

**T.C.  
AKDENİZ ÜNİVERSİTESİ**



**YAPISAL HASAR TESPİTİNDE ELEKTRİKLİ YILAN BALIĞI  
OPTİMİZASYON ALGORİTMASININ UYGULANMASI**

**Aleyna ZIVALIOĞLU**

**FEN BİLİMLERİ ENSTİTÜSÜ**

**İNŞAAT MÜHENDİSLİĞİ**

**ANABİLİM DALI**

**YÜKSEK LİSANS TEZİ**

**HAZİRAN 2025**

**ANTALYA**

**T.C.  
AKDENİZ ÜNİVERSİTESİ**



**YAPISAL HASAR TESPİTİNDE ELEKTRİKLİ YILAN BALIĞI  
OPTİMİZASYON ALGORİTMASININ UYGULANMASI**

**Aleyna ZIVALIOĞLU**

**FEN BİLİMLERİ ENSTİTÜSÜ**

**İNŞAAT MÜHENDİSLİĞİ**

**ANABİLİM DALI**

**YÜKSEK LİSANS TEZİ**

**HAZİRAN 2025**

**ANTALYA**

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

**YAPISAL HASAR TESPİTİNDE ELEKTRİKLİ YILAN BALIĞI**  
**OPTİMİZASYON ALGORİTMASININ UYGULANMASI**

**Aleyna ZIVALIOĞLU**  
**İNŞAAT MÜHENDİSLİĞİ**  
**ANABİLİM DALI**  
**YÜKSEK LİSANS TEZİ**

Bu tez 18/07/2025 tarihinde jüri tarafından Oybirliği / Oyçokluğu ile kabul edilmiştir.

Prof. Dr. İbrahim AYDOĞDU (Danışman)

Dr. Öğretim Üyesi Engin EMSEN

Dr. Öğretim Üyesi Seda TURAN

## ÖZET

### YAPISAL HASAR TESPİTİNDE ELEKTRİKLİ YILAN BALIĞI OPTİMİZASYON ALGORİTMASININ UYGULANMASI

Aleyna ZIVALIOĞLU

Yüksek Lisans Tezi, İnşaat Mühendisliği Anabilim Dalı

Danışman: Prof. Dr. İbrahim AYDOĞDU

Haziran 2025; 74 sayfa

Bu tez çalışmasında, yapıların titreşim tabanlı dinamik davranışları kullanılarak hasar tespiti yapılmasını sağlayan bir optimizasyon yaklaşımı geliştirilmiştir. Çalışmanın temel amacı, yapısal hasar tespitine yönelik meta-sezgisel yöntemlerden biri olan ve doğadan esinlenen Elektrikli Yılan Balığı Avlanma Optimizasyonu (EEFO) algoritmasının etkinliğini araştırmak ve bu algoritmayı yaygın kullanılan Harmony Search (HS) ve Yapay Arı Kolonisi (ABC) algoritmalarıyla karşılaştırmaktır. Hasar tespiti problemi, ölçülen ve model verileri arasındaki farkı minimize eden bir optimizasyon problemi olarak formüle edilmiştir. Bu amaçla, yapıların doğal frekansları ve mod şekillerine dayalı bir amaç fonksiyonu tanımlanmış, çözüm arayışı EEFO algoritmasıyla yürütülmüştür. Geliştirilen yöntem, MATLAB ortamında programlanmış ve analizler hem 10 elemanlı düzlemsel kafes hem de 72 elemanlı uzaysal kafes yapılar üzerinde gerçekleştirilmiştir. Ayrıca, farklı popülasyon büyüklüklerinde (20, 50, 75 ve 100) algoritmalar test edilerek performansları karşılaştırılmıştır. Elde edilen sonuçların güvenilirliğini sağlamak amacıyla SAP2000 yazılımıyla doğrulamalar yapılmıştır. Sonuçlar, EEFO algoritmasının yalnızca daha büyük popülasyon boyutlarında değil, aynı zamanda daha küçük popülasyonlarda da diğer algoritmalara kıyasla daha yüksek doğruluk ve sağlamlık sağladığını göstermektedir; burada hasar lokalizasyonu ve tahmin doğruluğu açısından HS ve ABC'yi hala geride bırakmıştır. EEFO, çözüm alanının daha dengeli bir keşfini gösterir, daha düşük hata oranları ve sınırlı popülasyon koşullarında bile daha tutarlı tahminler üretir. Bu bağlamda, EEFO yapısal hasar tespit sorunları için umut verici ve etkili bir alternatif olarak ortaya çıkmaktadır.

**ANAHTAR KELİMELE:** Elektrikli Yılan Balığı Avlanma Optimizasyon Algoritması, MATLAB, Modal Analiz, Modal Güvence Kriteri, Yapısal Hasar Tespiti, SAP2000

#### JÜRİ:

Prof. Dr. İbrahim AYDOĞDU

Dr. Öğretim Üyesi Engin EMSEN

Dr. Öğretim Üyesi Seda TURAN

## **ABSTRACT**

### **APPLICATION OF ELECTRIC EEL FORAGING OPTIMIZATION ALGORITHM IN STRUCTURAL DAMAGE DETECTION**

**Aleyna ZIVALIOĞLU**

**MSc Thesis in Civil Engineering**

**Supervisor: Prof. Dr. İbrahim AYDOĞDU**

**June 2025; 74 pages**

In this thesis, an optimization-based approach was developed for detecting structural damage by utilizing the vibration-based dynamic behavior of structures. The primary objective of the study is to investigate the effectiveness of the Electric Eel Foraging Optimization (EEFO) algorithm, a nature-inspired metaheuristic method, in the context of structural damage detection, and to compare its performance with widely used algorithms such as Harmony Search (HS) and Artificial Bee Colony (ABC). The study delves into the formulation of the damage detection problem as an optimization task, aiming to minimize the discrepancy between measured and modeled structural data. To this end, an objective function based on natural frequencies and mode shapes was defined and solved using the EEFO algorithm. The method was implemented in MATLAB, and analyzes were conducted on both a 10-element planar truss and a 72-element spatial truss system. Furthermore, the algorithms were tested under varying population sizes (20, 50, 75, and 100) to evaluate their sensitivity to population parameters. To validate the accuracy of the findings, comparative analyzes were performed using the SAP2000 software. The results demonstrate that the EEFO algorithm delivers higher accuracy and robustness compared to the other algorithms not only in larger population sizes, but also under smaller populations, where it still outperformed HS and ABC in terms of damage localization and estimation accuracy. EEFO shows a more balanced exploration of the solution space, yielding lower error rates and more consistent predictions even in limited population conditions. In this context, EEFO emerges as a promising and effective alternative for structural damage detection problems.

**KEYWORDS:** Electric Eel Foraging Optimization Algorithm, MATLAB, Modal Analysis, Modal Assurance Criterion, Structural Damage Detection, SAP2000

#### **COMMITTEE:**

Prof. Dr. İbrahim AYDOĞDU

Asst. Prof. Dr. Engin EMSEN

Asst. Prof. Dr. Seda TURAN

## ÖNSÖZ

Bu tez çalışması, yüksek lisans eğitimim süresince edindiğim bilgi ve deneyimlerin bir ürünü olup, yapıların hasar tespiti konusunda mevcut problemlere çözüm üretmeyi amaçlamaktadır. Yapıların güvenliği, özellikle deprem riski taşıyan bölgelerde yaşayan toplumlar için hayati önem taşımaktadır. Bu alandaki akademik çalışmaların yalnızca bilimsel bilgiye katkı sağlamakla kalmayıp, aynı zamanda insan hayatını doğrudan etkileyen sonuçlar doğurması, bu konuyu seçmemde belirleyici rol oynamıştır.

Tez sürecim boyunca zaman zaman çeşitli zorluklarla karşılaştım; ancak bu zorluklar, bana yalnızca akademik değil, kişisel olarak da direnç kazandırdı. Bilimsel bir çalışmayı tutarlılıkla sürdürebilmenin, sadece bilgi değil aynı zamanda kararlılık gerektirdiğini bu süreçte daha iyi anladım. Bu çalışmanın gelecekte yapılacak araştırmalara katkı sağlaması, yeni yöntemlerin geliştirilmesine ışık tutması en büyük dileğimdir.

Bu sürecin her aşamasında bana yol gösteren, bilgi ve tecrübeleriyle katkıda bulunan değerli tez danışmanım Prof. Dr. İbrahim Aydoğdu'na en içten teşekkürlerimi sunarım. Tez sürecinde teknik konulardaki desteği ve yapıcı yönlendirmeleriyle katkı sağlayan Araştırma Görevlisi Onur Altay'a ayrıca teşekkür ederim. Çalışma boyunca bana her zaman destek olan, yaşadığım her zorlukta beni motive eden aileme, arkadaşlarıma ve eğitim hayatım süresince emeği geçen tüm hocalarıma şükranlarımı sunarım.

## İÇİNDEKİLER

|                                                                                                |     |
|------------------------------------------------------------------------------------------------|-----|
| ÖZET.....                                                                                      | i   |
| ABSTRACT.....                                                                                  | ii  |
| ÖNSÖZ .....                                                                                    | iii |
| AKADEMİK BEYAN .....                                                                           | vi  |
| SİMGELER VE KISALTMALAR.....                                                                   | vii |
| ŞEKİLLER DİZİNİ.....                                                                           | xi  |
| ÇİZELGELER DİZİNİ .....                                                                        | xii |
| 1. GİRİŞ .....                                                                                 | 1   |
| 2. KAYNAK TARAMASI.....                                                                        | 3   |
| 2.1. Yapısal Hasar Tespitinde Meta-sezgisel Yöntemlerin Kullanılması.....                      | 3   |
| 2.2. EEFO'nun Çeşitli Problemlerdeki Performansının İncelenmesi .....                          | 17  |
| 2.3. Literatürdeki Eğilimler ve Özgünlük Değerlendirmesi.....                                  | 19  |
| 3. MATERYAL VE METOT.....                                                                      | 20  |
| 3.1. Model Tabanlı Yapısal Analiz .....                                                        | 20  |
| 3.1.1. Teori.....                                                                              | 20  |
| 3.1.2. Yapısal hasar tespitinde kullanılan test örneğinin MATLAB dilinde programlanması .....  | 22  |
| 3.1.3. Test Örneğindeki Düğüm Noktalarının Serbestliklerinin Belirlenmesi .....                | 22  |
| 3.1.4. Yapı Elemanlarının Kütle ve Rijitlik Matrislerinin Oluşturulması .....                  | 24  |
| 3.1.5. Modelin Frekans ve Mod Şekillerinin Elde Edilmesi .....                                 | 25  |
| 3.2. Yapısal Hasar Tespiti İçin Oluşturulan Optimizasyon Probleminin Matematiksel Modeli ..... | 26  |
| 3.3. Elektrikli Yılan Balığı Avlanma Optimizasyon Algoritması .....                            | 27  |
| 3.3.1. Teori.....                                                                              | 27  |
| 3.3.2. Etkileşim Aşaması .....                                                                 | 27  |
| 3.3.3. Dinlenme Aşaması.....                                                                   | 29  |
| 3.3.4. Avlama Aşaması.....                                                                     | 32  |
| 3.3.5. Göç Aşaması.....                                                                        | 33  |
| 3.3.6. Keşiften Sömürüye Geçiş .....                                                           | 35  |
| 3.3.7. Yöntemin Algoritması .....                                                              | 36  |

|                                                       |    |
|-------------------------------------------------------|----|
| 4. BULGULAR VE TARTIŞMA .....                         | 38 |
| 4.1. Sap2000 ve MATLAB Doğrulaması.....               | 38 |
| 4.1.1. 7 Elemanlı Düzlemsel Kafes Doğrulaması.....    | 38 |
| 4.1.2. 10 Elemanlı Düzlemsel Kafes Doğrulaması.....   | 39 |
| 4.1.3. 72 Elemanlı Uzaysal Kafes Doğrulaması .....    | 40 |
| 4.2. Hassaslık Analizleri.....                        | 43 |
| 4.2.1. 10 Elemanlı Düzlemsel Kafes .....              | 43 |
| 4.2.2. 72 Elemanlı Uzaysal Kafes.....                 | 54 |
| 4.3. İstatistiksel Sonuçların Karşılaştırılması ..... | 60 |
| 5. SONUÇLAR .....                                     | 67 |
| 6. KAYNAKLAR .....                                    | 70 |
| ÖZGEÇMİŞ .....                                        |    |

## AKADEMİK BEYAN

Yüksek Lisans Tezi olarak sunduğum “Yapısal Hasar Tespitinde Elektrikli Yılan Balığı Optimizasyon Algoritmasının Uygulanması” adlı bu çalışmanın, akademik kurallar ve etik değerlere uygun olarak yazıldığını belirtir, bu tez çalışmasında bana ait olmayan tüm bilgilerin kaynağını gösterdiğimi beyan ederim.

