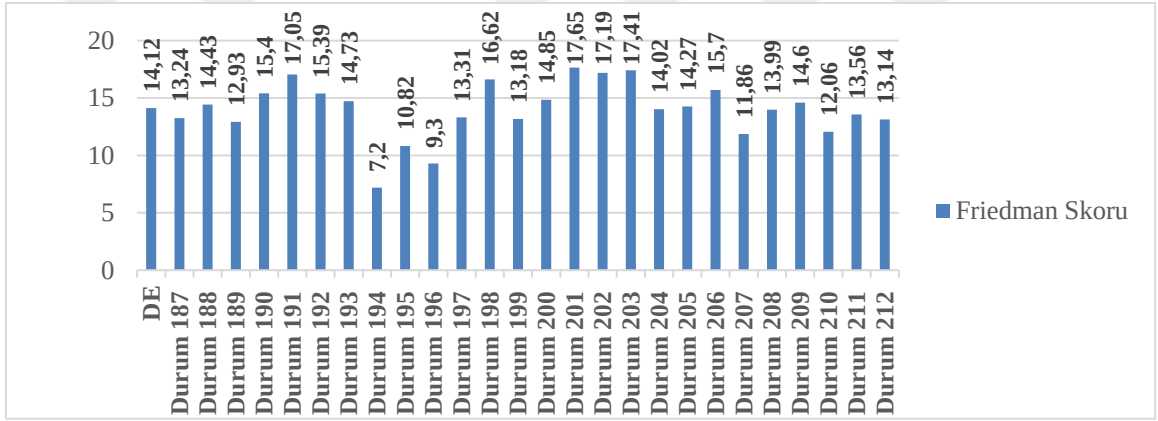
Şekil 120. Kısıtlı mühendislik 0.1 HED değeri için Friedman analiz sonucu



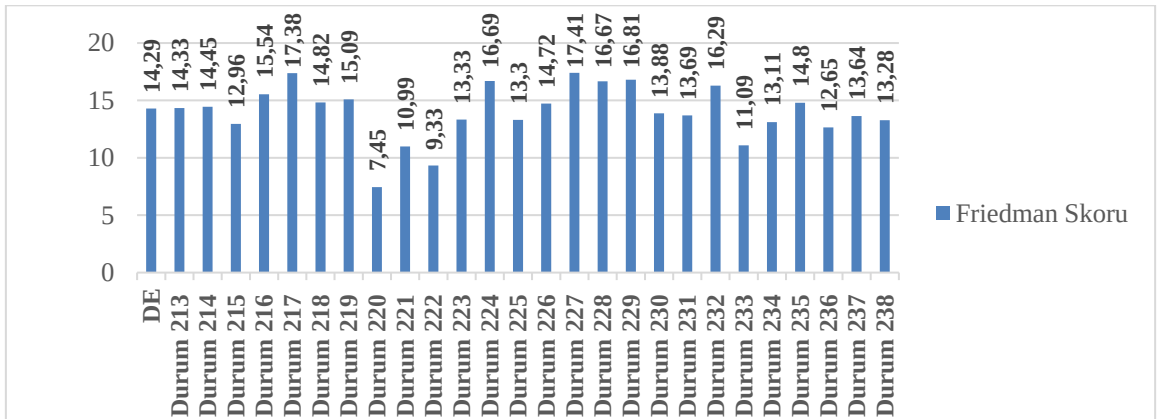
Şekil 121. Kısıtlı mühendislik 0.2 HED değeri için Friedman analiz sonucu



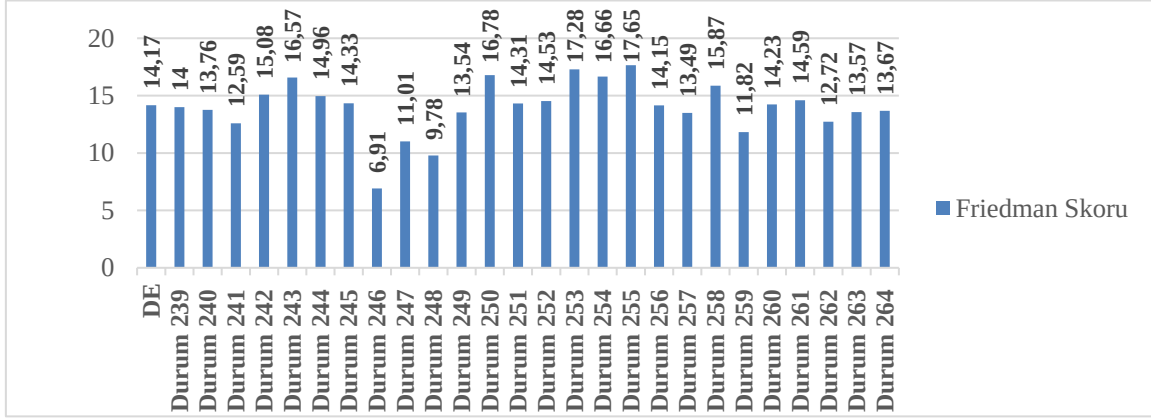
Şekil 122. Kısıtlı mühendislik 0.5 HED değeri için Friedman analiz sonucu



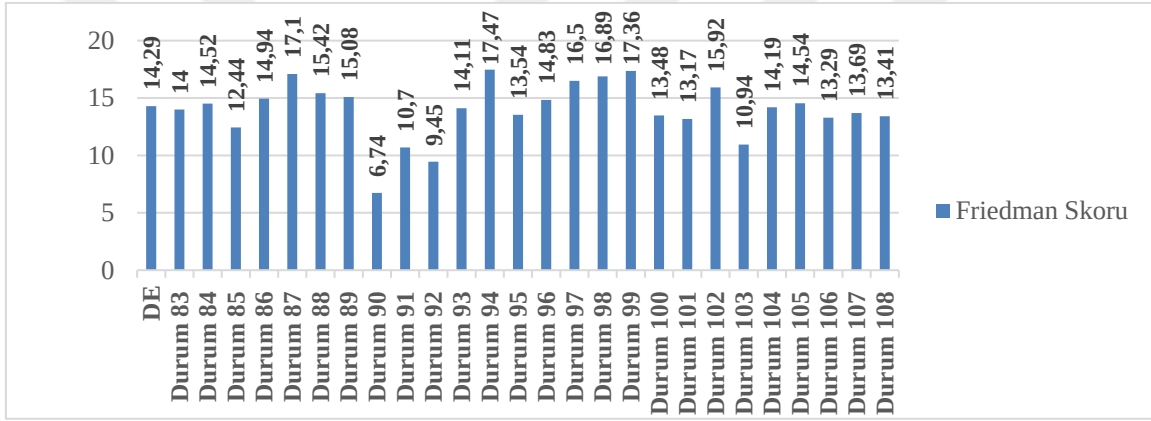
Şekil 123. Kısıtlı mühendislik 1 HED değeri için Friedman analiz sonucu



Şekil 124. Kısıtlı mühendislik 2 HED değeri için Friedman analiz sonucu



Şekil 125. Kısıtlı mühendislik 5 HED değeri için Friedman analiz sonucu



Şekil 117 ve Şekil 125 arasındaki şekiller incelendiğinde en iyi Friedman skor değerleri farklı HED değerleriyle birlikte 9-4 halka numaralarıyla oluşturulmuş zincirli durumlardır. Bu durumlar sırasıyla:

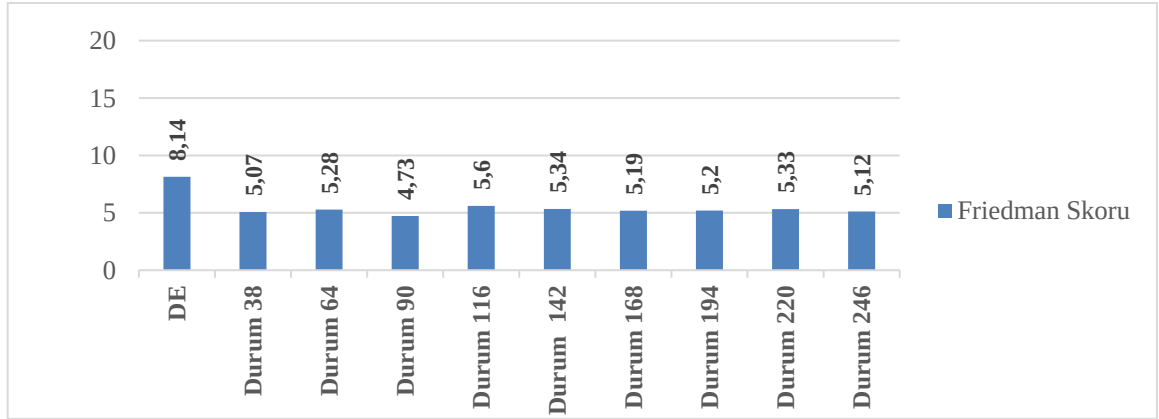
- 0.01 için Durum 38,
- 0.02 için Durum 64,
- 0.05 için Durum 116,
- 0.1 için Durum 142,
- 0.2 için Durum 168,
- 0.5 için Durum 194,
- 1 için Durum 220,
- 2 için Durum 246,
- 5 için Durum 90'dır.

Belirlenen 9 durum ile DE algoritmasının kısıtlı mühendislik problemlerinde performansları karşılaştırılmıştır. Mühendislik problemlerinde alınan sonuçlar Wilcoxon ve Friedman yöntemleri ile analiz edilmiştir. Analiz sonuçları Tablo 19 ve Şekil 126’da görülmektedir.

Tablo 19. En iyi durumlar için kısıtlı mühendislik Wilcoxon analiz sonuçları

DE vs.+/-	
DE VS	Wilcoxon Sonuçları
Durum 38	9/1/0
Durum 64	9/1/0
Durum 116	10/0/0
Durum 142	10/0/0
Durum 168	10/0/0
Durum 194	10/0/0
Durum 220	8/2/0
Durum 246	9/1/0
Durum 90	10/0/0

Şekil 126. En iyi durumlar için kısıtlı mühendislik Friedman analiz sonuçları



Sonuçlar incelendiğinde, HED değerindeki değişimin algoritmaya etkisi net bir şekilde görülebilmektedir. En iyi Friedman skoruna sahip 5 HED değerini kullanan “Durum 90”, 10 kısıtlı mühendislik probleminin 10’unda da baz algoritmadan daha yüksek performans göstermiştir. Bu durum önerilen yöntemin kısıtlı mühendislik problemlerinde de DE algoritmasının arama performansını önemli ölçüde arttırdığını göstermektedir.