18/07/2025

Aleyna Zıvalıoğlu

İmzası

## SİMGELER VE KISALTMALAR

### Simgeler

|    |                          |
|----|--------------------------|
| m  | : metre                  |
| kN | : kilonewton             |
| %  | : yüzde                  |
| Hz | : Hertz (frekans birimi) |
| A  | : Eleman kesit alanı     |
| E  | : Elastisite modülü      |

### Kısaltmalar

|       |                                            |
|-------|--------------------------------------------|
| 2D    | : İki boyutlu (Two-Dimensional)            |
| 3D    | : Üç boyutlu (Three-Dimensional)           |
| ABC   | : Yapay Arı Kolonisi                       |
| ACO   | : Karınca Kolonisi Optimizasyonu           |
| ALO   | : Karınca Aslanı Optimizasyonu             |
| BA    | : Yarasa Algoritması                       |
| BS    | : Sınır Stratejisi                         |
| CFNN  | : Basamaklı İleri Beslemeli Sinir Ağı      |
| CGO   | : Kaos Oyun Optimizasyonu                  |
| ChOA  | : Şempanze Optimizasyon Algoritması        |
| CSS   | : Şarjlı Sistem Arama                      |
| DEA   | : Diferansiyel Evrim Algoritması           |
| DG    | : Dağıtık Enerji Kaynakları                |
| DILI  | : Sapma Etki Hattı Göstergesi              |
| DIMSE | : Modal Gerinim Enerjisi Hasar İndeksi     |
| DOF   | : Serbestlik Derecesi (Degrees of Freedom) |
| EA    | : Evrimsel Algoritmalar                    |

|          |   |                                                                 |
|----------|---|-----------------------------------------------------------------|
| EBA      | : | Geliştirilmiş Yarasa Optimizasyon Algoritması                   |
| ECBO     | : | Geliştirilmiş Çarpışan Cisimler Optimizasyonu                   |
| EEFO     | : | Elektrikli Yılan Balığı Avlanma Optimizasyonu                   |
| EHVPSO   | : | Geliştirilmiş Parçacık Sürü Optimizasyonu                       |
| ESA      | : | Ekolokasyon Arama Algoritması                                   |
| FEM      | : | Sonlu Elemanlar Yöntemi (Finite Element Method)                 |
| FOA      | : | Meyve Sinekleri Optimizasyonu Algoritmasından                   |
| FRF      | : | Frekans Tepkisi Fonksiyonu                                      |
| FRFSEDR  | : | Frekans Tepki Fonksiyonu Gerinim Enerjisi Dağılım Oranı         |
| FSEDR    | : | Esneklik Gerinim Enerjisi Hasar Oranı                           |
| GA       | : | Genetik Algoritma                                               |
| GMDH     | : | Veri İşleme Grup Yöntemi                                        |
| GOA      | : | Çekirge Optimizasyon Algoritması                                |
| GRO      | : | Altın Hız Optimizasyon                                          |
| GWO      | : | Gri Kurt Optimizasyonu                                          |
| HESOS-SA | : | Geliştirilmiş Simbiyotik Organizmalar Arama-Benzetilmiş Tavlama |
| HHO      | : | Harris Şahin Optimizasyonu                                      |
| HPSO     | : | Hibrit Parçacık Sürü Optimizasyonu                              |
| HS       | : | Harmoni Arama                                                   |
| IDI      | : | Düzensizlik Tespit İndeksi                                      |
| IEPSO    | : | Güçlendirilmiş Parçacık Sürü Optimizasyonu                      |
| IGMM     | : | İdeal Gaz Moleküler Hareketi                                    |
| IP-NMA   | : | İyileştirilmiş Pincus-Nelder-Mead Optimizasyon Algoritması      |
| IRR-GA   | : | Gizli Yedekli Temsil Genetik Algoritması                        |
| IRSA     | : | Geliştirilmiş Sürünge Arama Algoritmasını                       |
| LA       | : | Öğrenme Otomatlarının                                           |

|        |   |                                                       |
|--------|---|-------------------------------------------------------|
| LS-SVM | : | En Küçük Kare Destek Vektör Makineleri                |
| LSTM   | : | Kuantum Uzun Kısa Süreli Bellek                       |
| LVRT   | : | Düşük Voltajda Kalma                                  |
| LWOA   | : | Geliştirilmiş Balina Optimizasyonu                    |
| MAC    | : | Modal Assurance Criterion                             |
| MDBA   | : | Modifiye Edilmiş Yönlü Yarasalar Algoritması          |
| MEEFO  | : | Modifiye Elektrikli Yılan Balığı Avlanma Optimizasyon |
| MFO    | : | Güve-Alevi Optimizasyonu                              |
| MFOA   | : | Çoklu-Sürü Meyve Sinekleri Optimizasyonu Algoritması  |
| MOPSO  | : | Çok Hedefli Parçacık Sürüsü Optimizasyonu             |
| MPPT   | : | Maksimum Güç Noktası İzleme                           |
| MSE    | : | Modal Gerinim Enerjisi                                |
| MSEBI  | : | Modal Gerinim Enerjisi Tabanlı İndeks                 |
| MTMAC  | : | Değiştirilmiş Toplam Modal Güvence Kriteri            |
| MVO    | : | Multiverse Optimizer                                  |
| NFVAC  | : | Doğal Frekans Vektör Güvence Kriterinin               |
| NSGA   | : | Hakim Olmayan Sıralama Genetik Algoritması            |
| OD     | : | Ortogonal Çözümleme                                   |
| PBDD   | : | Olasılığa Dayalı Hasar Tespiti                        |
| PCC    | : | Ortak bağlantı noktası                                |
| PI     | : | Oransal-İntegral                                      |
| P-NMA  | : | Pincus-Nelder-Mead Optimizasyon Algoritması           |
| PSO    | : | Parçacık Sürüsü Optimizasyonu                         |
| PV     | : | Fotovoltaik                                           |
| QLSTM  | : | Kuantum Uzun Kısa Süreli Bellek                       |
| RDN    | : | Radyal Dağıtım Şebekeleri                             |

|        |   |                                              |
|--------|---|----------------------------------------------|
| RFBDI  | : | Kalıntı Kuvvet Bazlı Hasar İndeksi           |
| RFV    | : | Kalıntı Kuvvet Vektörü                       |
| SA     | : | Simüle Edilmiş Tavlama                       |
| SDI    | : | Yapısal Hasar Tanımlama                      |
| SOS    | : | Simbiyotik Organizmalar Araması              |
| SSA    | : | Salp Sürü Algoritması                        |
| SSOA   | : | Karıştırılmış Çoban Optimizasyon Algoritması |
| SVD    | : | Tekil değer Ayırıştırma                      |
| TLBO   | : | Öğretim-Öğrenme Tabanlı Optimizasyonu        |
| TLCO   | : | Termit Yaşam Döngüsü Optimizasyon            |
| VPSA   | : | Görünür Parçacık Serisi Arama Algoritması    |
| W-ChOA | : | Balina- Şempanze Optimizasyon Algoritması    |
| WOA    | : | Balina Optimizasyon Algoritması              |
| YBS    | : | Yapay Bağışıklık Sistemi                     |

## ŞEKİLLER DİZİNİ

|                                                                                                                                               |    |
|-----------------------------------------------------------------------------------------------------------------------------------------------|----|
| Şekil 3.1. Örnek sistemin sonlu elemanlar modeli .....                                                                                        | 22 |
| Şekil 3.2. Dinlenme alanının belirlenmesi .....                                                                                               | 30 |
| Şekil 3.3. Kıvrılma hareketi gösterimi .....                                                                                                  | 33 |
| Şekil 3.4. Göç davranışının gösterimi .....                                                                                                   | 34 |
| Şekil 3.5. EEFO algoritma şeması .....                                                                                                        | 37 |
| Şekil 4.1. 7 elemanlı düzlemsel kafesin SAP2000 modeli .....                                                                                  | 38 |
| Şekil 4.2. 10 elemanlı düzlemsel kafesin SAP2000 modeli .....                                                                                 | 39 |
| Şekil 4.3. 72 elemanlı uzaysal kafesin SAP2000 modeli .....                                                                                   | 41 |
| Şekil 4.4. Popülasyon 20 – 10 elemanlı kafesin fonksiyon değeri değişimi.....                                                                 | 44 |
| Şekil 4.5. Popülasyon 50 – 10 elemanlı kafesin fonksiyon değeri değişimi.....                                                                 | 44 |
| Şekil 4.6. Popülasyon 75 – 10 elemanlı kafesin fonksiyon değeri değişimi.....                                                                 | 45 |
| Şekil 4.7. Popülasyon 100 – 10 elemanlı kafesin fonksiyon değeri değişimi.....                                                                | 46 |
| Şekil 4.8. Popülasyon 20 – 10 elemanlı kafes örneği için algoritmaların hasar oranı tahminleri ve gerçek değerlerin karşılaştırılması .....   | 46 |
| Şekil 4.9. Popülasyon 50 - 10 elemanlı kafes örneği için algoritmaların hasar oranı tahminleri ve gerçek değerlerin karşılaştırılması .....   | 47 |
| Şekil 4.10. Popülasyon 75 - 10 elemanlı kafes örneği için algoritmaların hasar oranı tahminleri ve gerçek değerlerin karşılaştırılması .....  | 48 |
| Şekil 4.11. Popülasyon 100 - 10 elemanlı kafes örneği için algoritmaların hasar oranı tahminleri ve gerçek değerlerin karşılaştırılması ..... | 48 |
| Şekil 4.12. Popülasyon 20 – 10 elemanlı kafesin fonksiyon değeri değişimi.....                                                                | 49 |
| Şekil 4.13. Popülasyon 50 – 10 elemanlı kafesin fonksiyon değeri değişimi.....                                                                | 50 |
| Şekil 4.14. Popülasyon 75 – 10 elemanlı kafesin fonksiyon değeri değişimi.....                                                                | 50 |
| Şekil 4.15. Popülasyon 100 – 10 elemanlı kafesin fonksiyon değeri değişimi .....                                                              | 51 |
| Şekil 4.16. Popülasyon 20 - 10 elemanlı kafes örneği için algoritmaların hasar oranı tahminleri ve gerçek değerlerin karşılaştırılması .....  | 52 |
| Şekil 4.17. Popülasyon 50 - 10 elemanlı kafes örneği için algoritmaların hasar oranı tahminleri ve gerçek değerlerin karşılaştırılması .....  | 52 |
| Şekil 4.18. Popülasyon 75 - 10 elemanlı kafes örneği için algoritmaların hasar oranı tahminleri ve gerçek değerlerin karşılaştırılması .....  | 53 |
| Şekil 4.19. Popülasyon 100 - 10 elemanlı kafes örneği için algoritmaların hasar oranı tahminleri ve gerçek değerlerin karşılaştırılması ..... | 54 |
| Şekil 4.20. Popülasyon 20 – 72 elemanlı kafesin fonksiyon değeri değişimi.....                                                                | 55 |
| Şekil 4.21. Popülasyon 50 – 72 elemanlı kafesin fonksiyon değeri değişimi.....                                                                | 56 |
| Şekil 4.22. Popülasyon 75 – 72 elemanlı kafesin fonksiyon değeri değişimi.....                                                                | 56 |
| Şekil 4.23. Popülasyon 100 – 72 elemanlı kafesin fonksiyon değeri değişimi .....                                                              | 57 |
| Şekil 4.24. Popülasyon 20 – 72 elemanlı kafesin fonksiyon değeri değişimi.....                                                                | 58 |
| Şekil 4.25. Popülasyon 50 – 72 elemanlı kafesin fonksiyon değeri değişimi.....                                                                | 58 |
| Şekil 4.26. Popülasyon 75 – 72 elemanlı kafesin fonksiyon değeri değişimi.....                                                                | 59 |
| Şekil 4.27. Popülasyon 100 – 72 elemanlı kafesin fonksiyon değeri değişimi .....                                                              | 60 |

## ÇİZELGELER DİZİNİ

|                                                                                                                                                                     |    |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| <b>Çizelge 4.1.</b> 7 elemanlı örneğin Matlab ve SAP2000 ile hesaplanan doğal frekansların karşılaştırılması .....                                                  | 39 |
| <b>Çizelge 4.2.</b> 10 elemanlı örneğin SAP2000, MATLAB ve literatürdeki frekans değerlerinin karşılaştırılması .....                                               | 40 |
| <b>Çizelge 4.3.</b> İki farklı hasar senaryosu için elde edilen doğal frekans değerlerinin SAP2000, MATLAB ve referans makale sonuçları ile karşılaştırılması ..... | 42 |
| <b>Çizelge 4.4.</b> 10 Elemanlı Kafes (Senaryo 1): Algoritma performans istatistikleri .....                                                                        | 61 |
| <b>Çizelge 4.5.</b> 10 Elemanlı Kafes (Senaryo 1): Algoritmaların hasar durumları.....                                                                              | 61 |
| <b>Çizelge 4.6.</b> 10 Elemanlı Kafes (Senaryo 2): Algoritma performans istatistikleri .....                                                                        | 62 |
| <b>Çizelge 4.7.</b> 10 Elemanlı Kafes (Senaryo 2, Popülasyon 20): Algoritmaların hasar durumları .....                                                              | 62 |
| <b>Çizelge 4.8.</b> 10 Elemanlı Kafes (Senaryo 2, Popülasyon 50): Algoritmaların hasar durumları .....                                                              | 63 |
| <b>Çizelge 4.9.</b> 10 Elemanlı Kafes (Senaryo 2, Popülasyon 75): Algoritmaların hasar durumları .....                                                              | 63 |
| <b>Çizelge 4.10.</b> 10 Elemanlı Kafes (Senaryo 2, Popülasyon 100): Algoritmaların hasar durumları .....                                                            | 63 |
| <b>Çizelge 4.11.</b> 72 Elemanlı Kafes (Senaryo 1): Algoritma performans istatistikleri .....                                                                       | 64 |
| <b>Çizelge 4.12.</b> 72 Elemanlı Kafes (Senaryo 1): Algoritmaların hasar durumları.....                                                                             | 64 |
| <b>Çizelge 4.13.</b> 72 Elemanlı Kafes (Senaryo 2): Algoritma performans istatistikleri .....                                                                       | 65 |
| <b>Çizelge 4.14.</b> 72 Elemanlı Kafes (Senaryo 2): Algoritmaların hasar durumları.....                                                                             | 65 |

## 1. GİRİŞ

Günümüzde yapı güvenliği, özellikle deprem kuşağında yer alan ülkeler için hayati bir öneme sahiptir. Yaşanabilir kentsel alanların sürekliliği, kamu güvenliği, can ve mal kayıplarının önlenmesi gibi birçok faktör doğrudan yapıların taşıyıcı sistemlerinin sağlamlığına bağlıdır. Zaman içerisinde yapıların maruz kaldığı çevresel etkiler, malzeme yorgunluğu, kullanım hataları ya da doğal afetler sonucu oluşabilecek yapısal hasarlar, büyük ölçekli felaketlere yol açabilmektedir. Bu nedenle, yapıların çalışma ömürleri boyunca sağlıklarının izlenmesi, oluşabilecek hasarların erken aşamada tespiti ve gerekli önlemlerin zamanında alınabilmesi, modern inşaat mühendisliğinde temel önceliklerden biri haline gelmiştir.

Yapısal sağlık izleme sistemleri, yapıların dinamik davranışlarını analiz ederek hasarların tespit edilmesini sağlayan yöntemler bütünü olarak tanımlanabilir. Özellikle büyük ölçekli, karmaşık ve sürekli izlenmesi gereken yapılar için, geleneksel gözlemsel ve muayene yöntemleri zaman alıcı, maliyetli ve çoğu zaman yeterince hassas olmayan sonuçlar üretmektedir. Bu noktada, titreşim tabanlı yapısal hasar tespit yöntemleri ön plana çıkmaktadır. Titreşim verileri; yapıların doğal frekansları, mod şekilleri ve sönüm oranları gibi dinamik parametreler üzerinden elde edilir ve bu parametrelerdeki değişimler, yapıdaki potansiyel hasarların belirlenmesinde önemli ipuçları sunar. Ancak bu yöntemler kendi içinde bazı zorlukları da beraberinde getirir. Özellikle karmaşık yapılar söz konusu olduğunda, hasarların yeri ve büyüklüğünü doğru bir şekilde belirlemek, ciddi matematiksel modelleme ve hesaplama gücü gerektirir.

Bu noktada, hasar tespiti problemi, genellikle bir optimizasyon problemi olarak formüle edilir. Yapının ölçülen dinamik verileri ile oluşturulan sayısal modelin sonuçları arasındaki fark, bir amaç fonksiyonu aracılığıyla ifade edilir ve bu fonksiyonun minimuma indirilmesi hedeflenir. Bu sayede, yapıda hangi elemanların ne oranda hasar aldığına dair en uygun çözüm elde edilmeye çalışılır. Ancak bu tür optimizasyon problemleri genellikle çok sayıda yerel minimum içerdiğinden ve doğrusal olmayan çözüm uzayına sahip olduğundan, klasik deterministik optimizasyon yöntemleri çoğu zaman yetersiz kalmaktadır. Bu nedenle, son yıllarda yapısal hasar tespiti problemlerinde meta-sezgisel optimizasyon algoritmalarının kullanımı yaygınlaşmıştır.