4. SONUÇLAR

Yüzde hata değişimi esaslı arama yöntemi, belirlenen 9 HED değeri üzerinde gerçekleştirilen çalışmalarda, kısıtsız ve kısıtlı test problemlerinde incelenmiştir. Bu kapsamlı analizler neticesinde, önerilen yöntemin DE algoritması üzerindeki olumlu etkileri açıkça gözlemlenmiştir.

Kısıtsız test problemlerinde kullanılan CEC20 karşılaştırma süitindeki 4 farklı tipteki 10 farklı optimizasyon probleminin çeşitli boyutları üzerinde yapılan deneylerde, önerilen yöntemin performansını artırdığı görülmüştür. Bu gelişmeler, özellikle belirli boyutlardaki problemlerde elde edilen başarıların, yöntemin genel etkinliğini vurgulamaktadır. Kısıtlı test problemlerinde ise, kullanılan tüm problemlerde gözlemlenen performans artışı, önerilen yöntemin kısıtlı optimizasyon problemleri üzerindeki güçlü etkilerini belirginleştirmiştir.

Yapılan kapsamlı çalışmalar sonucunda elde edilen başarılar, literatürdeki 51 farklı algoritma ile yapılan karşılaştırmada da ön plana çıkmıştır. Bu karşılaştırmada, toplamda 54 algoritma değerlendirilmiş olup, özellikle geliştirilen iki özel durum olan ÇHZ ve İHZ ile elde edilen sonuçlar dikkat çekicidir. Sıralamada ÇHZ birinci sıradayken, önerilen yöntem DE algoritmasından sonra gelen 14 farklı algoritmayı geride bırakarak dikkat çekici bir performans sergilemiştir. Görüldüğü gibi, önerilen yöntem, diğer rekabetçi algoritmaları geride bırakarak başarı sıralamasında önemli bir konumda yer almaktadır. Bu durum, yüzde hata değişimi esaslı arama yönteminin, çeşitli optimizasyon problemlerinde güçlü ve etkili bir performans sergilediğini ortaya koymaktadır.

Sonuç olarak, elde edilen analiz sonuçları, yüzde hata değişimi esaslı arama yönteminin DE algoritmasıyla entegrasyonunun, algoritmanın performansını artırıcı yönde önemli bir etki yarattığını ortaya koymaktadır. Kısıtsız ve kısıtlı problemlerde elde edilen başarılar, önerilen yöntemin geniş bir uygulama yelpazesi üzerinde etkili bir şekilde kullanılabileceğini göstermektedir. Bu durum hem teorik hem de pratik açıdan, optimizasyon problemlerine yönelik yeni ve etkili bir yaklaşım sunmaktadır.

5. ÖNERİLER

Bu tez çalışmasında MSA algoritmalarının yakınsama denklemlerinin tasarımı için yeni ve etkili bir yöntem olan yüzde hata değişimi esaslı yakınsama denklemi tasarlama önerilmiştir. Önerilen yöntemin başlıca yeniliği, MSA algoritmalarında yakınsama denkleminin tasarımında birçok adımdan oluşan yeni bir sürecin tanımlanmasıdır. Bu tez konusu ile çalışma yapmak isteyen uygulayıcı ve araştırmacılara iletmek istenilen öneriler aşağıda belirtilmiştir:

- HED değeri için kullanılan 9 farklı parametre optimum parametreler değildir. Problem türüne göre farklı parametreler ayarlanabilir.
- HED değerleri arama süreci yaşam döngüsü boyunca sabit tutulmaktadır. Arama süreci evrelere bölünerek, farklı evrelerde farklı HED değerleri kullanılabilir. Arama sürecinin sonlarına doğru en iyi çözüm adayının uygunluk değerindeki iyileşme miktarı azalmaktadır. Bundan dolayı arama sürecinin başlarında yüksek HED değeri, sonlara doğru ise düşük HED değeri uygulanabilir.
- Tez çalışmasında yüzde hata değişimi ile algoritmanın arama denklemlerinden oluşturulmuş halkaları kullanan zincirler aktif edilmektedir. Yüzde hata değişimine göre algoritmanın arama denklemlerinden bağımsız denklemler tasarlanabilir. Farklı MSA algoritmaları tasarlanarak halka olarak kullanılabilir.

6. KAYNAKLAR

- Abdel-Basset, M., El-Shahat, D., Jameel, M., & Abouhawwash, M. (2023). Exponential distribution optimizer (EDO): a novel math-inspired algorithm for global optimization and engineering problems. *Artificial Intelligence Review*, 1-72.
- Abdel-Basset, M., Mohamed, R., Sallam, K. M., & Chakraborty, R. K. (2022). Light spectrum optimizer: a novel physics-inspired metaheuristic optimization algorithm. *Mathematics*, 3466.
- Abdollahzadeh, B., Gharehchopogh, F. S., & Mirjalili, S. (2021). African vultures optimization algorithm: A new nature-inspired metaheuristic algorithm for global optimization problems. *Computers & Industrial Engineering*, 107408.
- Abdulhameed, S., & Rashid, T. A. (2022). Child drawing development optimization algorithm based on child's cognitive development. *Arabian Journal for Science and Engineering*, 1337-1351.
- Abualigah, L., Elaziz, M. A., Sumari, P., Geem, Z. W., & Gandomi, A. H. (2022). Reptile Search Algorithm (RSA): A nature-inspired meta-heuristic optimizer. *Expert Systems with Applications*, 116158.
- Abualigah, L., Yousri, D., Elaziz, M. A., Ewees, A. A., Al-qaness, M. A., & Gandomi, A. H. (2021). Aquila optimizer: a novel meta-heuristic optimization algorithm. *Computers & Industrial Engineering*, 107250.
- Arini, F. Y., Chiewchanwattana, S., Soomlek, C., & Sunat, K. (2022). Joint Opposite Selection (JOS): A premiere joint of selective leading opposition and dynamic opposite enhanced Harris' hawks optimization for solving single-objective problems. *Expert Systems with Applications*, 116001.
- Arora, S., & Singh, S. (2019). Butterfly optimization algorithm: a novel approach for global optimization. *Soft Computing*, 715-734.
- Askarzadeh, A. (2016). A novel metaheuristic method for solving constrained engineering optimization problems: crow search algorithm. *Computers & structures*, 1-12.
- Azizi, M., Talatahari, S., & Gandomi, A. H. (2023). Fire Hawk Optimizer: A novel metaheuristic algorithm. *Artificial Intelligence Review*, 287-363.
- Beyer, H. G., & Schwefel, H. P. (2002). Evolution strategies--a comprehensive introduction. *Natural computing*, s. 3-52.
- Braik, M., Ryalat, M. H., & Al-Zoubi, H. (2022). A novel meta-heuristic algorithm for solving numerical optimization problems: Ali Baba and the forty thieves. *Neural Computing and Applications*, 409-455.

- Cheng, M.-Y., & Prayogo, D. (2014). Symbiotic organisms search: a new metaheuristic optimization algorithm. *Computers & Structures*, s. 98-112.
- Chopra, N., & Ansari, M. M. (2022). Golden jackal optimization: A novel nature-inspired optimizer for engineering applications. *Expert Systems with Applications*, 116924.
- Dehkordi, A. A., Sadiq, A. S., Mirjalili, S., & Ghafoor, K. Z. (2021). Nonlinear-based chaotic harris hawks optimizer: algorithm and internet of vehicles application. *Applied Soft Computing*, 107574.
- Derrac, J., García, S., Molina, D., & Herrera, F. (2011). A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation*, 3-18.
- Dhargupta, S., Ghosh, M., Mirjalili, S., & Sarkar, R. (2020). Selective opposition based grey wolf optimization. *Expert Systems with Applications*, 113389.
- Dhiman, G., & Kumar, V. (2019). Seagull optimization algorithm: Theory and its applications for large-scale industrial engineering problems. *Knowledge-based systems*, 169-196.
- Dhiman, G., Garg, M., Nagar, A., Kumar, V., & Dehghani, M. (2021). A novel algorithm for global optimization: rat swarm optimizer. *Journal of Ambient Intelligence and Humanized Computing*, 8457-8482.
- Dorigo, M., Birattari, M., & Stutzle, T. (2006). Ant colony optimization. *IEEE computational intelligence magazine*, s. 28-39.
- Dosoglu, M. K., Kahraman, H. T., Sonmez, Y., Guvenc, U., & Duman, S. (2018). Symbiotic organisms search optimization algorithm for economic/emission dispatch problem in power systems. *Neural Computing and Applications*, s. 721-737.
- Erol, O. K., & Eksin, I. (2006). A new optimization method: big bang--big crunch. *Advances in engineering software*, s. 106-111.
- Fong, S., Yang, X.-s., & Deb, S. (2018). How Meta-heuristic Algorithms Contribute to Deep Learning in the Hype of Big Data Analytics. *Progress in Intelligent Computing Techniques: Theory, Practice, and Applications*, s. 3-25.
- Formato, R. A. (2009). Central force optimization: A new deterministic gradient-like optimization metaheuristic. *Opsearch*, s. 25-51.
- Ghafil, H. N., & Jármai, K. (2020). Dynamic differential annealed optimization: New metaheuristic optimization algorithm for engineering applications. *Applied Soft Computing*, 106392.
- Ghasemi, M., Akbari, E., Rahimnejad, A., Razavi, S. E., Ghavidel, S., & Li, L. (2019). Phasor particle swarm optimization: a simple and efficient variant of PSO. *Soft Computing*, 9701-9718.

- Ghorbani, N., & Babaei, E. (2014). Exchange market algorithm. *Applied soft computing*, 177-187.
- Gomes, G. F., Cunha, S. S., & Ancelotti, A. C. (2019). A sunflower optimization (SFO) algorithm applied to damage identification on laminated composite plates. *Engineering with Computers*, 619-626.
- Gupta, S., & Deep, K. (2019). A hybrid self-adaptive sine cosine algorithm with opposition based learning. *Expert Systems with Applications*, 210-230.
- Harifi, S., Mohammadzadeh, J., Khalilian, M., & Ebrahimnejad, S. (2021). Giza Pyramids Construction: an ancient-inspired metaheuristic algorithm for optimization. *Evolutionary Intelligence*, 1743-1761.
- Hashim, F. A., E. H., Mabrouk, M. S., Al-Atabany, W., & Mirjalili, S. (2019). Henry gas solubility optimization: A novel physics-based algorithm. *Future Generation Computer Systems*, 646-667.
- Hashim, F. A., Hussain, K., Houssein, E. H., Mabrouk, M. S., & Al-Atabany, W. (2021). Archimedes optimization algorithm: a new metaheuristic algorithm for solving optimization problems. *Applied Intelligence*, 1531-1551.
- Hatamlou, A. (2013). Black hole: A new heuristic optimization approach for data clustering. *Information sciences*, s. 175-184.
- Hayyolalam, V., & Kazem, A. A. (2020). Black widow optimization algorithm: a novel meta-heuristic approach for solving engineering optimization problems. *Engineering Applications of Artificial Intelligence*, 103249.
- Heidari, A. A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M., & Chen, H. (2019). Harris hawks optimization: Algorithm and applications. *Future generation computer systems*, 849-872.
- Holland, J. H. (1992). *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press.
- Houssein, E. H., Saad, M. R., Hashim, F. A., Shaban, H., & Hassaballah, M. (2020). Levy flight distribution: A new metaheuristic algorithm for solving engineering optimization problems. *Engineering Applications of Artificial Intelligence*, 103731.
- Jian Zhang, G. D., Qin, S., & Fu, J. (2019). Review of job shop scheduling research and its new perspectives under Industry 4.0. *Journal of Intelligent Manufacturing*, s. 1809-1830.
- Kahraman, H. T. (2016). A novel and powerful hybrid classifier method: Development and testing of heuristic k-nn algorithm with fuzzy distance metric. *Data & Knowledge Engineering*, s. 44-59.
- Kahraman, H. T., Bakir, H., Duman, S., Katı, M., Aras, S., & Guvenc, U. (2022). Dynamic FDB selection method and its application: modeling and optimizing of directional overcurrent relays coordination. *Applied Intelligence*, s. 1-36.

- Kahraman, H. T., Sagioglu, S., & Colak, I. (2013). The development of intuitive knowledge classifier and the modeling of domain dependent data. *Knowledge-Based Systems*, s. 283-295.
- Kahraman, T., Aras, S., & Gedikli, E. (2020). Fitness-distance balance (FDB): a new selection method for meta-heuristic search algorithms. *Knowledge-Based Systems*, 105169.
- Karaboga, D., & Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *Journal of global optimization*, s. 459-471.
- Kaveh, A., & Dadras, A. (2017). A novel meta-heuristic optimization algorithm: thermal exchange optimization. *Advances in engineering software*, 69-84.
- Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. *International conference on neural networks* (s. 1942-1948). Australia: IEEE .
- Khishe, M., & Mosavi, M. (2020). Chimp optimization algorithm. *Expert systems with applications*, 113338.
- Khodadadi, N., Snasel, V., & Mirjalili, S. (2022). Dynamic arithmetic optimization algorithm for truss optimization under natural frequency constraints. *IEEE Access*, 16188--16208.
- Kiran, M. S. (2015). TSA: Tree-seed algorithm for continuous optimization. *Expert Systems with Applications*, 6686-6698.
- Kumar, A., Wu, G., Ali, M. Z., Mallipeddi, R., Suganthan, P. N., & Das, S. (2020). A test-suite of non-convex constrained optimization problems from the real-world and some baseline results. *Swarm and Evolutionary Computation*, 100693.
- Lai, W., Zhou, M., Hu, F., Bian, K., & Song, Q. (2019). A New DBSCAN Parameters Determination Method Based on Improved MVO. *IEEE Access*, s. 104085-104095.
- Liang, J. J., Qu, B. Y., & Suganthan, P. N. (2013). Problem definitions and evaluation criteria for the CEC 2014 special session and competition on single objective real-parameter numerical optimization. *Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore*, 2.
- Lin, X., Zhang, F., & Xu, L. (2018). Design of gear reducer based on FOA optimization algorithm. *Advances in Smart Vehicular Technology, Transportation, Communication and Applications: Proceedings of the First International Conference on Smart Vehicular Technology, Transportation, Communication and Applications* (s. 240-247). Kaohsiung, Taiwan: Springer.
- Martin, L., Leblanc, R., & Toan, N. K. (1993). Tables for the Friedman rank test. *Canadian journal of statistics*, 39-43.

- McKay, M. D., Beckman, R. J., & Conover, W. J. (tarih yok). A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, s. 55-61.
- Meadows, O. A., Mu'Azur, M. B., & Salawudeen, A. T. (2022). *A Smell Agent Optimization Approach to Capacitated Vehicle Routing Problem for Solid Waste Collection*. 2022 IEEE Nigeria 4th International Conference on Disruptive Technologies for Sustainable Development (NIGERCON).
- Mirjalili, S. (2015). Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowledge-based systems*, 228-249.
- Mirjalili, S. (2016). SCA: a sine cosine algorithm for solving optimization problems. *Knowledge-based systems*, 120-133.
- Mirjalili, S., & Gandomi, A. H. (2017). Chaotic gravitational constants for the gravitational search algorithm. *Applied soft computing*, 407-419.
- Mirjalili, S., & Lewis, A. (2016). The whale optimization algorithm. *Advances in engineering software*, 51-67.
- Mirjalili, S., Gandomi, A. H., Mirjalili, S. Z., Saremi, S., Faris, H., & Mirjalili, S. M. (2017). Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems. *Advances in engineering software*, s. 163-191.
- Mirjalili, S., Lewis, A., & Mirjalili, S. M. (2014). Grey wolf optimizer. *Advances in engineering software*, 46-61.
- Mittal, H., Pal, R., Kulhari, A., & Saraswat, M. (2016). Chaotic kbest gravitational search algorithm (ckgsa). *2016 ninth international conference on contemporary computing (IC3)*, 1-6.
- Mohamed, A.-A. A., Hassan, S., Hemeida, A., Alkhalaf, S., Mahmoud, M., & Eldin, A. M. (2020). Parasitism--Predation algorithm (PPA): A novel approach for feature selection. *Ain Shams Engineering Journal*, 293-308.
- Mohammadi-Balani, A., Nayeri, M. D., Azar, A., & Taghizadeh-Yazdi, M. (2021). Golden eagle optimizer: A nature-inspired metaheuristic algorithm. *Computers & Industrial Engineering*, 107050.
- Naik, A., Satapathy, S. C., & Abraham, A. (2020). Modified social group optimization—a meta-heuristic algorithm to solve short-term hydrothermal scheduling. *Applied Soft Computing*, 106524.
- Naik, M. K., Panda, R., Wunnavu, A., Jena, B., & Abraham, A. (2021). A leader Harris hawks optimization for 2-D Masi entropy-based multilevel image thresholding. *Multimedia Tools and Applications*, 1-41.
- Özbay, F. A., & Özbay, E. (2021). MARTI OPTİMİZASYON ALGORİTMASININ KISITLI MÜHENDİSLİK TASARIM PROBLEMLERİ İÇİN PERFORMANS ANALİZİ. *Adıyaman Üniversitesi Mühendislik Bilimleri Dergisi*, s. 469-485.

- Peña-Delgado, A. F., Peraza-Vázquez, H., Almazán-Covarrubias, J. H., Cruz, N. T., García-Vite, P. M., Morales-Cepeda, A. B., & Ramirez-Arredondo, J. M. (2020). A novel bio-inspired algorithm applied to selective harmonic elimination in a three-phase eleven-level inverter. *Mathematical Problems in Engineering*, 1-10.
- Pham, D. T., & Karaboga, D. (2000). Intelligent Optimization Techniques. *Springer*, s. 302.
- Pham, H. N., & Triantaphyllou, E. (2009). An application of a new meta-heuristic for optimizing the classification accuracy when analyzing some medical datasets. *Expert Systems with Applications*, s. 9240-9249.
- Qais, M. H., Hasanien, H. M., & Alghuwainem, S. (2020). Transient search optimization: a new meta-heuristic optimization algorithm. *Applied Intelligence*, 3926-3941.
- Rashedi, E., Nezamabadi-Pour, H., & Saryazdi, S. (2009). GSA: a gravitational search algorithm. *Information sciences*, s. 2232-2248.
- Ravindran, A., Reklaitis, G. V., & Ragsdell, K. M. (2006). *Engineering optimization: methods and applications*. John Wiley & Sons.
- Shareef, H., Ibrahim, A. A., & Mutlag, A. H. (2015). Lightning search algorithm. *Applied Soft Computing*, 315-333.
- Shehadeh, H. A. (2021). A hybrid sperm swarm optimization and gravitational search algorithm (HSSOGSA) for global optimization. *Neural Computing and Applications*, 11739-11752.
- Storn, R. (1995). Differential evolution-a simple and efficient adaptive scheme for global optimization over continuous spaces. *Technical report, International Computer Science Institute*.
- Su, H., Zhao, D., Heidari, A. A., Liu, L., Zhang, X., Mafarja, M., & Chen, H. (2023). RIME: A physics-based optimization. *Neurocomputing*, 183-214.
- Suganthan, P. N., Hansen, N., Liang, J. J., Deb, K., Chen, Y.-P., Auger, A., & Tiwari, S. (2005). Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization. *KanGAL report*, 2005.
- Sulaiman, M. H., Mustaffa, Z., Saari, M. M., & Daniyal, H. (2020). Barnacles mating optimizer: A new bio-inspired algorithm for solving engineering optimization problems. *Engineering Applications of Artificial Intelligence*, 103330.
- Talatahari, S., Azizi, M., Tolouei, M., Talatahari, B., & Sareh, P. (2021). Crystal structure algorithm (CryStAl): a metaheuristic optimization method. *IEEE Access*, 71244-71261.
- Tayal, A., & Singh, S. P. (2018). Integrating big data analytic and hybrid firefly-chaotic simulated annealing approach for facility layout problem. *Annals of Operations Research*, s. 489-514.

- Villuendas-Rey, Y., Velázquez-Rodríguez, J. L., Alanis-Tamez, M. D., Moreno-Ibarra, M.-A., & Yáñez-Márquez, C. (2021). Mexican axolotl optimization: A novel bioinspired heuristic. *Mathematics*, 781.
- Wang, G.-G. (2018). Moth search algorithm: a bio-inspired metaheuristic algorithm for global optimization problems. *Memetic Computing*, 151-164.
- Yang, X.-S. (2010). *Nature-inspired metaheuristic algorithms*. Luniver press.
- Yue, C. T., Price, K. V., Suganthan, P. N., Liang, J. J., Ali, M. Z., Qu, B. Y., . . . Biswas, P. P. (2019). *Problem Definitions and Evaluation Criteria for the CEC 2020 Special Session and Competition on Single Objective Bound Constrained Numerical Optimization*. Zhengzhou University.
- Zeidabadi, F. A., & Dehghani, M. (2022). POA: Puzzle Optimization Algorithm. *International Journal of Intelligent Engineering & Systems*.