Meta-sezgisel algoritmalar, doğadaki çeşitli fenomenlerden (evrim, hayvan davranışları, fiziksel süreçler vb.) esinlenerek geliştirilmiş, geniş arama alanlarını tarayabilen ve küresel optimuma ulaşma potansiyeli yüksek yöntemlerdir. Genetik algoritma (GA), parçacık sürüsü optimizasyonu (PSO), yapay arı kolonisi (ABC), Harmony Search (HS), çekirge optimizasyonu (GOA) gibi yöntemler literatürde sıkça uygulanmış ve hasar tespit problemlerinde başarılı sonuçlar elde edilmiştir. Ancak her bir algoritmanın, problem türüne ve parametre yapılarına göre değişen başarı düzeyleri vardır. Bu bağlamda, her yeni algoritma, hem daha kararlı hem de daha doğru sonuçlar elde edebilmek için potansiyel bir alternatif olarak değerlendirilmektedir.

Bu çalışmada odaklanılan Elektrikli Yılan Balığı Avlanma Optimizasyonu (Electric Eel Foraging Optimization, EEFO) algoritması, doğada elektrikli yılan balıklarının avlanma stratejilerinden esinlenerek geliştirilmiş yenilikçi bir meta-sezgisel yöntemdir. EEFO, arama uzayında keşif ve sömürü dengesini etkili biçimde sağlayarak, özellikle çoklu minimum içeren karmaşık problemler karşısında başarılı çözümler sunma

potansiyeline sahiptir. Algoritmanın farklı aşamaları olan etkileşim, dinlenme, avlama ve göç aşaması çözüm uzayını hem yerel hem de küresel düzeyde taramakta etkili bir yapı sunmaktadır. Literatürde yeni sayılabilecek bu yöntemin yapısal hasar tespitine uygulanması, çalışmanın özgün katkılarından biridir.

Bu tez çalışmasının temel amacı, EEFO algoritmasının yapısal hasar tespitinde ne derece etkin olduğunu ortaya koymak ve sonuçları, literatürde sıklıkla karşılaşılan HS ve ABC gibi bilindik meta-sezgisel algoritmalarla karşılaştırmalı olarak değerlendirmektir. Bu kapsamda, MATLAB programlama ortamında geliştirilen bir kod altyapısı aracılığıyla, hem iki boyutlu (10 elemanlı düzlemsel kafes) hem de üç boyutlu (72 elemanlı uzaysal kafes) yapı sistemleri analiz edilmiş, hem tekli hem de çoklu hasar senaryoları test edilmiştir. Her bir algoritma, farklı popülasyon büyüklüklerinde (20, 50, 75, 100) ayrı ayrı çalıştırılarak, performanslarının popülasyon duyarlılığı da analiz edilmiştir.

Bu bağlamda, çalışma hem algoritma düzeyinde bir yenilik sunmakta, hem de yapısal hasar tespitine yönelik kapsamlı bir analiz ortaya koymaktadır. Ayrıca, önerilen algoritmanın SAP2000 yazılımıyla doğrulanması, elde edilen sonuçların güvenilirliğini artırmaktadır. Tezin devamında, kullanılan yöntem, matematiksel modelleme süreci, test senaryoları, elde edilen bulgular ve istatistiksel karşılaştırmalar detaylı olarak sunulmuş, sonuç bölümünde ise EEFO algoritmasının potansiyeli değerlendirilerek gelecek çalışmalara yönelik öneriler geliştirilmiştir.

## 2. KAYNAK TARAMASI

Yapısal hasar tespiti, özellikle mühendislik ve inşaat sektörlerinde güvenlik, sürdürülebilirlik ve maliyet etkinliği açısından kritik bir öneme sahiptir. Günümüzde yapılardaki hasarı tespit etmek için bir çok yöntem geliştirilmiştir ve özellikle optimizasyon süreçlerini içeren yöntemler sıklıkla tercih edilmektedir. Yapısal hasar tespitinde optimizasyon yöntemleri, yapıdaki hasarları modellemek ve bu model ile gerçek gözlemler arasındaki farkı minimize etmek için kullanılır. Bu süreç, hasar parametrelerini (yer ve büyüklük) doğru bir şekilde belirlemek amacıyla iteratif bir arama süreci gerektirir. Meta-sezgisel yöntemler, geleneksel optimizasyon yöntemlerinin aksine, daha hızlı ve daha verimli sonuçlar sunarak, yapısal hasar tespitinin doğruluğunu ve güvenilirliğini artırır.

Bu tez çalışması, meta-sezgisel optimizasyon yöntemlerine yeni bir bakış açısı getirerek, Elektrikli Yılan Balığı Avlanma Optimizasyonu (EEFO) algoritmasının, yapısal hasar tespitindeki etkinliğini incelemekte ve mevcut literatürdeki yöntemlerden farklı olarak, daha verimli bir çözüm arama stratejisi sunmaktadır. Bu bağlamda, kaynak taraması ‘Yapısal Hasar Tespitinde Meta-sezgisel Yöntemlerin Kullanılması’ ve ‘EEFO’nun Çeşitli Problemlerdeki Performansının İncelenmesi’ olmak üzere iki ana başlıkta incelenmiştir.

### 2.1.Yapısal Hasar Tespitinde Meta-sezgisel Yöntemlerin Kullanılması

Yapısal hasar tespitinde etkinliğini kanıtlamış yöntemlerden biri, James Kennedy ve Russel Eberhart (1995) tarafından tanıtilan parçacık sürüsü optimizasyon (PSO) algoritmasıdır. Parçacık sürüsü optimizasyon algoritmasının yapısal hasar tespitindeki bazı çalışmaları şunlardır:

Seyedpoor (2012) yaptığı çalışmada, çoklu hasar senaryolarında hasar yerini ve şiddetini doğru bir şekilde belirlemek için iki aşamalı bir yöntem önermektedir. İlk aşamada, yapısal elemanlardaki hasarı belirlemek amacıyla modal gerinim enerjisi tabanlı bir indeks (MSEBI) geliştirilmiştir. Bu indeks, sonlu eleman modellemesinden elde edilen mod analiz bilgileri kullanılarak hesaplanmakta ve hasarlı yapıdaki elemanlar için sıfırdan büyük değerler verir. MSEBI, hasar yerini tespit etmek için güçlü bir araçtır. İkinci aşamada ise, PSO algoritması kullanılarak, MSEBI ile belirlenen hasarlı elemanların kapsamı doğrulanır. Çalışma, 15 elemanlı bir konsol kirişi ve 31 çubuklu düzlemsel kafes yapısı üzerinde test edilmiştir. Sayısal sonuçlar, MSEBI ve PSO’nun birleşiminin çoklu yapısal hasarı doğru tespit etmek için etkili bir yöntem sunduğunu göstermektedir. Bu çalışma, yapısal hasar tespiti için önemli bir katkı sağlamaktadır.

Gökdağ ve Yıldız (2012) çalışmalarında, yapısal hasar tespitinde titreşim tabanlı yöntemlerin ve özellikle PSO algoritmasının kullanımını incelemiştir. Çalışmada, Timoshenko kirişinin sonlu eleman modeli ele alınarak, hasar rijitlik kaybı şeklinde modellenmiştir. Hasar yerleri ve büyüklükleri, modal parametre tabanlı amaç fonksiyonları minimize edilerek belirlenmiştir. Sonuçlar, modal esnekliğin en iyi hasar indeksi olduğunu ve PSO’nun yapısal hasar tespitinde etkili bir optimizasyon aracı sunduğunu göstermiştir. Ayrıca, PSO’nun yanı sıra genetik algoritma gibi evrimsel algoritmaların kullanımına da değinilmiş ve modal esnekliğin, diğer yöntemlere göre daha etkili olduğu vurgulanmıştır. Bu çalışma, PSO algoritmasının modal esneklikle

birlikte kullanılmasının, yapısal hasarın doğru tespiti açısından önemli katkılar sunduğunu ortaya koymaktadır.

Bazı araştırmacılar PSO'yu daha etkili hale getirmeye çalışmışlardır. Örneğin; Kang vd. (2012), yapıların hasar tespiti için Yapay Bağışıklık Sistemini (YBS), parçacık sürü optimizasyonuna entegre eden bir Bağışıklık Güçlendirilmiş Parçacık Sürüsü Optimizasyonu (IEPSO) algoritması sunmaktadır. Çalışmada, yapay bağışıklık sisteminin, seçim, reseptör düzenleme ve aşılama gibi bazı bağışıklık mekanizmaları, PSO algoritmasının performansını iyileştirmek için temel PSO'ya entegre edilmiştir. Hasar tespiti için kullanılan amaç fonksiyonu, doğal frekanslar ve mod şekilleri gibi titreşim verilerine dayanmaktadır. Çalışmada, IEPSO'nun performansı, temel PSO, diferansiyel evrim algoritması ve gerçek kodlu genetik algoritmayla iki örnekte karşılaştırılmıştır. Sonuçlar, IEPSO'nun yapısal hasarların yerini ve şiddetini belirlemede etkili olduğunu göstermiştir. IEPSO'nun, yapısal hasar tespiti için kullanılan klasik optimizasyon yöntemlerine göre bazı avantajları vardır. Bunlar arasında, daha hızlı yakınsama hızı, daha yüksek doğruluk ve daha etkili küresel arama yeteneği sayılabilir. Özellikle de, IEPSO algoritmasının geleneksel optimizasyon yöntemlerine göre daha etkili bir çözüm sağladığı gösterilmiştir.

Başka bir çalışmada, Wei vd. (2018), çalışmalarında, yapısal hasar tespiti için geliştirilmiş bir parçacık sürü optimizasyonu algoritması sunmaktadır. Algoritmanın ana özelliği, erken yakınsamadan kaçınmak için evrimsel sürece bir bozulma (disturbance) eklemesidir. Bu bozulma, küresel veya bireysel en iyi bilinen konumların mutasyonunu hedef alarak sürüyü yerel minimumdan uzaklaştırmaya yönlendirmektedir. Geliştirilmiş PSO algoritmasının etkinliği ve sağlamlığı, giriş, kafes ve plaka gibi farklı yapısal modeller üzerinde doğrulanmıştır. Sonuçlar, yöntemin ölçüm gürültüsü dikkate alındığında yapısal hasar tespiti için etkili ve verimli olduğunu göstermiştir. Önerilen yöntemin farklı yapısal modeller üzerinde test edilmesi ve performansının diğer yöntemlerle karşılaştırılması, bu çalışmanın yapısal hasar tespiti alanına önemli bir katkı sağladığını göstermektedir.

Minh vd. (2021), 3 boyutlu elektrik iletim kulesinde hasar tespiti için Geliştirilmiş Parçacık Sürü Optimizasyonu (EHVPSO) algoritmasını sunmaktadır. Çalışmada, yapıların hasar tespitinde kullanılan SAP2000 yazılımı ve MATLAB arasında kesintisiz veri alışverişini sağlayan bir program geliştirilmiştir. Bu program, her iki yazılım arasında etkileşim sağlayarak, yapıdaki parametrelerin değiştirilmesine ve hasarın doğru bir şekilde belirlenmesine olanak tanır. EHVPSO algoritmasında, her bir parçacığın yakınsama oranını kontrol etmek ve yerel ile küresel en iyi değerler arasındaki dengeyi sağlamak amacıyla iki yeni denklem önerilmiştir. Bu yenilikçi yaklaşım, optimizasyon sürecinin verimliliğini artırmakta ve daha hızlı sonuçlar elde edilmesini sağlamaktadır. Sonuçlar, EHVPSO algoritmasının yüksek doğruluk ve güvenilirlik ile yapısal hasarın yerini ve büyüklüğünü tespit edebildiğini ortaya koymuştur. Ayrıca, PSO'nun performansını iyileştiren yeni özellikler, algoritmanın karmaşık yapılar için daha etkili hale gelmesini sağlamaktadır.

Sandesh vd. (2010), çalışmalarında yapısal hasar tespiti için geliştirilmiş hibrit bir genetik algoritma (GA) ve parçacık sürüsü optimizasyonu algoritması sunmaktadır. Çalışmada, öncelikle PSO ve GA'nın performansı, Ackley ve Schwefel gibi çok modlu referans fonksiyonları üzerinde karşılaştırılmıştır. Sonuçlar, hibrit algoritmanın hem

yakınsama hem de doğruluk açısından PSO ve GA'dan daha üstün olduğunu göstermiştir. Ardından, hibrit algoritma, ince bir plaka üzerinde zaman tanım alanı formülasyonu kullanılarak çoklu çatlak hasarlarını tespit etmek için uygulanmıştır. Hasar tespiti, gerinim enerjisi eşdeğerliği ilkesine dayalı ortotropik bir Sonlu Eleman (FE) modeli ile gerçekleştirilmiştir. Zaman tanım alanı ivme yanıtları kullanılarak üç çatlak tespit edilmiştir. Elde edilen sonuçlar, hibrit algoritmanın, hem hız hem de doğruluk açısından, PSO ve GA'dan daha etkili olduğunu ortaya koymuştur. Bu çalışma, yapısal hasar tespitinde hibrit algoritmaların etkinliğini ve potansiyelini vurgulamaktadır.

Chen vd. (2018) tarafından yapılan bir çalışmada, yapısal hasar tespiti için Hibrit Parçacık Sürüsü Optimizasyonu (HPSO) ile Monte Carlo simülasyonlarının birleştirildiği yeni bir strateji önerilmektedir. Çalışmada, önce PSO algoritmalarının farklı parametreleri, Monte Carlo simülasyonları aracılığıyla test edilerek, en uygun parametre kombinasyonları belirlenmiştir. Bu adımda, hasar tespiti için hangi parametrelerin daha etkili olduğu analiz edilmiştir. Ardından, yerel arama algoritması olarak Nelder-Mead algoritması, PSO ile entegre edilerek, küresel arama kabiliyeti artırılmıştır. Sayısal simülasyonlar, 10 ve 20 sonlu elemanlı basitleştirilmiş kirişler üzerinde yapılmış ve önerilen yöntemle etkili hasar tespiti sağlanmıştır. Ayrıca, kutu kesitli çelik kirişler üzerinde yapılan deneysel çalışmalarda, farklı çatlak hasar örüntüleriyle yapısal frekanslar ve mod şekilleri ölçülmüş ve önerilen yöntem doğrulanmıştır. Sonuçlar, HPSO'nun karmaşık yapıların tespitinde daha etkili bir çözüm sunduğunu ortaya koymaktadır.

Yapısal hasar tespitinde kullanılan yöntemlerden biri, Saremi vd. (2017) tarafından ortaya atılan Çekirge Optimizasyon Algoritması (GOA)'dır. Çekirgelerin yiyecek arama davranışlarından esinlenilmiş, topluluk tabanlı bir algoritmadır. Yapısal hasar tespiti probleminde kullanılan çekirge optimizasyon algoritması ile ilgili çalışmalar aşağıdaki gibi listelenmiştir.

Nabavi vd. (2021), çerçeve elemanlarında hasar tespiti için zaman tanım alanı yanıtları ile GOA'yı kullanan etkili bir yöntem önermiştir. Çalışmada, ilk olarak hasarlı yapının ivme değerleri ile analitik modelin ivme değerlerinden bir amaç fonksiyonu tanımlanmış ve bu fonksiyon optimizasyon sürecinde kullanılmıştır. GOA yapının hasar yerini ve şiddetini belirlemek amacıyla optimizasyon problemine dönüştürülmüştür. Etkinlik analizi için Newmark yöntemi gibi sayısal bir teknik kullanılmıştır. Sonuçlar, ölçüm gürültüsüne rağmen, önerilen yöntemin hasarın yerini ve şiddetini doğru şekilde tespit ettiğini göstermektedir. Ayrıca, çelik köprülerde kiriş aşınması ve deniz platformlarındaki köşe üyelerinin bozulması gibi çeşitli yapılar incelenmiştir. Çalışma, GOA ile yapılan hasar tespitinin, bu alanda yeni bir bakış açısı sunduğunu ortaya koymaktadır.

Aynı araştırmacılar tarafından yürütülen başka bir çalışmada (2022), bu sefer aynı yöntem 16 sonlu elemana bölünmüş bir plakada uygulanmıştır. Yapı tam merkezinden bir kuvvet ile uyarılmıştır. Ölçüm gürültüsünü de dikkate alan bu çalışmada, GOA'nın

etkinliği farklı sistemlerde de verimli ve uygulanabilir olduğunu göstermiştir. GOA algoritmasının performansı, ölçüm gürültüsünün etkisini de dikkate alarak gerçekleştirilen sayısal örneklerle değerlendirilmiştir. Elde edilen sonuçlar, GOA'nın, yapısal hasarın yerini ve şiddetini doğru bir şekilde belirlemede etkili olduğunu göstermiştir. Ayrıca, GOA'nın performansı, Diferansiyel Evrim Algoritması (DEA) ile karşılaştırılmış ve GOA'nın daha etkili olduğu sonucuna varılmıştır.

Yapısal hasar tespiti problemindeki performansı en çok değerlendirilen algoritmalarından bir tanesi de Genetik Algoritma (GA)'dır (Holland, 1975). Algoritmanın çalışma prensibi, canlıların evrimsel süreçlerindeki çaprazlama, mutasyon gibi biyolojik evrelerine dayanmaktadır. Oldukça popüler olan bu algoritma, yapısal hasar tespitinde birçok kez kullanılmıştır.

Moslem ve Nafaspour (2002) tarafından yapılan bir çalışmada, yapısal hasar tespiti için eksik ölçümler kullanan iki aşamalı bir yöntem sunulmaktadır. İlk aşamada, kalan kuvvet yöntemi kullanılarak potansiyel hasarlı elemanlar belirlenmektedir. İkinci aşamada ise, simüle edilmiş evrim yöntemine dayalı bir optimizasyon şeması kullanılarak, birinci aşamadan elde edilen bilgiye göre hasarın büyüklüğü belirlenmektedir. Yöntem, genetik algoritmalar içerir ve bu algoritmaların yakınsamasını iyileştirmek için bazı stratejiler uygulanmıştır. Çalışmada, genetik algoritmaların yapısal hasar tespiti için kullanılmasının temel teorisi açıklanmaktadır ve yöntemin uygulanması için bir akış şeması sunulmaktadır. Ayrıca, iki farklı yapısal sistemde, yöntemin başarısını gösteren sayısal örnekler sunulmaktadır. Bu çalışma sayesinde, eksik ölçümlerle çalışabilen ve yapısal hasarın hem konumunu hem de büyüklüğünü belirleyebilen yeni bir hasar tespit yöntemi sunulmuştur. Yöntem, genetik algoritmalar kullanarak optimizasyon problemi çözüyor ve geleneksel yöntemlerden daha iyi bir performans sergiliyor. Sonuç olarak, makale, yapısal hasar tespiti için eksik ölçümlerle çalışabilen ve genetik algoritmalar kullanarak verimli ve etkili bir çözüm sunan yeni bir yöntem sunmaktadır.

Hao ve Xia (2002) yapısal hasarın tespiti için gerçek sayı kodlamasıyla genetik algoritma kullanarak yeni bir yöntem önerdiler. Bu yöntem, ölçülen titreşim verilerindeki değişiklikleri, hasarlı ve hasarsız yapı modelleri arasındaki farkları minimize ederek, yapısal hasarın yerini ve büyüklüğünü belirlemeyi amaçlıyor. Önerilen yöntem, iki farklı yapısal model üzerinde (konsollü giriş ve çerçeve) test ediliyor ve sonuçlar, ölçülen doğal frekansları ve mod şekillerini kullanarak yapısal hasarın yerini ve büyüklüğünü doğru bir şekilde tespit ettiğini ve gürültüye karşı dayanıklı olduğunu gösteriyor. Ayrıca, bu yöntem, geleneksel yöntemlere göre, daha az işlem gerektiren, daha hassas ve gürültüye karşı daha dayanıklı bir yöntem olarak öne çıkıyor. Çalışma, genetik algoritma kullanarak yapısal hasar tespiti için yeni bir bakış açısı getiriyor. Geleneksel yöntemlere göre, daha az işlem gerektiren, daha hassas ve gürültüye karşı daha dayanıklı bir yöntem olarak öne çıkıyor.

Cha ve Buyukozturk (2015), üç boyutlu çelik yapıların yapısal sağlığı izlenmesinde hasar tespiti için modal gerinim enerjisi (MSE) ve hibrit çok amaçlı optimizasyon algoritmaları kullanan yeni bir yöntem sunmaktadır. Geleneksel yöntemler, genellikle basit kirişlerde veya düzlemsel yapılarda etkili olmakla birlikte, üç boyutlu yapılardaki çoklu küçük hasarları tespit etmekte zorlanmaktadır. Bu zorluk, küçük hasarların yapının modal özelliklerinde belirgin değişiklikler yaratmamasından kaynaklanmaktadır. Çalışmada, bu sorunu çözmek için hibrit çok amaçlı genetik algoritmalar tabanlı bir yöntem önerilmektedir. MSE, yapısal hasarları simüle etmek için kullanılırken, çok amaçlı NS2-IRR GA algoritması, FE model güncelleme parametrelerini optimize ederek MSE'deki farkları en aza indirir. Yöntem, üç farklı üç boyutlu çelik çerçeve yapısında test edilmiş ve gürültülü ölçümlerde bile doğru hasar tespiti sağlanmıştır.

Alkayem vd. (2018) sonlu eleman model güncellemelerini evrimsel algoritmalar ile kullanarak yapısal hasar tespitini ele almıştır. FE model güncellemeleri, yapısal hasarın dinamik özelliklerle eşleştirilmesi amacıyla yaygın bir yöntemdir. Bu işlem, yapısal hasarın yerini ve şiddetini doğru bir şekilde belirlemek için gereklidir. Evrimsel algoritmalar, karmaşık ve doğrusal olmayan optimizasyon sorunlarını çözmede ve hasarın dağılımını belirlemede etkili bir araçtır. Çalışma, FE model güncellemeleri için tek hedefli ve çok hedefli EA'ların kullanımını incelemektedir. GA, PSO ve hibrit yöntemler gibi tek hedefli algoritmaların yanı sıra, NSGA-II ve IRR-GA gibi çok hedefli algoritmalar da kullanılmıştır. Çalışmada, her iki tür EA'nın performansı karşılaştırılmış ve çok hedefli EA'ların, hasar tespitinde daha iyi sonuçlar verdiği vurgulanmıştır. Bu çalışma, FE model güncellemesinde EA'ların uygulanmasına kapsamlı bir bakış sunarak, farklı algoritma türleri ve parametre seçim stratejilerini ele almaktadır.

Seyedpoor vd. (2019) çalışmasında, zaman alanı ivme tepkileri ve DEA kullanarak yapısal hasarın yerini ve şiddetini belirlemeyi amaçlayan yeni bir yöntem öneriyor. Geleneksel yöntemler, yapısal doğal frekanslar veya mod şekillerindeki değişiklikleri kullanarak hasar tespiti yaparken, özellikle büyük yapılarda ve çoklu küçük hasarlarda hassasiyet kaybı yaşanabiliyor. Ayrıca, zaman alanı yöntemleri daha fazla işlem gücü gerektiriyor ve yapısal değişiklikler karşısında hassasiyetini yitiriyor. Bu çalışmada, zaman alanı ivme tepkileri, darbe yüklerinden elde edilerek diferansiyel evrim algoritmasıyla optimize ediliyor. DEA, hedef fonksiyonu minimize ederek hasarın yerini ve şiddetini belirliyor. Yapısal model üzerinde yapılan testler, önerilen yöntemin hem hassasiyetini hem de gürültüye karşı dayanıklılığını gösteriyor. Ayrıca, bu yöntem, daha az sensörle çevrimiçi uygulamalar için uygun hale geliyor ve geleneksel yöntemlerden daha hızlı ve pratik bir çözüm sunuyor.

Bu konuda yapılan çalışmalarda literatürde karşılaşılan bir diğer algoritma Xin-She Yang (2010) tarafından geliştirilen Yarasa Algoritması (BA)'dır. Yarasaların çevrelerini algılamak ve avlarını tespit etmek için kullandıkları yankılanma tekniğini

taklit eden algoritma birçok çalışmada kullanılmış olup, yapısal hasar tespitindeki etkileri bazı çalışmalarda tartışılmıştır.

Ghiasi vd. (2019) büyük ölçekli uzay yapılarında hasar tespitine yönelik üç aşamalı bir yöntem önermektedir. Çalışmanın amacı, karmaşık yapısal sistemlerde hasar yerini ve şiddetini doğru bir şekilde belirlemektir. İlk aşamada, yapının alt yapıları oluşturulmakta ve her bir alt yapının rijitlik matrisleri elde edilmektedir. İkinci aşamada, Modal Gerinim Enerjisi Tabanlı İndeks (MSEBI) kullanılarak alt yapının hasar yeri tespit edilmektedir. Üçüncü aşamada ise, hasar şiddetleri, geliştirilmiş EBA ile tahmin edilmektedir. Bu yaklaşım, yapısal hasar tespitini kısıtlanmamış bir optimizasyon problemi olarak ele almaktadır. Önerilen yöntem, büyük ölçekli yapılar için uygulama zorluğunu aşarak, her türlü serbestlik derecesine sahip yapılar için etkili bir çözüm sunmaktadır. Çalışma, meta-sezgisel optimizasyon algoritmalarının kullanımını yenilikçi bir yaklaşım olarak vurgulamaktadır. Sonuçlar, yöntemin yüksek doğruluk sağladığını ve mühendislik uygulamalarında pratik bir çözüm sunduğunu göstermektedir.

Su vd. (2021), çalışmalarında, yapısal hasar tespiti için Modifiye Edilmiş Yönlü Yarasa Algoritması (MDBA) kullanarak yeni bir yöntem önermektedir. Normal yarasa algoritması, erken yakınsama ve çeşitlilik eksikliği gibi dezavantajlar taşıırken, MDBA algoritması bu sorunları, bireysel en uygun güncelleme mekanizması ve eleme stratejisi ekleyerek çözmektedir. Bireysel en uygun güncelleme mekanizması, yarasaların konumlarını, küresel en uygun çözümün etkisiyle güncellerken, eleme stratejisi kötü performans gösteren bireyleri eler ve yeni bireyler oluşturur. Çalışmada, MDBA algoritmasının performansı kiriş ve kafes tipi köprü modelleri üzerinde sayısal simülasyonlarla değerlendirilmiş ve diğer sürü optimizasyon algoritmalarıyla karşılaştırılmıştır. Sonuçlar, MDBA'nın daha hassas hasar tespiti ve hızlı yakınsama sağladığını gösteriyor. Ayrıca, MDBA, yüksek gürültü seviyelerinde bile doğru hasar tespiti yaparak, gürültüye karşı dayanıklı bir yöntem olduğunu kanıtlamaktadır.

Greco vd. (2019), çerçeve yapılarındaki hasarın tespiti için yeni bir optimizasyon algoritması geliştirmeyi amaçlamaktadır. Yapısal sağlık izleme, mevcut yapıların güvenliğini sağlamak açısından kritik öneme sahiptir ve erken hasar tespiti büyük bir gerekliliktir. Çalışmada, hasarın tespiti için dinamik özellikler, özellikle doğal frekanslardaki değişiklikler incelenmiştir. Geliştirilen algoritma, NetLogo yazılım ortamında uygulanarak beş katlı bir çerçeve yapısı üzerinde test edilmiştir. Hasarın yerini tespit etmek için 49 kritik kesim değerlendirilmiş ve her bir kesim için sekiz farklı hasar yoğunluğu seviyesi analiz edilmiştir. Algoritma, hem hasarın yerini doğru şekilde belirlemiş hem de fitness fonksiyonunu minimize ederek hasar yoğunluğunu tespit etmiştir. Sonuçlar, yalnızca birkaç doğal frekansın kullanılarak yüksek başarıyla hasar tespiti yapılabildiğini göstermektedir. Bu çalışma, mevcut yöntemlere kıyasla daha etkili ve verimli bir yaklaşım sunarak önemli bir katkı sağlamaktadır.

Yapısal hasar tespiti problemindeki performansı değerlendirilen bir diğer yöntem de Storn ve Price (1997) tarafından geliştirilen Diferansiyel Evrim Algoritması (DEA)'dır. Topluluk tabanlı meta-sezgisel bir yöntem olan bu algoritma pek çok çalışmada kullanılmış olup, yapısal hasar tespitinde de kullanılan algoritmalarından biridir. Bu konudaki çalışmaları ise aşağıda anlatılmıştır.

Seyedpoor vd. (2015), çalışmalarında, yapısal hasarı tespit etmek için diferansiyel evrim algoritmasından yararlanmışlardır. Çalışmada önerilen yöntem, yapısal hasar tespitini, sürekli değişkenler içeren standart bir optimizasyon problemine dönüştürüyor. Daha sonra, DEA algoritması, hasarın yerini ve şiddetini bulmak için optimizasyon problemini çözmek için kullanılıyor. Yöntemin performansını değerlendirmek için, ölçüm gürültüsünü de dikkate alarak bazı test örnekleri sayısal olarak test edilmiştir. Sayısal sonuçlar, yöntemin, hasarın yerini ve şiddetini doğru bir şekilde belirleyebildiğini göstermiştir. DEA, standart parçacık sürü optimizasyon algoritmasından daha iyi performans göstermiştir.