7. EKLER

7.1. Ek 1. Gerçek Dünya Kısıtlı Mühendislik Test Problemleri

Germe/Sıkıştırma Yay Tasarımı (Tension/Compression Spring Design):

Germe/Sıkıştırma yayı tasarımı kısıtlı mühendislik probleminin (Kaveh & Dadras, 2017) amaç fonksiyonu, kısıt fonksiyonu ve parametre bilgileri aşağıda verilmiştir.

Amaç Fonksiyonu:

$$\text{Minimize } f(\vec{x}) = (x_3 + 2)x_2x_1^2 \quad (\text{E.1})$$

Kısıt Fonksiyonları:

$$G_1(\vec{x}) = 1 - \frac{x_2^2x_3}{71785x_1^4} \leq 0 \quad (\text{E.2})$$

$$G_2(\vec{x}) = \frac{4x_2^2 - x_1x_2}{12566(x_2x_1^3 - x_1^4)} + \frac{1}{5108x_1^2} \leq 0 \quad (\text{E.3})$$

$$G_3(\vec{x}) = 1 - \frac{140.45x_1}{x_2^2x_3} \leq 0 \quad (\text{E.4})$$

$$G_4(\vec{x}) = \frac{x_1 + x_2}{1.5} - 1 \leq 0 \quad (\text{E.5})$$

Parametre Ayarları:

- $0.05 \leq x_1 \leq 2.0$
- $0.25 \leq x_2 \leq 1.3$
- $2.0 \leq x_3 \leq 15.0$

Kaynaklı Kiriş Tasarımı (Welded Beam Design):

Kaynaklı kiriş tasarımı kısıtlı mühendislik probleminin (Kaveh & Dadras, 2017) amaç fonksiyonu, kısıt fonksiyonu ve parametre bilgileri aşağıda verilmiştir.

Amaç Fonksiyonu:

$$\text{Minimize } f(\vec{x}) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14.0 + x_2) \quad (\text{E.6})$$

Kısıt Fonksiyonlar:

$$G_1(\vec{x}) = \tau(\vec{x}) - 13600 \leq 0 \quad (\text{E.7})$$

$$G_2(\vec{x}) = \sigma(\vec{x}) - 30000 \leq 0 \quad (\text{E.8})$$

$$G_3(\vec{x}) = \delta(\vec{x}) - 0.25 \leq 0 \quad (\text{E.9})$$

$$G_4(\vec{x}) = x_1 - x_4 \leq 0 \quad (\text{E.10})$$

$$G_5(\vec{x}) = 6000 - P_c(\vec{x}) \leq 0 \quad (\text{E.11})$$

$$G_6(\vec{x}) = 0.125 - x_1 \leq 0 \quad (\text{E.12})$$

$$G_7(\vec{x}) = 1.10471x_1^2 + 0.04811x_3x_4(14.0 + x_2) - 5.0 \leq 0 \quad (\text{E.13})$$

Parametre Ayarları:

- $0.1 \leq x_1, x_4 \leq 2$
- $0.1 \leq x_2, x_3 \leq 10.0$

$$\tau(\vec{x}) = \sqrt{(\tau')^2 + (\tau'')^2 + \frac{(x_2\tau'\tau'')}{\sqrt{0.25(x_2^2 + (x_1 + x_3)^2)}}} \quad (\text{E.14})$$

$$\tau' = \frac{6000}{\sqrt{2}x_1x_2} \quad (\text{E.15})$$

$$\sigma(\vec{x}) = \frac{504000}{x_3^2x_4} \quad (\text{E.16})$$

$$\delta(\vec{x}) = \frac{65856000}{(30 * 10^6)x_4x_3^3} \quad (\text{E.17})$$

$$\tau'' = \frac{6000(14 + 0.5x_2)\sqrt{0.25(x_2^2 + (x_1 + x_3)^2)}}{2 \left[0.707x_1x_2 \left(\frac{x_2^2}{12} + 0.25(x_1 + x_3)^2 \right) \right]} \quad (\text{E.18})$$

$$P_c(\vec{x}) = 64746.022(1 - 0.0282346x_3)x_3x_4^3 \quad (\text{E.19})$$

Dişli Sistemi Tasarımı (Gear Train Design):

Dişli sistemi tasarımı kısıtlı mühendislik probleminin (Lin, Zhang, & Xu, 2018) amaç fonksiyonu, kısıt fonksiyonu ve parametre bilgileri aşağıda verilmiştir.

Amaç Fonksiyonu:

$$\text{Minimize } f(\vec{x}) = \left(\frac{1}{6.931} - \frac{x_2x_3}{x_1x_4} \right)^2 \quad (\text{E.19})$$

Parametre Ayarları:

- $12 \leq x_i \leq 60 \quad i = 1,2,3,4$

Üç Çubuk Kafes Tasarımı (Three Bar Truss Design):

Üç çubuk kafes tasarımı kısıtlı mühendislik probleminin (Kumar, ve diğerleri, 2020) amaç fonksiyonu, kısıt fonksiyonu ve parametre bilgileri aşağıda verilmiştir.

Amaç Fonksiyonu:

$$\text{Minimize } f(\vec{x}) = (2\sqrt{2}x_1 + x_2) * l \quad (\text{E.20})$$

Kısıt Fonksiyonları:

$$G_1(\vec{x}) = \left(\frac{\sqrt{2}x_1 + x_2}{\sqrt{2}x_1^2 + 2x_1x_2} P \right) - \sigma \leq 0 \quad (\text{E.21})$$

$$G_2(\vec{x}) = \left(\frac{x_2}{\sqrt{2}x_1^2 + 2x_1x_2} P \right) - \sigma \leq 0 \quad (\text{E.22})$$

$$G_3(\vec{x}) = \left(\frac{1}{\sqrt{2}x_2 + x_1} P \right) - \sigma \leq 0 \quad (\text{E.23})$$

Parametre Ayarları:

- $0 \leq x_i \leq 1, i = 1,2$
- $l = 100 \text{ cm}$
- $P = 2 \text{ kN/cm}^2$
- $\sigma = 2 \text{ kN/cm}^2$

Konsol Kiriş Tasarımı (Cantilever Beam Design):

Konsol giriş tasarımı kısıtlı mühendislik probleminin (Gupta & Deep, 2019) amaç fonksiyonu, kısıt fonksiyonu ve parametre bilgileri aşağıda verilmiştir.

Amaç Fonksiyonu:

$$\text{Minimize } f(\vec{x}) = \left(\sum_{i=1}^5 x_i \right) * 0.6224 \quad (\text{E.24})$$

Kısıt Fonksiyonları:

$$G_1(\vec{x}) = \frac{61}{x_1^3} + \frac{27}{x_2^3} + \frac{19}{x_3^3} + \frac{7}{x_4^3} + \frac{1}{x_5^3} - 1 \leq 0 \quad (\text{E.25})$$

Parametre Ayarları:

- $0.01 \leq x_i \leq 100, i = 1, 2 \dots 5$

Eşanjör Ağ Tasarımı (Heat Exchanger Network Design (Case 1)):

Eşanjör ağ tasarımı kısıtlı mühendislik probleminin (Kumar, ve diğerleri, 2020) amaç fonksiyonu, kısıt fonksiyonu ve parametre bilgileri aşağıda verilmiştir.

Amaç Fonksiyon:

$$\text{Minimize } f(\vec{x}) = 35x_1^{0.6} + 35x_2^{0.6} \text{Minimize } f(\vec{x}) = \left(\sum_{i=1}^5 x_i \right) * 0.6224 \quad (\text{E.26})$$

Kısıt Fonksiyonları:

$$G_1(\vec{x}) = 200x_1x_4 - x_3 = 0 \quad (\text{E.27})$$

$$G_2(\vec{x}) = 200x_2x_6 - x_5 = 0 \quad (\text{E.28})$$

$$G_3(\vec{x}) = x_3 - 10000(x_7 - 100) = 0 \quad (\text{E.29})$$

$$G_4(\vec{x}) = x_5 - 10000(300 - x_7) = 0 \quad (\text{E.30})$$

$$G_5(\vec{x}) = x_3 - 10000(600 - x_8) = 0 \quad (\text{E.31})$$

$$G_6(\vec{x}) = x_5 - 10000(900 - x_9) = 0 \quad (\text{E.32})$$

$$G_7(\vec{x}) = x_4 \ln(x_8 - 100) - x_4 \ln(600 - x_7) - x_8 + x_7 + 500 = 0 \quad (\text{E.33})$$

$$G_8(\vec{x}) = x_6 \ln(x_9 - x_7) - x_6 \ln(600) - x_9 + x_7 + 600 = 0 \quad (\text{E.34})$$

Parametre Ayarları:

- $0 \leq x_1 \leq 10$
- $0 \leq x_2 \leq 200$
- $0 \leq x_3 \leq 100$
- $0 \leq x_4 \leq 200$
- $1000 \leq x_5 \leq 2000000$
- $0 \leq x_6 \leq 600$
- $100 \leq x_7 \leq 600$
- $100 \leq x_8 \leq 600$
- $100 \leq x_9 \leq 900$

Harmanlama Havuzlama Ayırma Problemi (Blending Pooling Separation Problem):

Harmanlama havuzlama ayırma kısıtlı mühendislik probleminin (Kumar, ve diğerleri, 2020) amaç fonksiyonu, kısıt fonksiyonu ve parametre bilgileri aşağıda verilmiştir.