Kim vd. (2019), düzlem ve uzay kafes yapılarındaki çoklu hasarların yerini ve boyutlarını tespit etmek için DEA ve titreşim verilerini kullanarak etkili bir hasar tespit tekniği önermektedir. Çalışmada, genel denge denklemleri ve düğümlerdeki reaksiyon kuvvetleri dikkate alınarak uyum denklemleri, tekil değer ayrıştırma (SVD) tekniği ile sunulmuş ve kuvvet yöntemi temel alınarak genişletilmiş hareket denklemleri geliştirilmiştir. Hasar tespiti için hedef fonksiyon, doğal frekanslar ve mod şekilleri gibi titreşim verilerine dayanır. Çalışma, düzlem ve uzay kafes yapıları için üç sayısal örnekle doğrulanmış ve sonuçlar, DEA'nın GA'ya kıyasla çoklu hasarların yerini ve boyutunu tespit etmede daha hızlı ve doğru sonuçlar verdiğini göstermektedir. DE algoritması, düzlem kafes yapısında %0 hata oranı ile GA'ya kıyasla %36.83 hata oranı elde edilmesini sağlamıştır. Sonuç olarak, DE ve kuvvet yöntemi kombinasyonu, bu alanda önemli bir katkı sunmaktadır.

Yapısal hasar tespitinde kullanılan bir diğer algoritma, Rao vd. (2011) tarafından tanıtilen Öğretim-Öğrenme Tabanlı Optimizasyon (TLBO) algoritmasıdır. Yapısal hasar tespitindeki çalışmaları aşağıdaki gibidir.

Du vd. (2018) çalışmalarında, 2D yapıların hasar tespiti için yeni bir yöntem önermektedir. Yöntem, tam olmayan ölçümler kullanılarak yapılan model güncellemesi ve geliştirilmiş TLBO uygulanmaktadır. Önerilen yöntem, yapısal çerçevelerin titreşim verileri tabanlı objektif fonksiyon kullanmaktadır. Çalışmada, yalnızca beş adet tamam olmayan mod kullanılarak hasarın yerini ve şiddetini tahmin etmek için yapılan simülasyon sonuçları sunulmuştur. Elde edilen sonuçlar, yöntemin, gürültü varlığında bile (ölçülen verilerde %4'lük gürültü) yapısal hasar tespitinde doğru sonuçlar ürettiğini göstermiştir. Ayrıca, yöntemin geleneksel optimizasyon algoritmalarına (örneğin, PSO, GA, HS, ABC) kıyasla daha hızlı ve doğru sonuçlar ürettiği ve hesaplama maliyetini düşürdüğü vurgulanmıştır.

Bir başka çalışmada Barman vd. (2021), kirişler, uzay çerçeveleri ve plakalar gibi yapıların yapısal sağlık izleme için kullanılabilen, Bayes veri kaynaştırmasını TLBO algoritması ile birleştiren yeni bir yöntem sunuyor. Yöntem, yapısal hasarın yerini ve şiddetini belirlemek için, doğal frekanslar ve mod şekilleri gibi titreşim parametrelerini kullanan bir dizi hasar indeksinden faydalaniyor. Bu hasar indeksleri arasında DIMSE, FRFSEDR, FSEDR ve RFBDI bulunuyor. Daha sonra, Bayes veri kaynaştırması, yapıdaki olası hasarlı elemanları belirlemek için bu hasar indekslerine uygulanıyor. Önerilen yaklaşım, yapısal optimizasyon algoritmasının işlem süresini azaltarak, yapıdaki olası hasarlı eleman sayısını önemli ölçüde azaltıyor. Bu algoritma, gürültülü ortamlarda da umut vadeden performans gösteriyor. Önerilen yöntemin, kiriş, uzay çerçevesi ve plaka yapıları üzerinde test edildiğini gösteriyor. Ayrıca, yöntemin çeşitli hasar senaryolarına, gürültüye karşı dayanıklılığına ve sensör yerleşimi değişkenliğine karşı hassasiyetine ilişkin sonuçlar da sunuluyor. Sonuçlar, yöntemin yapısal hasarı doğru bir şekilde tespit edebildiğini ve geleneksel tek aşamalı optimizasyon yöntemlerinden daha verimli olduğunu gösteriyor.

Literatürde, yapısal hasar tespiti probleminde, yukarıda bahsedilen çalışmalar dışında daha pek çok farklı algoritmaların kullanıldığı çalışmalar da yer almaktadır. Farklı algoritmaların yapısal hasar tespiti üzerindeki performanslarını değerlendiren çalışmalar aşağıdaki gibi özetlenmiştir.

Kaveh ve Maniat (2014) yapıların hasar tespiti için yeni bir Şarjlı Sistem Arama algoritması (CSS) uygulamıştır. Çalışmada, yapısal elemanların hasar görmesi durumunda, yapının davranışının değiştiği ve bu değişikliklerin hasarın yerini ve şiddetini belirlemek için kullanılabileceği vurgulanmaktadır. Gürültünün ölçülen veriler üzerindeki etkisini azaltmak için ceza yaklaşımı uygulanmıştır. Kiriş, çerçeve ve köprü modelleri üzerinde yapılan sayısal analizler sonuçları sunulmuştur. Elde edilen sonuçlar, CSS'nin, hem gürültülü hem de gürültüsüz veriler kullanılarak yapılan simülasyonlarda, yapısal hasarları yerini ve şiddetini doğru bir şekilde belirleyebildiğini göstermiştir.

Majumdar vd. (2014) çalışmalarında, mod testleri ile belirlenen titreşim verileri ile analitik model tarafından hesaplanan veriler arasındaki farkın en aza indirilmesini amaçlayan ters bir problem oluşturulmuş ve bu problem, sürekli karınca kolonisi optimizasyonu (ACO) algoritması kullanılarak çözülmüştür. Hasar, rijitlik azaltma faktörü olarak formüle edilmiştir. Çalışma, geliştirilen kodun geniş bir çeşitlilikteki ters tanımlama problemlerini çözme potansiyelini göstermiştir.

Li ve Lu (2015) çalışmasında, Çoklu-Sürü Meyve Sinekleri Optimizasyon Algoritması (MFOA) kullanarak yapısal hasar tespit etmeyi amaçlamaktadır. Çalışmada, hasarın sadece eleman rijitliğini azaltması sonucu ortaya çıktığı varsayılmıştır. Hasarlı ve sağlam durumdaki yapının doğal frekanslarındaki ve mod şekillerindeki farklılıklar, hedef fonksiyonu oluşturmak için kullanılmış ve bu yöntemle bir hasar tespit problemi bir optimizasyon problemine dönüştürülmüştür. MFOA'nın verimliliği ve doğruluğu üç

farklı yapı kullanılarak gösterilmiştir. Sayısal sonuçlar, MFOA'nın, orijinal Meyve Sinekleri Optimizasyon Algoritmasından (FOA) daha iyi bir kapasiteye sahip olduğunu göstermiştir. Bu çalışma, altyapı hasar tespit etmek için yeni bir sürü zekâ algoritması sunmaktadır. Bu yöntem, altyapının gerçek hasar bilgilerini uygulamaya koymayı amaçlamaktadır. Altyapının hasar yerini ve şiddetini belirlemek için doğal frekanslardaki değişikliklerin ve doğal frekans vektör teminat kriterinin (NFVAC) kullanıldığı bir hedef fonksiyon sunulmaktadır. Fonksiyonu en küçük değere indirmek için, SSA kullanılmıştır. Önerilen yöntemin verimliliği, üç deneysel örnek kullanılarak doğrulanmıştır.

Nhamagea vd. (2015) tarafından yapılan bir çalışmada, yapısal hasar tespiti için yeni bir hibrit stokastik/deterministik Pincus-Nelder-Mead optimizasyon algoritması (P-NMA) sunmaktadır. Önerilen algoritma, titreşim tabanlı hasar tespiti problemini çözmek için geliştirilmiştir. Algoritmanın en önemli özelliği, Pincus formülünün örnekleme alanını hasarlı elemanların maksimum sayısını kısıtlayarak azaltmasıdır. Bu değişiklik, hasarlı elemanların maksimum sayısını belirleyen yeni bir parametre (npmax) tanımlanmıştır. Bu parametre, tasarımcı tarafından seçilmektedir. Değişikliğin amacı, P-NMA algoritmasının yakınsama hızını artırmak ve hesaplama maliyetini düşürmektir. Literatürden seçilen bir dizi sayısal örnek üzerinde çalışmalar yapılmıştır. Önerilen iyileştirilmiş optimizasyon algoritmasının (IP-NMA), orijinal P-NMA ve meta-sezgisel uyum arama algoritmasına (HS) kıyasla daha iyi performans gösterdiği kanıtlanmıştır. IP-NMA daha doğru sonuçlar üretmiş ve hesaplama maliyetini düşürmüştür. Algoritma, P-NMA'nın yakınsama hızını artırıyor ve hesaplama maliyetini düşürüyor. Algoritmanın orijinal P-NMA ve meta-sezgisel HS algoritmalarından daha iyi performans gösterdiği kanıtlanmıştır.

Nobahari vd. (2016) yapısal hasar tespiti için Ekolokasyon Arama Algoritması (ESA) kullanmışlardır. ESA, ekolokasyon yapan hayvanların yiyecek bulma davranışlarından esinlenerek geliştirilmiştir ve hasarın şiddetini optimizasyon değişkeni olarak kullanır. Algoritma, rastgele seçilen bir noktadan başlayarak, çevredeki noktalarla etkileşime girer ve fitness değerlerine göre hareket yönünü belirler. Çalışmada, ESA algoritması 31, 10 ve 47 elemanlı yapısal sistemler üzerinde test edilmiştir. Sonuçlar, ESA'nın hem gürültüsüz hem de gürültülü ölçümlerde doğru sonuçlar verdiğini ve diğer sezgisel algoritmalara (örneğin GA, PSO) kıyasla daha etkili ve verimli olduğunu göstermiştir.

Nhamagea vd. (2016) çalışmalarında, altyapı hasar tespit sürecindeki hesaplama maliyetini azaltmayı hedefleyerek, hasar tespitini bir olasılığa dayalı çerçeve ile ele almışlardır. Bu amaçla, altyapı cevabını değerlendirmek için eğitilmiş vekil modeller kullanılmış ve model parametrelerindeki belirsizlikleri yönetmek için Monte Carlo simülasyonu uygulanmıştır. Vekil model geliştirilip, hasar varlığı olasılığı, olasılık yoğunluk fonksiyonuna dayalı olarak hesaplanmıştır. Ayrıca, olasılığa dayalı hasar tespiti (PBDD) çerçevesi, meta-sezgisel optimizasyon algoritmaları ve vekil modellerin birleşimiyle kurulmuştur. Çalışma, üç popüler meta-modelleme tekniğini (CFNN, LS-

SVM, Kriging) kullanarak, her bir algoritmanın performansını değerlendirmiştir. Sonuçlar, IGMN optimizasyon algoritması ile birlikte LS-SVM vekil modelinin, hasar tahmininde daha doğru ve verimli olduğunu ortaya koymuştur.

Mishra vd. (2019) çalışmalarında, yapısal sağlık izleme için yeni bir Karınca Aslanı Optimizasyon algoritması (ALO) sunmaktadır. ALO, yapısal hasar tespiti için kullanılabilecek bir popülasyon tabanlı meta-sezgisel optimizasyon algoritmasıdır. Algoritma, bir yapıdaki hasarın yerini ve şiddetini belirlemek için, titreşim verilerini kullanarak bir hedef fonksiyonu minimizasyonuna dayalıdır. Çalışmada, farklı hasar senaryolarını simüle etmek için, bir kiriş, düzlemsel çerçeve ve uzay çerçevesi modelleri kullanılmıştır. Simülasyon sonuçları, ALO'nun, diğer meta-sezgisel optimizasyon algoritmalarına (örneğin, PSO, GA, HS, ABC) kıyasla daha iyi performans gösterdiğini kanıtlamıştır. Özellikle çoklu hasar tespiti durumlarında, ALO, daha doğru sonuçlar üretmiş ve hesaplama maliyetini düşürmüştür. Yöntem, hem tekli hem de çoklu hasar tespiti için uygulanabilir bir yöntem olarak sunulmaktadır.

Ghannadi ve Kourehli (2019) çalışmalarında, Salp Sürü Algoritması (SSA) kullanarak çok katlı kesme çerçevelerdeki hasar tespiti yapılmıştır. İlk olarak, yapının sonlu eleman modeli (FEM) kullanılarak güncellenmesi sağlanmış, ardından hasarın yeri ve şiddeti SSA ile belirlenmiştir. Üç farklı deneysel örnekle yöntemin verimliliği test edilmiştir: üç katlı, beş katlı ve sekiz katlı kesme çerçevelerdeki tekil ve çoklu hasar durumları. Sonuçlar, SSA'nın çok katlı çerçevelerde hasarın yerini ve şiddetini doğru ve verimli bir şekilde belirlediğini göstermiştir. Ayrıca, SSA'nın, parçacık sürü optimizasyonu (PSO) algoritması ile karşılaştırıldığında daha yüksek doğruluk sağladığı bulunmuştur.

Ghannadi ve Kourehli (2019) bu çalışmada, yapısal hasar tespiti, modal güvence kriteri esnekliği ve doğal frekanstan oluşan bir terimden oluşan bir hedef fonksiyonun en aza indirilmesiyle gerçekleştirilen bir optimizasyon algoritması olan Güve Alevi Optimizasyonu (MFO) kullanılarak ele almışlardır. Önerilen yöntemin performansını göstermek için, kiriş ve çerçeve sistem olmak üzere iki sayısal örnek incelenmiştir. Ayrıca, Los Alamos Ulusal Laboratuvarı test yapısı da doğrulama amaçlı kullanılmıştır. Hem deneysel hem de sayısal örnekler için yapının mod özellikleri çıkarmak için MATLAB yazılımı kullanılmıştır. Mod şekilleri ve doğal frekanslar sayısal örneklerde gürültüyle kirletilmiştir. Ek olarak, bu çalışmada, klasik optimizasyon algoritmalarından birini olan parçacık sürü optimizasyonu, MFO ile kıyaslanmıştır. Kısacası, sayısal ve deneysel örneklerden elde edilen sonuçlar, önerilen yöntemin yapısal hasarı tespit etmede verimli olduğunu göstermiştir.

Ghannadi vd. (2020), hasar tespiti problemini çözmek için, Gri Kurt Optimizasyonu (GWO) ve Harris Şahin Optimizasyonu (HHO) algoritmaları kullanılmıştır. Bu algoritmanın başlıca özellikleri; küçük sayıda parametreye sahip olması, kolay uygulanabilmesi, esnek ve ölçeklenebilir olmasıdır. HHO, en zekî

kuşlardan birini olan Harris Şahinin koordinasyonlu av stratejilerinden esinlenerek geliştirilmiştir. Önerilen yöntemin performansı, bir 56 çubuklu kutup uzay çerçevesi modeli üzerinde yapılan sayısal simülasyonlar ile bir çelik çerçeve üzerinde yapılan deneysel çalışmalar ile değerlendirilmiştir. Elde edilen sonuçlar, GWO ve HHO'nun kombinasyonunun, sistem eşdeğer indirgeme ve genişletme yöntemi ile birlikte kullanıldığında, yapıların hasarlarının yeri ve şiddetinin belirlenmesi için güvenilir bir yaklaşım sağladığını göstermiştir.

Ghannadi ve Kourehli (2020), yapısal hasar tespitinde Çoklu Evren Optimizasyonu (MVO) algoritmasının etkinliğini incelemiş ve Sinüs Kosinüs Algoritması (SCA) ve HHO gibi diğer algoritmalarla karşılaştırmıştır. Çalışmada, hasar tespiti için Modal Assurance Criterion (MAC) ve Modified Total Modal Assurance Criterion (MTMAC) hedef fonksiyonları kullanılmıştır. MVO, bu hedef fonksiyonlarıyla birlikte yüksek doğruluk ve güvenilirlik sağlarken, özellikle MTMAC ile kombinasyonu, diğer algoritmalara göre en yüksek doğruluk oranını elde etmiştir. Elde edilen sonuçlar, MVO'nun yapısal hasar tespitinde etkili bir yöntem olduğunu ve mühendislik uygulamalarında daha güvenilir hasar tespiti sağladığını göstermektedir. Bu çalışma, MVO'nun gelecekteki araştırmalara temel oluşturacak yeni bir optimizasyon yaklaşımını sunduğu için literatüre önemli bir katkı sağlamaktadır.

Kaveh vd. (2021), meta-sezgisel optimizasyon algoritmaları kullanılarak yapısal hasar tespiti için yeni bir Sınır Stratejisi (BS) sunulmuştur. Sınır Stratejisi (BS), optimizasyon işlemi sırasında sağlıklı yapısal elemanların etkilerini yavaşça etkisiz hale getirmektedir. Bu sayede, optimizasyon yöntemi, geleneksel yöntemlere göre daha iyi optimum çözüme ulaşmaktadır. Geliştirilen BS, algoritmaların hem doğruluk hem de yakınsama hızını artırmaktadır. Yöntemin performansını değerlendirmek için, yeni bir hasara duyarlı maliyet fonksiyonu kullanılmıştır. Bu fonksiyon, yapısal titreşim verilerine dayanmaktadır. Fonksiyon, Karışık Çoban Optimizasyon Algoritması (SSOA) kullanılarak optimizasyon problemi çözümü için kullanılmıştır. Çalışmada, çerçeve, giriş ve düzlemsel çerçeve modelleri kullanılarak yapılan sayısal analizler sonuçları sunulmuştur. Elde edilen sonuçlar, BS'nin, gürültülü veriler dahil, farklı hasar senaryolarında, kullanılan optimizasyon algoritmalarının (TLBO, GWO, MFO) performansını artırdığını göstermiştir. Çalışmada sunulan BS stratejisi, yapısal hasar tespit problemlerini çözmek için kullanılabilen, geniş ölçekli yapılarda uygulanabilen etkili bir yaklaşım olarak sunulmaktadır.

Huang vd. (2021), yapılardaki hasarı tespit etmek için alt yapı yöntemi ve Geliştirilmiş Balina Optimizasyon Algoritması (LWOA)'nı kullanarak yeni bir yöntem önermiştir. İlk olarak, yapının global modeli alt yapılara bölünerek analiz verimliliği artırılmıştır. Ardından, LWOA algoritması ile hasarın şiddeti hesaplanmış ve Levy-fight yöntemi ile balina optimizasyonunun performansı iyileştirilmiştir. Eleman göreceli mod gerinim enerjisi değişimine dayalı bir hasar indeksiyle oluşturulan hedef fonksiyonu, yapısal hasarı tespit etmekte kullanılmıştır. Üç farklı örnek üzerinde yapılan sayısal

simülasyonlar, hasarlı elemanların mod gerinim enerjisinin belirgin şekilde değiştiğini, hasar tespiti verimliliğini artıran bu yöntemin doğru sonuçlar sağladığını göstermektedir. Elde edilen bulgular, yöntemlerin farklı yapı türlerinde yüksek doğrulukla hasar yerini ve şiddetini tespit edebileceğini kanıtlamaktadır.

Mohebian vd. (2022), çalışmada Görünür Parçacık Serisi Arama Algoritması'nın (VPSSA) geliştirilmesi ve uygulanmasını ele almışlardır. VPSSA, yapısal hasar tespitinde daha iyi sonuçlar elde etmek amacıyla geliştirilmiştir. Çalışmada, VPSSA'nın temel prensipleri açıklanmakta ve algoritmanın nasıl çalıştığı detaylandırılmaktadır. Algoritma, PSO ve görünürlük grafiği analizi gibi teknikleri birleştirerek, hasar tespit sürecini optimize etmektedir. Uygulanan yöntem, yapısal dinamik verilerin analizine dayanmakta ve bu verilerden elde edilen bilgileri kullanarak hasar yerini ve büyüklüğünü belirlemektedir. Çalışma, çeşitli simülasyonlar ve deneysel verilerle desteklenmiş olup, VPSSA'nın etkinliğini göstermektedir. Sonuçlar, VPSSA'nın geleneksel yöntemlere göre daha yüksek doğruluk oranları ve daha hızlı hesaplama süreleri sunduğunu ortaya koymaktadır. Ayrıca, algoritmanın farklı yapısal sistemlerde uygulanabilirliği test edilmiş ve olumlu sonuçlar elde edilmiştir. VPSSA'nın, mühendislik uygulamalarında kullanılabilirliği, gelecekteki araştırmalar için yeni bir yol açmaktadır.

Chen vd. (2022) çalışmada, Şempanze Optimizasyon Algoritması (ChOA) ve Balina Optimizasyon Algoritması (WOA) birleştirilerek yapısal hasar tespiti için yeni bir hibrit algoritma olan Balina-Şempanze Optimizasyon Algoritması (W-ChOA) önerilmiştir. Algoritmanın hassasiyetini artırmak için, ChOA'nın ilk popülasyon başlatma aşamasında Sobol dizisi kullanılmıştır. Sobol dizisi, popülasyonun çözüm uzayını eşit olarak doldurmasını sağlamaktadır. Çalışmada ayrıca, ChOA algoritmasının yerel arama yeteneğini geliştirmek için, WOA'nın kabarcık ağ avlama mekanizması ve rastgele arama mekanizması ChOA'ya entegre edilmiştir. Algoritmanın geçerliliği ve uygulanabilirliği, iki katlı bir yapısal çerçeve ve basit destekli bir kiriş modeli ile gösterilmiştir. Simülasyon sonuçları, W-ChOA'nın, özellikle çoklu hasar tespiti durumlarında, ChOA'ya kıyasla önemli ölçüde daha iyi performans gösterdiğini göstermiştir. W-ChOA, yanlış teşhisleri azaltmada ve hesaplama verimliliğini artırmada iyi performans göstermiş, bu nedenle yapısal hasar tespiti için tercih edilen bir seçenektir.

Sarjamei vd. (2022), yapıların hasar tespiti için yeni bir ters optimizasyon problemi çözümü olarak Altın Hız Optimizasyon (GRO) algoritması sunulmuştur. Hasar tespiti, mod şekilleri ve doğal frekanslar kombinasyonu ile tanımlanmıştır. GRO, insan düşünme ve karar verme gücünden yararlanan yapay zeka tabanlı bir sistemdir. GRO algoritması, MATLAB ortamında programlanmıştır. Algoritmanın geçerliliği ve doğruluğu, 10, 25, 200, 272, ve 582 çubuklu köprüler üzerinde yapılan simülasyonlar sonucu doğrulanmıştır. Çalışmada, GRO ile MCSS, PSO ve TLBO algoritmalarının karşılaştırılması yapılmış, GRO'nun küresel optimumu daha hızlı ve doğru bir şekilde bulduğu sonucu çıkarılmıştır. GRO algoritması kullanılarak yapısal hasar tespiti hem güvenilir hem de stabildir.

Khanahmadi vd. (2023) inşaat projelerinde yaygın olarak kullanılan 3D sandviç panellerde hasar tespiti için yeni bir yöntem önermektedir. Yöntem, modal dinamik verileri kullanarak hasarlı bölgeleri tanımlamak amacıyla bir Düzensizlik Tespit İndeksi (IDI) geliştirmektedir. IDI, dalgacık analizi ile elde edilen yatay, dikey ve çapraz detay katsayılarına dayanmaktadır. Çalışmanın amacı, hasar şiddeti arttıkça IDI değerlerinin de artması ve hasarlı bölgelerin kolayca tespit edilmesidir. Literatürde, 3D sandviç panellerde hasar tespiti için bu tür bir indeksin kullanılmadığı belirtilmektedir. Sonuçlar, IDI'nin hasar yerlerini belirlemede etkili olduğunu ve hasar şiddetinin artmasıyla birlikte IDI değerlerinin yükseldiğini göstermektedir. Bu çalışma, yapısal sağlık izleme alanında önemli bir katkı sağlamaktadır.

Li vd. (2024) barajlardaki hasar tespitini geliştirmek için entegre bir vekil model destekli ve iyileştirilmiş Termite Yaşam Döngüsü Optimizasyon (TLCO) algoritması önermektedir. Yapısal hasar tanımlama (SDI) yöntemleri, yapısal sağlık izleme alanında kritik bir öneme sahiptir. Önerilen yöntem, maliyet etkin vekil modelleme teknikleri ve güçlü küresel optimizasyon yeteneklerine sahip sürü zekası algoritmalarını birleştirerek, hasar tespitinde yüksek doğruluk ve verimlilik sağlamaktadır. Çalışmanın sonucunda, önerilen yöntemle geleneksel SDI yöntemlerine göre hesaplama verimliliğinin 100 kat artırıldığını göstermektedir. Ayrıca, farklı vekil modellerin doğruluğu ile SDI yönteminin etkinliği arasında doğrudan bir ilişki olmadığı bulunmuştur. Bu çalışma, büyük ölçekli yapılar için hasar tanımlama konusunda umut verici bir araç sunmakta ve literatürdeki mevcut yöntemlere önemli bir katkı sağlamaktadır.

Farhadi vd. (2024) truss yapıların hasar tespiti için optimize edilmiş grup veri işleme (GMDH) vekil modeli ve meta-sezgisel algoritmalar kullanarak iki aşamalı bir hasar tespit yaklaşımı önermektedir. İlk aşamada, hasar olasılığı yüksek bölgeler, kalıntı kuvvet vektörü (RFV) ile belirlenmektedir. İkinci aşamada ise, PSO, DE ve geliştirilmiş çarpışan cisimler optimizasyonu (ECBO) gibi üç farklı optimizasyon algoritması kullanılarak hasarın boyutu değerlendirilmektedir. Çalışmanın amacı, büyük ölçekli yapılar için hasar tespit süreçlerinin hesaplama süresini azaltmak ve doğruluğunu artırmaktır. Sonuçlar, önerilen PSO tabanlı GMDH modelinin, geleneksel yöntemlere göre önemli ölçüde daha hızlı ve doğru sonuçlar sağladığını göstermektedir. Bu çalışma, yapısal sağlık izleme alanında önemli bir katkı sunmaktadır.

D. Dinh-Cong ve T. Nguyen-Thoi (2024) yarı rijit çerçevelerde hem yapısal bağlantı hem de eleman hasarını aynı anda tanımlamak için genişletilmiş Frekans Tepkisi Fonksiyonu (FRF) verilerini kullanan etkili bir model güncelleme yöntemi önermektedir. Yazarlar, hasar tanımlama sürecini optimize etmek için Kaos Oyun Optimizasyonu (CGO) algoritmasını benimsemişlerdir. Çalışmanın amacı, eksik ölçülen FRF verileri ile hasar yerlerini ve şiddetini doğru bir şekilde belirlemektir. Literatürde, çoğu çalışma yalnızca eleman hasarına odaklanırken, bu çalışma hem bağlantı hem de eleman hasarını aynı anda ele alarak önemli bir boşluğu doldurmaktadır. Sonuçlar, önerilen yöntemin çeşitli hasar senaryolarında yüksek doğruluk sağladığını ve CGO algoritmasının diğer

yeni geliştirilmiş algoritmalarla göre daha iyi performans gösterdiğini ortaya koymaktadır. Bu bulgular, yarı rijit çerçevelerin güvenliği ve dayanıklılığı için önemli bir katkı sunmaktadır.

Zhang vd. (2024), çalışmasında, yapısal hasar tespitine yönelik yeni bir ters analiz yaklaşımı geliştirdiler. Önerilen yöntem, çok hedefli parçacık sürüsü optimizasyonu (MOPSO) ile öğrenme otomalarının (LA) entegrasyonunu kullanarak, hasar senaryolarını doğru bir şekilde tespit etmeyi amaçlamaktadır. Çalışmanın amacı, yüksek boyutlu ve çok modlu hedef fonksiyonlar içeren optimizasyon problemlerinin çözümünde yerel optimum tuzaklarına düşmeden, küçük ama çeşitli bir çözüm seti elde etmektir. Literatürde, mevcut yöntemlerin çoğu tek hedefli optimizasyon tekniklerine dayanmakta ve bu da sınırlı çözümler sunmaktadır. Önerilen yaklaşım, hem sayısal simülasyonlar hem de deneysel verilerle doğrulanmış ve yüksek doğrulukla hasar tespiti gerçekleştirmiştir. Sonuçlar, bu yöntemlerin yapısal sağlık izleme alanında önemli bir katkı sağladığını göstermektedir.

Hassani vd. (2024), çalışmalarında, gürültülü giriş sinyallerinin yol açtığı zorluklar nedeniyle yapısal sağlık izleme sistemlerinin duyarlılığına odaklanmaktadır. Çalışma, dışarıdan gürültü ve ölçüm hatalarının yol açtığı aykırı değerlerin etkisini azaltan yeni bir model güncelleme tabanlı optimizasyon algoritması önermektedir. Yöntem, ilk olarak ölçülen verileri düzgünleştirmek için Gauss yumuşatma tekniği kullanır ve ardından Johansen eşbütünleşmesi, ham veri füzyonu yapmak için kullanılır. Daha sonra, bu çalışma, karşılıklı bilgi kavramına dayanan bir optimizasyon algoritması olan Geliştirilmiş Sürüngen Arama Algoritmasını (IRSA) tanıtır ve daha sonra hasar indekslerini optimize etmek için kullanılır. Sayısal ve deneysel çalışmalar, IRSA'nın diğer mevcut optimizasyon algoritmalarına göre önemli ölçüde daha hızlı bir yakınsama oranı ve artan doğruluk sağladığını göstermektedir. Ayrıca, önerilen metot, literatürdeki diğer yöntemlerle karşılaştırılmıştır ve üstün performans göstermiştir. Bu metot, yapısal sağlık izleme alanındaki gürültülü sinyallerle ilişkili zorlukları gidermeye yönelik kapsamlı bir çözüm sunmaktadır.