Amaç Fonksiyonu:

$$\text{Minimize } f(\vec{x}) = 0.9979 + 0.00432x_5 + 0.01517x_{13} \quad (\text{E.35})$$

Kısıt Fonksiyonları:

$$G_1(\vec{x}) = \sum_{i=1}^4 x_i = 300 \quad (\text{E.36})$$

$$G_2(\vec{x}) = x_6 - \sum_{i=7}^8 x_i = 0 \quad (\text{E.37})$$

$$G_3(\vec{x}) = x_9 - \sum_{i=10}^{12} x_i = 0 \quad (\text{E.38})$$

$$G_4(\vec{x}) = x_{14} - \sum_{i=15}^{17} x_i = 0 \quad (\text{E.39})$$

$$G_5(\vec{x}) = x_{18} - \sum_{i=19}^{20} x_i = 0 \quad (\text{E.40})$$

$$G_6(\vec{x}) = x_5 x_{21} - x_6 x_{22} - x_9 x_{23} = 0 \quad (\text{E.41})$$

$$G_7(\vec{x}) = x_5 x_{24} - x_6 x_{25} - x_9 x_{26} = 0 \quad (\text{E.42})$$

$$G_8(\vec{x}) = x_5 x_{27} - x_6 x_{28} - x_9 x_{29} = 0 \quad (\text{E.43})$$

$$G_9(\vec{x}) = x_{13} x_{30} - x_{14} x_{31} - x_{18} x_{32} = 0 \quad (\text{E.44})$$

$$G_{10}(\vec{x}) = x_{13} x_{33} - x_{14} x_{34} - x_{18} x_{35} = 0 \quad (\text{E.45})$$

$$G_{11}(\vec{x}) = x_{13} x_{36} - x_{14} x_{37} - x_{18} x_{35} = 0 \quad (\text{E.46})$$

$$G_{12}(\vec{x}) = 0.333x_1 + x_{15} x_{31} - x_5 x_{21} = 0 \quad (\text{E.47})$$

$$G_{13}(\vec{x}) = 0.333x_1 + x_{15} x_{34} - x_5 x_{24} = 0 \quad (\text{E.48})$$

$$G_{14}(\vec{x}) = 0.333x_1 + x_{15} x_{37} - x_5 x_{27} = 0 \quad (\text{E.49})$$

$$G_{15}(\vec{x}) = 0.333x_2 + x_{10} x_{23} - x_{13} x_{30} = 0 \quad (\text{E.50})$$

$$G_{16}(\vec{x}) = 0.333x_2 + x_{10} x_{26} - x_{13} x_{33} = 0 \quad (\text{E.51})$$

$$G_{17}(\vec{x}) = 0.333x_2 + x_{10}x_{29} - x_{13}x_{36} = 0 \quad (\text{E.52})$$

$$G_{18}(\vec{x}) = 0.333x_3 + x_7x_{22} + x_{11}x_{23} + x_{16}x_{31} + x_{19}x_{32} = 30 \quad (\text{E.53})$$

$$G_{19}(\vec{x}) = 0.333x_3 + x_7x_{25} + x_{11}x_{26} + x_{16}x_{34} + x_{19}x_{35} = 50 \quad (\text{E.54})$$

$$G_{20}(\vec{x}) = 0.333x_3 + x_7x_{28} + x_{11}x_{29} + x_{16}x_{37} + x_{19}x_{38} = 30 \quad (\text{E.55})$$

$$G_{21}(\vec{x}) = x_{21} + x_{24} + x_{27} = 1 \quad (\text{E.56})$$

$$G_{22}(\vec{x}) = x_{22} + x_{25} + x_{28} = 1 \quad (\text{E.57})$$

$$G_{23}(\vec{x}) = x_{23} + x_{26} + x_{29} = 1 \quad (\text{E.58})$$

$$G_{24}(\vec{x}) = x_{30} + x_{33} + x_{36} = 1 \quad (\text{E.59})$$

$$G_{25}(\vec{x}) = x_{31} + x_{34} + x_{37} = 1 \quad (\text{E.60})$$

$$G_{26}(\vec{x}) = x_{30} + x_{33} + x_{36} = 1 \quad (\text{E.61})$$

$$G_{26}(\vec{x}) = x_{32} + x_{35} + x_{38} = 1 \quad (\text{E.62})$$

$$G_{27}(\vec{x}) = x_{25} = 0 \quad (\text{E.63})$$

$$G_{28}(\vec{x}) = x_{28} = 0 \quad (\text{E.64})$$

$$G_{29}(\vec{x}) = x_{23} = 0 \quad (\text{E.65})$$

$$G_{30}(\vec{x}) = x_{37} = 0 \quad (\text{E.66})$$

$$G_{31}(\vec{x}) = x_{32} = 0 \quad (\text{E.67})$$

$$G_{32}(\vec{x}) = x_{35} = 0 \quad (\text{E.68})$$

Parametre Ayarları:

- $0 \leq x_i \leq 90, i = 1, 3, 5, 6, 8, 9, 10, 13, 14, 16, 18, 20$
- $0 \leq x_i \leq 150, i = 2, 4, 7, 11, 12, 15, 17, 19$
- $0 \leq x_i \leq 1, i = 21, 23, 24, 25, 27, 28$
- $0 \leq x_i \leq 1.2, i = 22, 32, 34, 35, 37, 38$
- $0 \leq x_i \leq 0.5, i = 26, 29, 30, 31, 33, 36$

Propan, İzobütan, n-Bütan Keskin Olmayan Ayırma (Propane, Isobutane, n-Butane Nonsharp Separation):

Propan, izobütan, n-bütan kesin olmayan ayırma kısıtlı mühendislik probleminin (Kumar, ve diğerleri, 2020) amaç fonksiyonu, kısıt fonksiyonu ve parametre bilgileri aşağıda verilmiştir.

Amaç Fonksiyonu:

$$\text{Minimize } f(\vec{x}) = c_{11} + (c_{21} + c_{31}x_{24} + c_{41}x_{28} + c_{51}x_{33} + c_{61}x_{34})x_5 + \dots \\ c_{12} + (c_{22} + c_{32}x_{26} + c_{42}x_{31} + c_{52}x_{38} + c_{62}x_{39})x_{13} \quad (\text{E.69})$$

Kısıt Fonksiyonları:

$$G_1(\vec{x}) = \sum_{i=1}^4 x_i = 300 \quad (\text{E.70})$$

$$G_2(\vec{x}) = x_6 - \sum_{i=7}^8 x_i = 0 \quad (\text{E.71})$$

$$G_3(\vec{x}) = x_9 - \sum_{i=10}^{12} x_i = 0 \quad (\text{E.72})$$

$$G_4(\vec{x}) = x_{14} - \sum_{i=15}^{17} x_i = 0 \quad (\text{E.73})$$

$$G_5(\vec{x}) = x_{18} - \sum_{i=19}^{20} x_i = 0 \quad (\text{E.74})$$

$$G_6(\vec{x}) = x_6 x_{21} - x_{24} x_{25} = 0 \quad (\text{E.75})$$

$$G_7(\vec{x}) = x_{14} x_{22} - x_{26} x_{27} = 0 \quad (\text{E.76})$$

$$G_8(\vec{x}) = x_9 x_{23} - x_{28} x_{29} = 0 \quad (\text{E.77})$$

$$G_9(\vec{x}) = x_{18} x_{30} - x_{31} x_{32} = 0 \quad (\text{E.78})$$

$$G_{10}(\vec{x}) = x_{25} - x_5 x_{33} = 0 \quad (\text{E.79})$$

$$G_{12}(\vec{x}) = x_{35} - x_5 x_{36} = 0 \quad (\text{E.80})$$

$$G_{13}(\vec{x}) = x_{37} - x_{13} x_{38} = 0 \quad (\text{E.81})$$

$$G_{14}(\vec{x}) = x_{27} - x_{13} x_{39} = 0 \quad (\text{E.82})$$

$$G_{15}(\vec{x}) = x_{32} - x_{13} x_{40} = 0 \quad (\text{E.83})$$

$$G_{16}(\vec{x}) = x_{25} - x_6 x_{21} - x_9 x_{41} = 0 \quad (\text{E.84})$$

$$G_{17}(\vec{x}) = x_{29} - x_6 x_{42} - x_9 x_{23} = 0 \quad (\text{E.85})$$

$$G_{18}(\vec{x}) = x_{35} - x_6 x_{43} - x_9 x_{44} = 0 \quad (\text{E.86})$$

$$G_{19}(\vec{x}) = x_{37} - x_{14} x_{45} - x_{18} x_{46} = 0 \quad (\text{E.87})$$

$$G_{20}(\vec{x}) = x_{27} - x_{14} x_{22} - x_{18} x_{47} = 0 \quad (\text{E.88})$$

$$G_{21}(\vec{x}) = x_{32} - x_{14} x_{48} - x_{18} x_{30} = 0 \quad (\text{E.89})$$

$$G_{22}(\vec{x}) = 0.333x_1 + x_{15} x_{45} - x_{25} = 0 \quad (\text{E.90})$$

$$G_{23}(\vec{x}) = 0.333x_1 + x_{15} x_{22} - x_{29} = 0 \quad (\text{E.91})$$

$$G_{24}(\vec{x}) = 0.333x_1 + x_{15} x_{48} - x_{35} = 0 \quad (\text{E.92})$$

$$G_{25}(\vec{x}) = 0.333x_2 + x_{10} x_{41} - x_{37} = 0 \quad (\text{E.93})$$

$$G_{26}(\vec{x}) = 0.333x_2 + x_{10} x_{23} - x_{27} = 0 \quad (\text{E.94})$$

$$G_{27}(\vec{x}) = 0.333x_2 + x_{10} x_{44} - x_{32} = 0 \quad (\text{E.95})$$

$$G_{28}(\vec{x}) = 0.333x_3 + x_7 x_{21} + x_{11} x_{41} + x_{16} x_{45} + x_{19} x_{46} = 30 \quad (\text{E.96})$$

$$G_{29}(\vec{x}) = 0.333x_3 + x_7x_{42} + x_{11}x_{23} + x_{16}x_{22} + x_{19}x_{47} = 50 \quad (\text{E.97})$$

$$G_{30}(\vec{x}) = 0.333x_3 + x_7x_{43} + x_{11}x_{44} + x_{16}x_{48} + x_{19}x_{30} = 30 \quad (\text{E.98})$$

$$G_{31}(\vec{x}) = x_{33} + x_{34} + x_{36} = 1 \quad (\text{E.99})$$

$$G_{32}(\vec{x}) = x_{21} + x_{42} + x_{43} = 1 \quad (\text{E.100})$$

$$G_{33}(\vec{x}) = x_{41} + x_{23} + x_{44} = 1 \quad (\text{E.101})$$

$$G_{34}(\vec{x}) = x_{38} + x_{39} + x_{40} = 1 \quad (\text{E.102})$$

$$G_{35}(\vec{x}) = x_{45} + x_{22} + x_{48} = 1 \quad (\text{E.103})$$

$$G_{36}(\vec{x}) = x_{46} + x_{47} + x_{30} = 1 \quad (\text{E.104})$$

$$G_{37}(\vec{x}) = x_{43} = 0 \quad (\text{E.105})$$

$$G_{38}(\vec{x}) = x_{46} = 0 \quad (\text{E.106})$$

Parametre Ayarları:

- $0 \leq x_i \leq 150, i = 1, 2 \dots 20$
- $0 \leq x_i \leq 30, i = 25, 27, 29, 32, 35, 37$
- $0 \leq x_i \leq 1, i = 21, 22, 23, 30, 33, 34, 36, 38, 39, 40, 42, 43, 44, 45, 46, 47, 48$
- $0.85 \leq x_i \leq 1, i = 24, 26, 28, 31$

C parametre ayarları Ek Tablo 1'deki gibidir.