Makiabadi vd. (2024), hareketli yükler altında köprü yapıları için hasar tespiti amacıyla, Geliştirilmiş Simbiyotik Organizmalar Arama-Benzetilmiş Tavlama (HESOS-SA) algoritmasını birleştiren hibrit bir yöntem sunmaktadır. Yöntem, yapısal hasarı tespit etmek için üç farklı noktadan ölçülen hareketli yükler nedeniyle statik sapmaları kullanmaktadır. Çalışma ayrıca, hasar tespiti için yeni bir kriter olan "Sapma Etki Hattı Göstergesi"ni (DILI) sunmaktadır. Önerilen yöntemin etkinliğini değerlendirmek için, basit destekli bir kiriş ve iki farklı boyutlarda düzlemsel kafes köprü olmak üzere üç referans yapı kullanılmıştır. Hesaplamalı sonuçlar, hem SOS hem de HESOS-SA algoritmalarının, gürültüsüz veriler için DILI kriterini kullanarak hem hasarın yerini hem de şiddetini doğru bir şekilde tespit edebildiğini göstermektedir. Gürültülü veriler için HESOS-SA algoritmasının, SOS algoritmasına göre hasar tespitinde daha sağlam bir performans sergilediği kanıtlanmıştır.

Tran vd. (2024), kiriş köprülerdeki hasar tespiti için Ortogonal Çözümleme (OD) ile SSA birleştiren yeni bir yaklaşım sunmaktadır. Ancak, yalnızca lider salp'ı kullanarak en iyi çözümleri aramasından dolayı, salp sürüsünün sömürme ve keşif yetenekleri arasında bir dengesizlik vardır. Makalede, OD'nin lider salp'ın arama kabiliyetini artırmak için kullanılması önerilmektedir. OD, arama alanını daraltmada ve sonraki yinelemelerde yüksek kaliteli salp elemanlarını seçmede önemli bir rol oynar. Önerilen metot, 12 karşılaştırma fonksiyonu ve Cua Rao Köprüsü ile ilgili bir dizi hasar senaryosu kullanılarak test edilmiştir. Sonuçlar, OD kullanıldığında, tek bir hasar senaryosu için doğruluk sapma oranının yalnızca %1,412 olduğunu ve SSA'da bu oran %2,224 olduğunu göstermiştir. Hasar gören eleman sayısı arttıkça, SSA'nın hasar gören elemanları tespit etmedeki hatası %99,667'ye yükselirken, OD kullanımıyla oluşan hatalar %10'un altında kalmıştır. Buna ek olarak, her iki senaryoda da hesaplama hızı yaklaşık %40 oranında iyileştirilmiştir.

Literatür özetinde görüldüğü gibi, meta-sezgisel yöntemler, yapısal hasar tespitinde etkin ve güvenilir sonuçlar sunarak mühendislik uygulamalarında önemli bir rol oynamaktadır. Ancak, bu yöntemlerin stokastik doğası nedeniyle, farklı algoritmalar arasında performans açısından belirgin farklılıklar gözlemlenmektedir. Her bir meta-sezgisel yöntem, belirli bir problem için optimize edilmiş olsa da, yeni bir yöntem geliştirildiğinde, bu yöntemin performansı da değişkenlik gösterebilir. Bu durum, yapısal hasar tespitinde kullanılan yöntemlerin sürekli olarak geliştirilmesi ve optimize edilmesi gerektiğini ortaya koymaktadır. Sonuç olarak, her yeni yaklaşım, mevcut yöntemlerin etkinliğini artırma potansiyeline sahip olabilir, bu da araştırma ve uygulama alanında sürekli bir yenilik arayışını teşvik etmektedir.

## 2.2. EEFO'nun Çeşitli Problemlerdeki Performansının İncelenmesi

Yapısal hasar tespitinde meta-sezgisel yöntemlerin kullanılması oldukça popülerdir. Literatürde birçok çalışma yer alsa da, karmaşık problemleri çözmek için, her zaman daha etkili ve daha hızlı yöntem arayışı olmaktadır. Bu anlamda, sürekli olarak yeni yöntemler ortaya çıkmaktadır. Tez çalışmasında 2024 yılında Zhao vd. (2024) tarafından geliştirilmiş topluluk tabanlı yeni bir meta-sezgisel optimizasyon yöntemin, yapısal hasar tespiti üzerindeki etkinliği üzerinde durulacaktır. EEFO yeni tanıtılmış olmasına rağmen birçok matematiksel problemde incelenmiş ve algoritma farklı problemleri çözmek için test edilmiştir.

EEFO algoritması üzerine gerçekleştirilen bazı çalışmalar aşağıda anlatılacak olup, bu yenilikçi optimizasyon yönteminin, geleneksel meta-sezgisel algoritmalara kıyasla daha hızlı ve hassas sonuçlar sunduğunu ve özellikle çok karmaşık yapıların hasar tespiti gibi zorlu mühendislik problemlerinde daha etkili olduğunu ortaya koymaktadır.

Mohammed vd. (2024) bu çalışmada, karmaşık mühendislik sorunlarını çözmek için Kuantum Uzun Kısa Süreli Bellek (QLSTM) modeli ile Modifiye Elektrikli Yılan Balığı Avlanma Optimizasyon (MEEFO) algoritmasını entegre eden yeni bir

yaklaşım sunulmuştur. QLSTM, derin öğrenme modellerinde kullanılan, karmaşık veri dizilerini işleme konusunda etkili olan bir modeldir. Ancak, QLSTM modelinin performansını etkileyen, parametrelerin optimal olarak belirlenmesi, zamana bağlı ve NP-problem niteliğindedir. Bu sorunu çözmek için, geleneksel EEFO algoritması üzerinde modifikasyonlar yapılarak, MEEFO algoritması geliştirilmiştir. MEEFO, geleneksel EEFO'nun arama kapasitesini artırmak için Üçgensel Mutasyon Operatörünü kullanmaktadır. Bu çalışmanın en önemli katkısı, inşaat ve yapı mühendisliği ile ilgili beş farklı gerçek dünya veri kümesi kullanarak, QLSTM modelinin performansını artırmak için MEEFO'nun kullanılmasını önermesi ve bu modelin etkililiğini göstermesidir. Sonuçlar, MEEFO'nun QLSTM modelinin performansını önemli ölçüde artırdığını göstermiştir. Çalışma ayrıca, QLSTM algoritması ile ilgili literatür taraması da sunulmuştur. Bu tarama, QLSTM algoritmasının, klasik LSTM modelinden farklı olarak kuantum hesaplama prensiplerinden faydalanarak, veri işleme konusunda daha etkili olduğunu göstermektedir. QLSTM modelinin, yapısal hasar tespiti, beton yapıların dayanımının öngörülmesi, inşaat yönetimi ve yol yapımları gibi çeşitli mühendislik alanlarında kullanılabileceği vurgulanmaktadır.

Nourhan vd. (2024), Elektrikli Yılan Balığı Avlanma Optimizasyon (EEFO) algoritması kullanılarak, şebekeye bağlı fotovoltaiik (PV) santrallerinin düşük voltajda kalabilme (LVRT) kapasitesinin nasıl iyileştirilebileceği araştırılmıştır. Çalışmada, PV dizilerinin, DC-DC boost dönüştürücü, DC bağlantı kapasitörü, şebeke bağlantısı invertör ve üç fazlı adım-yukarı trafo aracılığıyla şebekeye bağlandığı bir sistem kullanılmıştır. DC-DC boost dönüştürücü devresi, artımlı iletkenlik yöntemi kullanılarak, maksimum güç noktası izleme (MPPT) işlemi için tasarlanmıştır. Şebekeye bağlantı invertör devresi, DC bağlantı voltajını ve PCC'deki terminal voltajını düzenlemek için kademeli kontrol döngüsü kullanmaktadır. EEFO algoritmasına dayalı, oransal-integral (PI) kontrolör, elektronik devrelerin gücünü düzenlemek için kullanılmıştır. EEFO'nun, genetik ve parçacık sürüsü optimizasyon algoritmalarından daha etkili bir kontrol stratejisi sunduğu da çalışmada gösterilmiştir. Çalışmada, sistemin simetrik ve asimetrik arızalara ve kalıcı arızalar nedeniyle başarısız devre kesici yeniden bağlantılarına maruz bırakılarak incelenmiştir. Sonuç olarak, EEFO tabanlı PI kontrol sisteminin, PV sisteminin kararlılığını ve güvenilirliğini artırdığı ve LVRT kapasitesini iyileştirdiği gözlemlenmiştir.

Maurya vd. (2024) çalışmalarında, dağıtık enerji kaynaklarının (DG) optimal yerleştirilmesi ve şebeke yeniden yapılandırılması için Elektrik Yılan Balığı Avlanma Optimizasyon (EEFO) algoritmasını sunmuştur. Çalışmada, EEFO algoritması, enerji faktörünü kullanarak keşif ve sömürme aşamalarını dengeleyerek optimizasyon sürecini basitleştirir. Algoritmanın etkinliği, 33-node ve 84-node radyal dağıtım şebekelerinde (RDN) farklı DG güç faktörü senaryoları altında test edilmiştir. Sonuçlar, DG'lerin optimal yerleştirilmesi ve şebeke yeniden yapılandırılması ile voltaj sapması, aktif güç kaybı ve reaktif güç kaybının önemli ölçüde azaldığını göstermiştir. Ayrıca, EEFO algoritması, Zebra Optimizasyon, Balina Optimizasyon ve Güve Alevi Optimizasyon

algoritmalarıyla karşılaştırıldığında, DG yerleştirme ve şebeke yeniden yapılandırma konusunda daha etkili ve verimli bulunmuştur.

Bu tez çalışması, yapısal hasar tespitinde Elektrikli Yılan Balığı Avlanma Optimizasyonu (EEFO) algoritmasının uygulanmasını incelemektedir. Literatürde, EEFO'nun yapısal hasar tespitinde kullanılmadığı görülmektedir. Bu çalışma, bu yenilikçi algoritmanın farklı bir alanda test edilmesi açısından önemli bir katkı sunmaktadır. Yapısal hasar tespitinde, mevcut optimizasyon yöntemleri genellikle gürültüye karşı dayanıklılık, doğruluk ve hesaplama maliyeti gibi kriterler üzerinden değerlendirilmiştir. EEFO, meta-sezgisel yöntemler arasında yer alarak, daha hızlı ve etkili sonuçlar sunma potansiyeline sahiptir.

### **2.3. Literatürdeki Eğilimler ve Özgünlük Değerlendirmesi**

Kaynak taraması, yapısal hasar tespitinde meta-sezgisel yöntemlerin kullanımı ve Elektrikli Yılan Balığı Avlanma Optimizasyonu (EEFO) algoritmasının çeşitli problemlerdeki performansı olmak üzere iki ana başlık altında incelenmiştir. İlk başlıkta, yapısal hasar tespitinin güvenlik, sürdürülebilirlik ve maliyet etkinliği açısından önemi vurgulanmakta; optimizasyon yöntemlerinin, hasar parametrelerini doğru bir şekilde belirlemek için nasıl kullanıldığı açıklanmaktadır. İkinci başlıkta ise, EEFO algoritmasının, geleneksel optimizasyon yöntemlerine göre daha hızlı ve etkili sonuçlar sunduğu, karmaşık mühendislik problemlerinde sağladığı avantajlar ve literatürdeki diğer yöntemlerle karşılaştırmaları ele alınmaktadır. Bu iki başlık, yapısal hasar tespitinde yenilikçi ve etkili çözümler arayışının önemini ortaya koymakta ve EEFO'nun bu alandaki potansiyelini vurgulamaktadır. Bu çalışma, EEFO'nun yapısal hasar tespitindeki etkinliğini ortaya koyarak, mühendislik ve inşaat sektörlerinde güvenlik ve sürdürülebilirlik açısından önemli bir yenilik sağlayacaktır. Sonuç olarak, bu tez çalışması, yapısal hasar tespitinde EEFO'nun uygulanabilirliğini değerlendirerek, literatüre yeni bir bakış açısı kazandırmayı hedeflemektedir.

### 3. MATERYAL VE METOT

Tezin bu kısmında, yapısal hasar tespiti sürecinde model tabanlı analiz ve optimizasyon yöntemi detaylı bir şekilde açıklanmaktadır. İlk olarak, yapının dinamik davranışlarını temsil etmek için özdeğer ve özvektör problemleri çözülerek doğal frekanslar ve mod şekillerin nasıl elde edileceği anlatılmaktadır. Çalışmada, belirli bir örnek yapı modeli üzerinden yapılan analizler, kullanılan matematiksel formülasyonlarla açıklanacak, modelin MATLAB ortamında nasıl simüle edildiği ve çözümlendiği ayrıntılarıyla anlatılacaktır. İkinci olarak, yapısal hasarın tespiti için optimizasyon probleminin nasıl oluşturulduğu, modelin doğruluğunu değerlendirmek için hangi hata fonksiyonlarının kullanıldığı detaylı bir şekilde açıklanacaktır. Son olarak yapısal hasarın tespiti için kullanılan optimizasyon yöntemi olan Elektrikli Yılan Balığı Avlanma Optimizasyonu (EEFO) algoritması detaylandırılacaktır. Bu üç ana aşama, bu çalışmanın hedeflerine ulaşmasını sağlayacak temel araçları sunmaktadır.