Ek Tablo 1. C parametre ayarları

C	$\dot{\mathbf{I}} = 1$	$\dot{\mathbf{I}} = 2$
C _{1i}	0.23947	0.75835
C _{2i}	-0.0139904	-0.0661588
C _{3i}	0.0093514	0.0338147
C _{4i}	0.0077308	0.0373349
C _{5i}	-0.0005719	0.0016371
C _{6i}	0.0042656	0.0288996

Alkilasyon Süreci Optimizasyonu (Alkylation Process Optimization):

Alkilasyon süreci kısıtlı mühendislik probleminin (Ravindran, Reklaitis, & Ragsdell, 2006) amaç fonksiyonu, kısıt fonksiyonu ve parametre bilgileri aşağıda verilmiştir.

Amaç Fonksiyonu:

$$\text{Minimize } f(\vec{x}) = 0.063x_4x_7 - 5.04x_1 - 0.035x_2 - 10x_3 - 3.36x_5 \quad (\text{E.107})$$

Kısıt Fonksiyonları:

$$G_1(\vec{x}) = \frac{x_4}{100x_1} - 112 - 13.167x_8 - 0.6667x_8^2 = 0 \quad (\text{E.108})$$

$$G_2(\vec{x}) = x_8 - \frac{x_5 + x_2}{x_1} = 0 \quad (\text{E.109})$$

$$G_3(\vec{x}) = x_4 - x_1 - x_5 + 0.22x_4 = 0 \quad (\text{E.110})$$

$$G_4(\vec{x}) = 10^3x_3 - x_4x_9\left(\frac{x_6}{98 - x_6}\right) = 0 \quad (\text{E.111})$$

$$G_5(\vec{x}) = x_9 - 35.82 - 0.222x_{10} = 0 \quad (\text{E.112})$$

$$G_6(\vec{x}) = x_{10} + 133 - 3x_7 = 0 \quad (\text{E.113})$$

$$G_7(\vec{x}) = x_7 - 86.35 - 1.098x_8 + 0.038x_8^2 + \left(\frac{1.3}{4}\right)(x_6 - 89) = 0 \quad (\text{E.114})$$

Parametre Ayarları:

- $0 \leq x_i \leq 2000, i = 1$
- $0 \leq x_i \leq 16000, i = 2$
- $0 \leq x_i \leq 120, i = 3$
- $0 \leq x_i \leq 5000, i = 4$
- $0 \leq x_i \leq 2000, i = 5$
- $85 \leq x_i \leq 93, i = 6$
- $90 \leq x_i \leq 95, i = 7$
- $3 \leq x_i \leq 12, i = 8$
- $1.2 \leq x_i \leq 4, i = 9$
- $145 \leq x_i \leq 162, i = 10$

Dört Kademeli Dişli Kutusu (Four Stage Gear Box):

Dört kademeli dişli kutusu kısıtlı mühendislik probleminin (Kumar, ve diğerleri, 2020) amaç fonksiyonu, kısıt fonksiyonu ve parametre bilgileri aşağıda verilmiştir.

Amaç Fonksiyonu:

$$\text{Minimize } f(\vec{x}) = \left(\frac{\pi}{1000}\right) \sum_{i=1}^4 \left(\frac{b_i c_i^2 (N_{pi}^2 + N_{gi}^2)}{(N_{pi} + N_{gi})^2} \right) \quad (\text{E.115})$$

Kısıt Fonksiyonları:

$$G_1(\vec{x}) = \left(\frac{366000}{\pi \omega_1} + \frac{2c_1 N_{p1}}{N_{p1} + N_{g1}} \right) \left(\frac{(N_{p1} + N_{g1})^2}{4b_1 c_1^2 N_{p1}} \right) - \frac{\sigma_N J_R}{0.0167 W K_0 K_m} \leq 0 \quad (\text{E.116})$$

$$= \left(\frac{366000 N_{g1}}{\pi \omega_1 N_{p1}} + \frac{2c_2 N_{p2}}{N_{p2} + N_{g2}} \right) \left(\frac{G_2(\vec{x})}{4b_2 c_2^2 N_{p2}} \right) - \frac{\sigma_N J_R}{0.0167 W K_0 K_m} \leq 0 \quad (\text{E.117})$$

$$= \left(\frac{366000 N_{g1} N_{g2}}{\pi \omega_1 N_{p1} N_{p2}} + \frac{2c_3 N_{p3}}{N_{p3} + N_{g3}} \right) \left(\frac{G_3(\vec{x})}{4b_3 c_3^2 N_{p3}} \right) - \frac{\sigma_N J_R}{0.0167 W K_0 K_m} \leq 0 \quad (\text{E.118})$$

$$= \left(\frac{366000N_{g1}N_{g2}N_{g3}}{\pi\omega_1N_{p1}N_{p2}N_{p3}} + \frac{2c_4N_{p4}}{N_{p4} + N_{g4}} \right) \frac{G_4(\vec{x})}{\left(\frac{(N_{p4} + N_{g4})^2}{4b_4c_4^2N_{p4}} \right)} - \frac{\sigma_N J_R}{0.0167WK_0K_m} \leq 0 \quad (\text{E.119})$$

$$= \left(\frac{366000}{\pi\omega_1} + \frac{2c_1N_{p1}}{N_{p1} + N_{g1}} \right) \frac{G_5(\vec{x})}{\left(\frac{(N_{p1} + N_{g1})^3}{4b_1c_1^2N_{g1}N_{p1}^2} \right)} - \left(\frac{\sigma_H}{C_p} \right)^2 \frac{\sin(\emptyset) \cos(\emptyset)}{0.0334WK_0K_m} \leq 0 \quad (\text{E.120})$$

$$= \left(\frac{366000N_{g1}}{\pi\omega_1N_{p1}} + \frac{2c_2N_{p2}}{N_{p2} + N_{g2}} \right) \frac{G_6(\vec{x})}{\left(\frac{(N_{p2} + N_{g2})^3}{4b_2c_2^2N_{g2}N_{p2}^2} \right)} - \left(\frac{\sigma_H}{C_p} \right)^2 \frac{\sin(\emptyset) \cos(\emptyset)}{0.0334WK_0K_m} \leq 0 \quad (\text{E.121})$$

$$= \left(\frac{366000N_{g1}N_{g2}}{\pi\omega_1N_{p1}N_{p2}} + \frac{2c_3N_{p3}}{N_{p3} + N_{g3}} \right) \frac{G_7(\vec{x})}{\left(\frac{(N_{p3} + N_{g3})^3}{4b_3c_3^2N_{g3}N_{p3}^2} \right)} - \left(\frac{\sigma_H}{C_p} \right)^2 \frac{\sin(\emptyset) \cos(\emptyset)}{0.0334WK_0K_m} \leq 0 \quad (\text{E.122})$$

$$= \left(\frac{366000N_{g1}N_{g2}N_{g3}}{\pi\omega_1N_{p1}N_{p2}N_{p3}} + \frac{2c_4N_{p4}}{N_{p4} + N_{g4}} \right) \frac{G_8(\vec{x})}{\left(\frac{(N_{p4} + N_{g4})^3}{4b_4c_4^2N_{g4}N_{p4}^2} \right)} - \left(\frac{\sigma_H}{C_p} \right)^2 \frac{\sin(\emptyset) \cos(\emptyset)}{0.0334WK_0K_m} \leq 0 \quad (\text{E.123})$$

$$= -N_{p1} \sqrt{\frac{\sin^2(\emptyset)}{4} - \frac{1}{N_{p1}} + \left(\frac{1}{N_{p1}} \right)^2} + N_{g1} \sqrt{\frac{\sin^2(\emptyset)}{4} + \frac{1}{N_{g1}} + \left(\frac{1}{N_{g1}} \right)^2} + \dots$$

$$\frac{\sin(\emptyset) (N_{p1} + N_{g1})}{2} + CR_{min}\pi \cos(\emptyset) \leq 0 \quad (\text{E.124})$$

$$= -N_{p2} \sqrt{\frac{\sin^2(\emptyset)}{4} - \frac{1}{N_{p2}} + \left(\frac{1}{N_{p2}} \right)^2} + N_{g2} \sqrt{\frac{\sin^2(\emptyset)}{4} + \frac{1}{N_{g2}} + \left(\frac{1}{N_{g2}} \right)^2} + \dots$$

$$\frac{\sin(\emptyset) (N_{p2} + N_{g2})}{2} + CR_{min}\pi \cos(\emptyset) \leq 0 \quad (\text{E.125})$$

$$= -N_{p3} \sqrt{\frac{\sin^2(\emptyset)}{4} - \frac{1}{N_{p3}} + \left(\frac{1}{N_{p3}} \right)^2} + N_{g3} \sqrt{\frac{\sin^2(\emptyset)}{4} + \frac{1}{N_{g3}} + \left(\frac{1}{N_{g3}} \right)^2} + \dots$$

$$\frac{\sin(\emptyset) (N_{p3} + N_{g3})}{2} + CR_{min}\pi \cos(\emptyset) \leq 0 \quad (\text{E.126})$$

$$= -N_{p4} \sqrt{\frac{\sin^2(\emptyset)}{4} - \frac{1}{N_{p4}} + \left(\frac{1}{N_{p4}} \right)^2} + N_{g4} \sqrt{\frac{\sin^2(\emptyset)}{4} + \frac{1}{N_{g4}} + \left(\frac{1}{N_{g4}} \right)^2} + \dots$$

$$\frac{\sin(\emptyset) (N_{p4} + N_{g4})}{2} + CR_{min}\pi \cos(\emptyset) \leq 0 \quad (\text{E.127})$$

$$G_{13}(\vec{x}) = d_{min} - \frac{2c_1 N_{p1}}{N_{p1} + N_{g1}} \leq 0 \quad (\text{E.128})$$

$$G_{14}(\vec{x}) = d_{min} - \frac{2c_2 N_{p2}}{N_{p2} + N_{g2}} \leq 0 \quad (\text{E.129})$$

$$G_{15}(\vec{x}) = d_{min} - \frac{2c_3 N_{p3}}{N_{p3} + N_{g3}} \leq 0 \quad (\text{E.130})$$

$$G_{16}(\vec{x}) = d_{min} - \frac{2c_4 N_{p4}}{N_{p4} + N_{g4}} \leq 0 \quad (\text{E.131})$$

$$G_{17}(\vec{x}) = d_{min} - \frac{2c_1 N_{g1}}{N_{p1} + N_{g1}} \leq 0 \quad (\text{E.132})$$

$$G_{18}(\vec{x}) = d_{min} - \frac{2c_2 N_{g2}}{N_{p2} + N_{g2}} \leq 0 \quad (\text{E.133})$$

$$G_{19}(\vec{x}) = d_{min} - \frac{2c_3 N_{g3}}{N_{p3} + N_{g3}} \leq 0 \quad (\text{E.134})$$

$$G_{20}(\vec{x}) = d_{min} - \frac{2c_4 N_{g4}}{N_{p4} + N_{g4}} \leq 0 \quad (\text{E.135})$$

$$G_{21}(\vec{x}) = x_{p1} + \left(\frac{(N_{p1} + 2)c_1}{N_{p1} + N_{g1}} \right) - L_{max} \leq 0 \quad (\text{E.136})$$

$$G_{22}(\vec{x}) = x_{g2} + \left(\frac{(N_{p2} + 2)c_2}{N_{p2} + N_{g2}} \right) - L_{max} \leq 0 \quad (\text{E.137})$$

$$G_{23}(\vec{x}) = x_{g3} + \left(\frac{(N_{p3} + 2)c_3}{N_{p3} + N_{g3}} \right) - L_{max} \leq 0 \quad (\text{E.138})$$

$$G_{24}(\vec{x}) = x_{g4} + \left(\frac{(N_{p4} + 2)c_4}{N_{p4} + N_{g4}} \right) - L_{max} \leq 0 \quad (\text{E.139})$$

$$G_{25}(\vec{x}) = \left(\frac{(N_{p1} + 2)c_1}{N_{p1} + N_{g1}} \right) - x_{p1} \leq 0 \quad (\text{E.140})$$

$$G_{26}(\vec{x}) = \left(\frac{(N_{p2} + 2)c_2}{N_{p2} + N_{g2}} \right) - x_{g2} \leq 0 \quad (\text{E.141})$$

$$G_{27}(\vec{x}) = \left(\frac{(N_{p3} + 2)c_3}{N_{p3} + N_{g3}} \right) - x_{g3} \leq 0 \quad (\text{E.142})$$

$$G_{28}(\vec{x}) = \left(\frac{(N_{p4} + 2)c_4}{N_{p4} + N_{g4}} \right) - x_{g4} \leq 0 \quad (\text{E.143})$$

$$G_{29}(\vec{x}) = y_{p1} + \left(\frac{(N_{p1} + 2)c_1}{N_{p1} + N_{g1}} \right) - L_{max} \leq 0 \quad (\text{E.144})$$

$$G_{30}(\vec{x}) = y_{g2} + \left(\frac{(N_{p2} + 2)c_2}{N_{p2} + N_{g2}} \right) - L_{max} \leq 0 \quad (\text{E.145})$$

$$G_{31}(\vec{x}) = y_{g3} + \left(\frac{(N_{p3} + 2)c_3}{N_{p3} + N_{g3}} \right) - L_{max} \leq 0 \quad (\text{E.146})$$

$$G_{32}(\vec{x}) = y_{g4} + \left(\frac{(N_{p4} + 2)c_4}{N_{p4} + N_{g4}} \right) - L_{max} \leq 0 \quad (\text{E.147})$$

$$G_{33}(\vec{x}) = \left(\frac{(N_{p1} + 2)c_1}{N_{p1} + N_{g1}} \right) - y_{p1} \leq 0 \quad (\text{E.148})$$

$$G_{34}(\vec{x}) = \left(\frac{(N_{p2} + 2)c_2}{N_{p2} + N_{g2}} \right) - y_{g1} \leq 0 \quad (\text{E.149})$$

$$G_{35}(\vec{x}) = \left(\frac{(N_{p3} + 2)c_3}{N_{p3} + N_{g3}} \right) - y_{g2} \leq 0 \quad (\text{E.150})$$

$$G_{36}(\vec{x}) = \left(\frac{(N_{p4} + 2)c_4}{N_{p4} + N_{g4}} \right) - y_{g3} \leq 0 \quad (\text{E.151})$$

$$G_{37}(\vec{x}) = x_{g1} + \left(\frac{(N_{g1} + 2)c_1}{N_{p1} + N_{g1}} \right) - L_{max} \leq 0 \quad (\text{E.152})$$

$$G_{38}(\vec{x}) = x_{g2} + \left(\frac{(N_{g2} + 2)c_2}{N_{p2} + N_{g2}} \right) - L_{max} \leq 0 \quad (\text{E.153})$$

$$G_{39}(\vec{x}) = x_{g3} + \left(\frac{(N_{g3} + 2)c_3}{N_{p3} + N_{g3}} \right) - L_{max} \leq 0 \quad (\text{E.154})$$

$$G_{40}(\vec{x}) = x_{g4} + \left(\frac{(N_{g4} + 2)c_4}{N_{p4} + N_{g4}} \right) - L_{max} \leq 0 \quad (\text{E.155})$$

$$G_{41}(\vec{x}) = \left(\frac{(N_{g1} + 2)c_1}{N_{p1} + N_{g1}} \right) - x_{g1} \leq 0 \quad (\text{E.156})$$

$$G_{42}(\vec{x}) = \left(\frac{(N_{g2} + 2)c_2}{N_{p2} + N_{g2}} \right) - x_{g2} \leq 0 \quad (\text{E.157})$$

$$G_{43}(\vec{x}) = \left(\frac{(N_{g3} + 2)c_3}{N_{p3} + N_{g3}} \right) - x_{g3} \leq 0 \quad (\text{E.158})$$

$$G_{44}(\vec{x}) = \left(\frac{(N_{g4} + 2)c_4}{N_{p4} + N_{g4}} \right) - x_{g4} \leq 0 \quad (\text{E.159})$$

$$G_{45}(\vec{x}) = y_{g1} + \left(\frac{(N_{g1} + 2)c_1}{N_{p1} + N_{g1}} \right) - L_{max} \leq 0 \quad (\text{E.160})$$

$$G_{46}(\vec{x}) = y_{g2} + \left(\frac{(N_{g2} + 2)c_2}{N_{p2} + N_{g2}} \right) - L_{max} \leq 0 \quad (\text{E.161})$$

$$G_{47}(\vec{x}) = y_{g3} + \left(\frac{(N_{g3} + 2)c_3}{N_{p3} + N_{g3}} \right) - L_{max} \leq 0 \quad (\text{E.162})$$

$$G_{48}(\vec{x}) = y_{g4} + \left(\frac{(N_{g4} + 2)c_4}{N_{p4} + N_{g4}} \right) - L_{max} \leq 0 \quad (\text{E.163})$$

$$G_{49}(\vec{x}) = \left(\frac{(N_{g1} + 2)c_1}{N_{p1} + N_{g1}} \right) - y_{g1} \leq 0 \quad (\text{E.164})$$

$$G_{50}(\vec{x}) = \left(\frac{(N_{g2} + 2)c_2}{N_{p2} + N_{g2}} \right) - y_{g2} \leq 0 \quad (\text{E.165})$$

$$G_{51}(\vec{x}) = \left(\frac{(N_{g3} + 2)c_3}{N_{p3} + N_{g3}} \right) - y_{g3} \leq 0 \quad (\text{E.166})$$

$$G_{52}(\vec{x}) = \left(\frac{(N_{g4} + 2)c_4}{N_{p4} + N_{g4}} \right) - y_{g4} \leq 0 \quad (\text{E.167})$$

$$= (b_1 - 8.255)(b_1 - 5.715)(b_1 - 12.70) \left(\frac{G_{53}(\vec{x})}{-N_{p1} + 0.945c_1 - N_{g1}} \right) (-1) \leq 0 \quad (\text{E.168})$$

$$= (b_2 - 8.255)(b_2 - 5.715)(b_2 - 12.70) \left(\frac{G_{54}(\vec{x})}{-N_{p2} + 0.945c_2 - N_{g2}} \right) (-1) \leq 0 \quad (\text{E.169})$$

$$= (b_3 - 8.255)(b_3 - 5.715)(b_3 - 12.70) \left(\frac{G_{55}(\vec{x})}{-N_{p3} + 0.945c_3 - N_{g3}} \right) (-1) \leq 0 \quad (\text{E.170})$$

$$= (b_4 - 8.255)(b_4 - 5.715)(b_4 - 12.70) \left(\frac{G_{56}(\vec{x})}{-N_{p4} + 0.945c_4 - N_{g4}} \right) (-1) \leq 0 \quad (\text{E.171})$$

$$= (b_1 - 8.255)(b_1 - 3.175)(b_1 - 12.70) \left(\frac{G_{57}(\vec{x})}{-N_{p1} + 0.646c_1 - N_{g1}} \right) \leq 0 \quad (\text{E.172})$$

$$= (b_2 - 8.255)(b_2 - 3.175)(b_2 - 12.70) \left(\frac{G_{58}(\vec{x})}{-N_{p2} + 0.646c_2 - N_{g2}} \right) \leq 0 \quad (\text{E.173})$$

$$= (b_3 - 8.255)(b_3 - 3.175)(b_3 - 12.70) \left(\frac{G_{59}(\vec{x})}{-N_{p3} + 0.646c_3 - N_{g3}} \right) \leq 0 \quad (\text{E.174})$$

$$= (b_4 - 8.255)(b_4 - 3.175)(b_4 - 12.70) \left(\frac{G_{60}(\vec{x})}{-N_{p4} + 0.646c_4 - N_{g4}} \right) \leq 0 \quad (\text{E.175})$$

$$= (b_1 - 5.715)(b_1 - 3.175)(b_1 - 12.70) \left(\frac{G_{61}(\vec{x})}{-N_{p1} + 0.504c_1 - N_{g1}} \right) \leq 0 \quad (\text{E.176})$$

$$= (b_2 - 5.715)(b_2 - 3.175)(b_2 - 12.70) \left(\frac{G_{62}(\vec{x})}{-N_{p2} + 0.504c_2 - N_{g2}} \right) \leq 0 \quad (\text{E.177})$$

$$= (b_3 - 5.715)(b_3 - 3.175)(b_3 - 12.70)(-N_{p3} + 0.504c_3 - N_{g3}) \leq 0 \quad (\text{E.178})$$

$$= (b_4 - 5.715)(b_4 - 3.175)(b_4 - 12.70)(-N_{p4} + 0.504c_4 - N_{g4}) \leq 0 \quad (\text{E.179})$$

$$= (b_1 - 5.715)(b_1 - 3.175)(b_1 - 8.255)(0c_1 - N_{g1} - N_{p1}) \leq 0 \quad (\text{E.180})$$

$$= (b_2 - 5.715)(b_2 - 3.175)(b_2 - 8.255)(0c_2 - N_{g2} - N_{p2}) \leq 0 \quad (\text{E.181})$$

$$= (b_3 - 5.715)(b_3 - 3.175)(b_3 - 8.255)(0c_3 - N_{g3} - N_{p3}) \leq 0 \quad (\text{E.182})$$

$$= (b_4 - 5.715)(b_4 - 3.175)(b_4 - 8.255)(0c_4 - N_{g4} - N_{p4}) \leq 0 \quad (\text{E.183})$$

$$= (b_1 - 8.255)(b_1 - 5.715)(b_1 - 12.70)(N_{g1} + N_{p1} - 1.812c_1)(-1) \leq 0 \quad (\text{E.184})$$

$$= (b_2 - 8.255)(b_2 - 5.715)(b_2 - 12.70)(N_{g2} + N_{p2} - 1.812c_2)(-1) \leq 0 \quad (\text{E.185})$$

$$= (b_3 - 8.255)(b_3 - 5.715)(b_3 - 12.70)(N_{g3} + N_{p3} - 1.812c_3)(-1) \leq 0 \quad (\text{E.186})$$

$$= (b_4 - 8.255)(b_4 - 5.715)(b_4 - 12.70)(N_{g4} + N_{p4} - 1.812c_4)(-1) \leq 0 \quad (\text{E.187})$$

$$= (b_1 - 8.255)(b_1 - 3.175)(b_1 - 12.70)(N_{g1} + N_{p1} - 0.945c_1) \leq 0 \quad (\text{E.188})$$

$$= (b_2 - 8.255)(b_2 - 3.175)(b_2 - 12.70)(N_{g2} + N_{p2} - 0.945c_2) \leq 0 \quad (\text{E.189})$$

$$= (b_3 - 8.255)(b_3 - 3.175)(b_3 - 12.70)(N_{g3} + N_{p3} - 0.945c_3) \leq 0 \quad (\text{E.190})$$

$$= (b_4 - 8.255)(b_4 - 3.175)(b_4 - 12.70)(N_{g4} + N_{p4} - 0.945c_4) \leq 0 \quad (\text{E.191})$$

$$= (b_1 - 5.715)(b_1 - 3.175)(b_1 - 12.70)(N_{g1} + N_{p1} - 0.646c_1)(-1) \leq 0 \quad (\text{E.192})$$

$$= (b_2 - 5.715)(b_2 - 3.175)(b_2 - 12.70)(N_{g2} + N_{p2} - 0.646c_2)(-1) \leq 0 \quad (\text{E.193})$$

$$= (b_3 - 5.715)(b_3 - 3.175)(b_3 - 12.70)(N_{g3} + N_{p3} - 0.646c_3)(-1) \leq 0 \quad (\text{E.194})$$

$$= (b_4 - 5.715)(b_4 - 3.175)(b_4 - 12.70)(N_{g4} + N_{p4} - 0.646c_4)(-1) \leq 0 \quad (\text{E.195})$$

$$= (b_1 - 5.715)(b_1 - 3.175)(b_1 - 8.255)(N_{g1} + N_{p1} - 0.504c_1) \leq 0 \quad (\text{E.196})$$

$$= (b_2 - 5.715)(b_2 - 3.175)(b_2 - 8.255)(N_{g2} + N_{p2} - 0.504c_2) \leq 0 \quad (\text{E.197})$$

$$= (b_3 - 5.715)(b_3 - 3.175)(b_3 - 8.255)(N_{g3} + N_{p3} - 0.504c_3) \leq 0 \quad (\text{E.198})$$

$$= (b_4 - 5.715)(b_4 - 3.175)(b_4 - 8.255)(N_{g4} + N_{p4} - 0.504c_4) \leq 0 \quad (\text{E.199})$$

$$G_{85}(\vec{x}) = w_{min} - \left(\frac{w_1(N_{p1}N_{p2}N_{p3}N_{p4})}{(N_{g1}N_{g2}N_{g3}N_{g4})} \right) \leq 0 \quad (\text{E.200})$$

$$G_{86}(\vec{x}) = \left(\frac{w_1(N_{p1}N_{p2}N_{p3}N_{p4})}{(N_{g1}N_{g2}N_{g3}N_{g4})} \right) - w_{max} \leq 0 \quad (\text{E.201})$$

Parametre Ayarları:

- $\vec{x} = \{N_{p1}, N_{g1}, N_{p2}, N_{g2}, \dots, b_1, b_2, \dots, x_{p1}, x_{g1}, x_{g2}, \dots, y_{p1}, y_{g1}, y_{g2}, \dots, y_{g4}\}$
- $c_1 = \sqrt{(y_{g1} - y_{p1})^2 + (x_{g1} - x_{p1})^2}$
- $c_2 = \sqrt{(y_{g2} - y_{p2})^2 + (x_{g2} - x_{p2})^2}$
- $c_3 = \sqrt{(y_{g3} - y_{p3})^2 + (x_{g3} - x_{p3})^2}$

- $c_4 = \sqrt{(y_{g4} - y_{p4})^2 + (x_{g4} - x_{p4})^2}$
- $K_o = 1.5$
- $d_{min} = 25$
- $J_R = 0.2$
- $\emptyset = 120^\circ$
- $W = 55.9$
- $K_M = 1.6$
- $CR_{min} = 1.4$
- $L_{max} = 127$
- $C_p = 464$
- $\sigma_H = 3290$
- $w_{max} = 255$
- $w_1 = 5000$
- $\sigma_N = 2090$
- $w_{min} = 245$
- $b_i \in \{3.175, 12.7, 8.255, 5.715\}, i = 1, 2, 3, 4$
- $y_{p1}, x_{p1}, y_{gi}, x_{gi} \in \dots$
- $\{12.7, 38.1, 25.4, 50.8, 76.2, 63.5, 88.9, 114.3, 101.6\}, i = 1, 2, 3, 4$
- $7 \leq N_{gi}, N_{pi} \leq 76 \in integer, i = 1, 2, 3, 4$

ÖZGEÇMİŞ

Mehmet KATI, İlköğrenimini İhsan Dikmen 3 İlköğretim Okulu'nda bitirdikten sonra lise eğitimini 2015 yılında Bursa Mimar Sinan Mesleki ve Teknik Anadolu Lisesi'nde tamamladı. 2016-2017 eğitim yılında Karadeniz Teknik Üniversitesi Of Teknoloji Fakültesi'nde Yazılım Mühendisliği bölümünde lisans eğitimine başladı. 2020 yılında bu bölümden mezun oldu ve HAVELSAN da yazılım mühendisi olarak çalışmaya başladı. Aynı yıl Karadeniz Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Yazılım Mühendisliği Anabilim dalında tezli yüksek lisans programına başladı. Yüksek lisans programı sürecinde dokuz adet akademik çalışmaya katkıda bulundu. Bu çalışmalar APA formatında aşağıda verilmiştir;

- Kahraman, H. T., Hassan, M. H., Kati, M., Tostado-Véliz, M., Duman, S., & Kamel, S. (2023). Dynamic-fitness-distance-balance stochastic fractal search (dFDB-SFS algorithm): an effective metaheuristic for global optimization and accurate photovoltaic modeling. *Soft Computing*, 1-28.
- Kahraman, H. T., Kati, M., Aras, S., & Taşci, D. A. (2023). Development of the Natural Survivor Method (NSM) for designing an updating mechanism in metaheuristic search algorithms. *Engineering Applications of Artificial Intelligence*, 122, 106121.
- Tasci, D. A., Kahraman, H. T., Kati, M., & Yilmaz, C. (2023). Improved Gradient-Based Optimizer with Dynamic Fitness Distance Balance for Global Optimization Problems. *Smart Applications with Advanced Machine Learning and Human-Centred Problem Design*, 1, 247.
- Duman, S., Kahraman, H. T., & Kati, M. (2023). Economical operation of modern power grids incorporating uncertainties of renewable energy sources and load demand using the adaptive fitness-distance balance-based stochastic fractal search algorithm. *Engineering Applications of Artificial Intelligence*, 117, 105501.
- Kahraman, H. T., Akbel, M., Duman, S., Kati, M., & Sayan, H. H. (2022). Unified space approach-based Dynamic Switched Crowding (DSC): A new method for designing Pareto-based multi/many-objective algorithms. *Swarm and Evolutionary Computation*, 75, 101196.

- Sonmez, Y., Duman, S., Kahraman, H. T., Kati, M., Aras, S., & Guvenc, U. (2022). Fitness-distance balance based artificial ecosystem optimisation to solve transient stability constrained optimal power flow problem. *Journal of Experimental & Theoretical Artificial Intelligence*, 1-40.
- Duman, S., Kahraman, H. T., Sonmez, Y., Guvenc, U., Kati, M., & Aras, S. (2022). A powerful meta-heuristic search algorithm for solving global optimization and real-world solar photovoltaic parameter estimation problems. *Engineering Applications of Artificial Intelligence*, 111, 104763.
- Kahraman, H. T., Bakir, H., Duman, S., Kati, M., Aras, S., & Guvenc, U. (2022). Dynamic FDB selection method and its application: modeling and optimizing of directional overcurrent relays coordination. *Applied Intelligence*, 1-36.
- Guvenc, U., Duman, S., Kahraman, H. T., Aras, S., & Kati, M. (2021). Fitness–Distance Balance based adaptive guided differential evolution algorithm for security-constrained optimal power flow problem incorporating renewable energy sources. *Applied Soft Computing*, 108, 107421.