#### 3.1. Model Tabanlı Yapısal Analiz

##### 3.1.1. Teori

Titreşim denklemi bir sistemin zamanla değişen hareketini tanımlayan bir diferansiyel denklemdir. Bu yüzden özdeğer özvektör problemine benzetilerek yapının hareketi incelenebilir. Titreşim denklemi, sistemin kütlesi, sönümleme, rijitlik ve dış kuvvetler faktörleri dikkate alarak yazılabilir. Bu çalışmada problemi basitleştirmek adına denklem (3.1)'de ifade edildiği gibi sönümsüz serbest titreşim denklemi kullanılmaktadır.

$$m\ddot{x} + kx = 0 \quad (3.1)$$

Denklem (3.1), ikinci dereceden homojen diferansiyel denklem olduğu için denklemin analitik çözümü denklem (3.2)'de yer almaktadır. Tipik olarak harmonik hareketi ifade eden denklem, aynı zamanda titreşim denkleminin deplasman fonksiyonudur. Burada  $X$  genlik,  $\omega$  açısal frekans,  $\phi$  sapma açısını ifade etmektedir. Bu fonksiyonun iki kere türevi alındığında denklem (3.2)'deki gibi ivmesi elde edilir.

$$x = X \cdot \sin(\omega t + \phi); \quad \ddot{x} = -X \cdot \omega^2 \sin(\omega t + \phi) \quad (3.2)$$

Buradaki  $x$  ve  $\ddot{x}$  değerleri denklem (3.1)'de yerine yazılıp, ortak paranteze alındığında aşağıdaki denklem elde edilmektedir.

$$-m \cdot X \cdot \omega^2 \sin(\omega t + \phi) + k \cdot X \sin(\omega t + \phi) = 0 \quad (3.3)$$

Denklem (3.3)'teki  $\sin(\omega t + \phi)$  ifadesi ortak paranteze alındığında denklem (3.4)'teki forma dönüşmektedir.

$$(k - m \cdot \omega^2) \cdot X \sin(\omega t + \phi) = 0 \quad (3.4)$$

Denklem (3.4)'teki  $X \cdot \sin(\omega t + \phi)$  ifadesi, sıfır olması söz konusu olamaz. Bu nedenle, denklem (3.5)'teki gibi aşağıdaki terimler sıfıra eşit olmalıdır.

$$(k - m \cdot \omega^2) \cdot X = 0 \quad (3.5)$$

Denklem (3.5) tek serbestlik derecesine sahip bir sistemin titreşim denklemidir. Birden fazla serbestlik derecesine sahip sistemlerin titreşim denklemleri denklem (3.6)'daki gibi lineer denklem sistemine dönüşmektedir.

$$([K] - [M] \cdot \omega^2) \vec{X} = 0 \quad (3.6)$$

Denklem (3.6)'da yer alan  $[K]$  ve  $[M]$  sistemin rijitlik matrisi ve sistemin kütle matrisidir. Özdeğer özvektör denkleminde geçiş yapmak için denklem (3.6)'daki  $\vec{X}$  ifadesi parantez içine dağıtılır ve  $[M]$  ifadesi karşı tarafa geçerek denklemin son hali denklem (3.7)'deki gibi olmaktadır.

$$[M]^{-1} \cdot [K] \cdot \vec{X} = \omega^2 \cdot \vec{X} \quad (3.7)$$

Kütle matrisinin tersiyle rijitlik matrisinin çarpımına  $[A]$  matrisi dersek ( $[A] = [M]^{-1} \cdot [K]$ ), Frekans değerinin karesine de  $\lambda$  (lamda) dersek ( $\omega^2 = \lambda$ ), ve bu ifadeler denklem (3.6)'da yerine yazılırsa denklem (3.8) elde edilir.

$$([A] - [I]\lambda) \cdot \vec{X} = 0 \quad (3.8)$$

Burada  $[I]$  ifadesi  $[M]^{-1}$  ile  $[M]$  ifadelerinin çarpımının sonucu olan birim matristir. Denklem (3.8), yukarıdaki hali ile bir özdeğer özvektör problemine dönüşmüştür. Denklem (3.8)'in çözümünden elde edilen  $\vec{X}$  ve  $\lambda$  sırasıyla özvektör ve özdeğeri ifade etmektedir.

Yapı analizinde özvektörler mod şekillerini, özdeğerler frekans değerlerini ifade eder. Yapıda oluşan hasar yapının frekans ve mod şekilleri gibi modal parametrelerini değiştirir. Değişen bu modal parametreler ise denklem (3.6) görüldüğü gibi sadece yapının rijitliğine ve kütlelerine bağlıdır. Yapıya hasar iki şekilde verilebilir. İlki rijitlik matrisinin değişmesi, ikinci seçenek ise kütle matrisini değiştirmektir. Bu matrislerden birinin değişmesi yapıyı hasarlı gösterecek, yapının frekans ve mod şekilleri de buna bağlı olarak değişecektir. Bu çalışmada hasar elastisite modülünün azaltılmasıyla simüle edilmektedir. 0 ile 1 aralığında bir değerle, sağlam elemanın elastisite modülü çarpılarak, elemanın hasarlı olması sağlanır. Özdeğer özvektör problemi MATLAB programındaki 'eig' komutuyla çözülmektedir. Kafes sistem özelliklerinin programlanması sonraki bölümde (Bölüm 3.1.2.) yapılmaktadır.

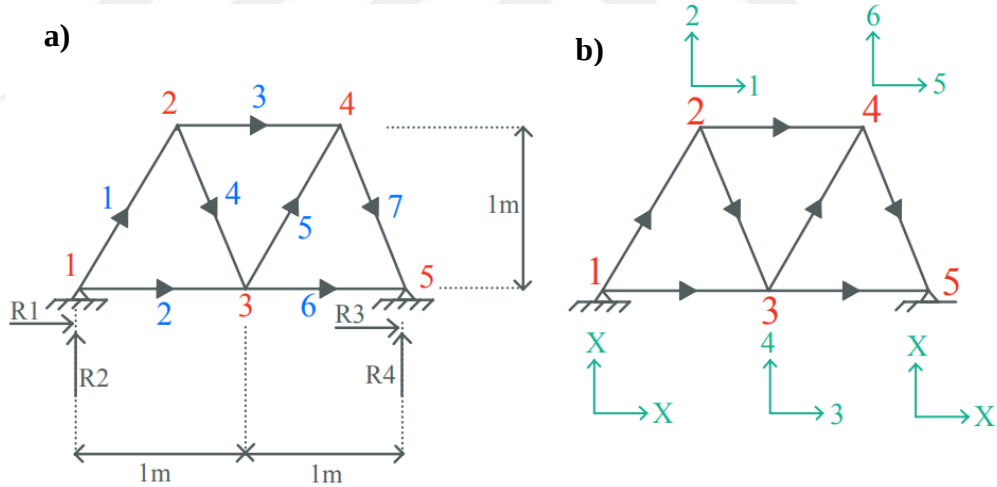
### 3.1.2. Yapısal hasar tespitinde kullanılan test örneğinin MATLAB dilinde programlanması

Bu bölümde, yapısal hasar tespiti için geliştirilen MATLAB kodunun nasıl oluşturulduğu ve bir modelin programa nasıl uyarlandığı açıklanmaktadır. Bunun daha iyi anlaşılması için 7 elemandan ve 5 düğümden oluşan bir kafes sistemi örneği üzerinden açıklamalar yapılacaktır. Test örneği olarak kullanılacak bu kafes modelinin sistem özellikleri matematiksel ifadelerle dönüştürülerek, hasar tespiti için MATLAB programına hazır hale getirilir.

Programın temel olarak üç ana aşaması vardır. İlk aşama, girdilerin tanımlanması ve serbestlik derecelerinin belirlenmesini içerir. İkinci aşama, rijitlik ve kütle matrislerinin oluşturulmasıdır. Son olarak; özdeğer özvektör probleminin çözülerek sistemin frekans ve mod şekillerinin elde edilir.

### 3.1.3. Test Örneğindeki Düğüm Noktalarının Serbestliklerinin Belirlenmesi

Bu ilk aşamada, yapısal hasar tespitinde kullanılan sonlu elemanlar yöntemi (FEM) kullanarak kafes sisteminin modellenmesinde düğüm noktaları ve reaksiyon kuvvetlerinin numaralandırılması ile serbestlik derecelerinin belirlenmesi aşağıdaki şekil 3.1.'de açıklanmaktadır.



**Şekil 3.1.** Örnek sistemin sonlu elemanlar modeli; **a)** kafes sistemin eleman, düğüm noktaları ve reaksiyon kuvvet numaralandırılması; **b)** serbestlik derecelerinin gösterimi

Şekil 3.1 (a), kafes sisteminin eleman, düğüm noktaları ve reaksiyon kuvvet numaralandırılmasını gösterirken, (b) şekli ise bu sistemin her düğüm noktasında yatay (X) ve dikey (Y) yönlerindeki serbestlik derecelerinin nasıl gösterildiğini açıklamaktadır. Bu gösterim, MATLAB ortamında kafes sisteminin modelinin kurulması ve daha sonra hasar tespiti için optimizasyon işleminin yapılması için gereklidir.

MATLAB'ta kullanılan bu test örneği için hazırlanması gereken girdilere ihtiyaç vardır. Bunlar aşağıdaki gibi sırasıyla yazılmıştır. Tüm elemanların elastisite modülü ve

kesit alanları sırasıyla  $1kN/m^2$  ve  $1 m^2$  olarak seçilmiştir. Son düzenlemelerle birlikte, aşağıda hazırlanmış olan veriler MATLAB koduna eklenir.

- Elemanların başlangıç noktaları  $M1=[1,1,2,2,3,3,4]$
- Elemanların bitiş noktaları  $M2=[2,3,4,3,4,5,5,]$
- Düğümlerin x koordinatları  $X=[0,0.5,1,1.5,2]$
- Düğümlerin y koordinatları  $Y=[0,1,0,1,0]$
- Reaksiyon kuvvetlerinin düğüm noktaları  $RJ=[1,1,5,5]$
- Reaksiyon kuvvetlerinin yönleri (X:0,Y:1)  $RD=[0,1,0,1]$
- Düğümlere etki eden kütleler  $Mass=[10,10,10,10,10]$

Yukarıda belirtildiği şekilde girdiler tanımlandıktan sonra sistemin serbestlik derecesini saklamak için  $DOF$  matrisi oluşturulur (denklem (3.9)). Denklem (3.9)'da yer alan  $DOF$  sistemin serbestlik derecesini ifade etmektedir.  $DOF$  matrisindeki değerler her bir düğüm noktasının x ve y yönündeki serbestliklerini gösterir (1.sütun x, 2. Sütun y). Eğer düğümde serbestlik tanımlanmamışsa (mesnet bulunuyorsa)  $DOF$  matrisinde o serbestlik derecesine 0 atanır.

$$DOF = \begin{bmatrix} 0 & 0 \\ 1 & 2 \\ 3 & 4 \\ 5 & 6 \\ 0 & 0 \end{bmatrix} \quad (3.9)$$

$DOF$  matrisinin belirlenmesinden sonra her bir serbestlik derecesine atanan kütleleri tanımlayan kütle matrisi (*Massmatrix*) denklem (3.10)'deki gibi oluşturulur. SAP2000'de "joint mass" (düğüm kütlesi), her düğüm noktasındaki kütleyi ifade eder ve yapıların dinamik analizlerinde kritik öneme sahiptir. Kafes sistemlerde, düğüm kütlesi yapının titreşim modlarını, doğal frekanslarını ve dinamik yanıtını belirlemede kullanılır. Ayrıca, dış etkiler (deprem, rüzgar vb.) altında yapının nasıl hareket edeceğini ve bu yüklerin nasıl dağıldığını etkiler. Yapının toplam kütlesi, düğüm noktalarında toplanmış gibi kabul edilerek dinamik analizlerde dikkate alınır. Düğüm kütlesinin doğru bir şekilde atanması, sistemin genel davranışının doğru bir şekilde hesaplanabilmesi için gereklidir.

$$Massmatrix = \begin{bmatrix} 10 & 0 & 0 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 & 0 & 0 \\ 0 & 0 & 10 & 0 & 0 & 0 \\ 0 & 0 & 0 & 10 & 0 & 0 \\ 0 & 0 & 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 0 & 0 & 10 \end{bmatrix} \quad (3.10)$$

$DOF$  ve *Massmatrix* oluşturulduktan sonra her bir elemanın hasar oranını belirten 'hasar' vektörü tanımlanır. 'hasar' hasarlı sayılacak elemanların elastisite modülünü hasar oranında azaltılarak saklanmaktadır. *Massmatrix* daha sonra bölüm 3.1.5.'de frekans ve mod şekillerinin bulunması sırasında kullanılmaktadır.

### 3.1.4. Yapı Elemanlarının Kütle ve Rijitlik Matrislerinin Oluşturulması

Bu aşamada, kafes sisteminin rijitlik ve kütle matrislerinin oluşturulması detaylı olarak açıklanmaktadır. Bu işlem, her bir elemanın özelliklerini (elastisite modülü, kesit alanı ve boyu) kullanarak lokal rijitlik matrislerinin hesaplanması ve daha sonra, bu lokal matrislerin dönüşüm matrisleri aracılığıyla global rijitlik matrisine dönüştürülür. Bu dönüşüm, her elemanın koordinat sistemindeki yönünü dikkate alarak yapılır. Elde edilen global rijitlik ve kütle matrisleri, daha sonra özdeğer problemi çözülerek yapısal hasar tespiti optimizasyonu için kullanılır.

Bu aşamada ilk olarak her eleman için lokal rijitlik matrisleri ( $kl$ ) denklem (3.11)'deki gibi oluşturulur.

$$kl = \begin{bmatrix} EA/L & -EA/L \\ -EA/L & EA/L \end{bmatrix} \quad (3.11)$$

Denklem (3.11)'de yer  $E, A$  ve  $L$  sırasıyla, elemanın elastisite modülünü, alanını ve uzunluğunu ifade etmektedir. Elemanların lokal rijitlik matrisleri tanımlandıktan sonra, dönüşüm matrisleri oluşturulur.

$$t = \begin{bmatrix} c & s & 0 & 0 \\ 0 & 0 & c & s \end{bmatrix} \quad (3.12)$$

Burada  $c$  ve  $s$  ifadeleri sırasıyla elemanın  $+x$  eksenine yaptığı açının kosinüs ve sinüs değerleridir. Elemanların lokal rijitlik ve dönüşüm matrisleri oluştuktan sonra denklem (3.13)'deki ifadeye göre elemanların aşağıdaki formülle lokal rijitlik matrisi global rijitlik matrisine dönüştürülür.

$$kg = t^T \times kl \times t \quad (3.13)$$

Elemanların global rijitlik matrislerini oluşturulduktan sonra eleman uçlarındaki serbestlik derecelerine göre global rijitlik matrisindeki değerler sistem rijitlik matrisine ( $K_{sys}$ ) aktarılır. Daha önce oluşturulan  $DOF$  matrisi sistem rijitlik matrisinde hangi adrese gideceğini belirler.

$$K_{sys} = \begin{bmatrix} 1.3578 & 0 & -0.1789 & 0.3578 & -1 & 0 \\ 0 & 1.4311 & 0.3578 & -0.7155 & 0 & 0 \\ -0.1789 & 0.3578 & 2.3578 & 0 & -0.1789 & -0.3578 \\ 0.3578 & -0.7155 & 0 & 1.4311 & -0.3578 & -0.7155 \\ -1 & 0 & -0.1789 & -0.3578 & 1.2683 & 0.1789 \\ 0 & 0 & -0.3578 & -0.7155 & 0.1789 & 1.0733 \end{bmatrix} \quad (3.14)$$

Denklem (3.14)'de ifade edilen  $K_{sys}$  matrisi, kafes sisteminin global rijitlik matrisidir. Bu matris, sistemin yük altındaki deformasyonunu belirler ve bölüm 3.1.5'de doğal frekans ve mod şekillerini hesaplamak için kullanılmaktadır.

### 3.1.5. Modelin Frekans ve Mod Şekillerinin Elde Edilmesi

Bu son aşamada, önceden oluşturulmuş global rijitlik ve kütle matrislerini kullanarak, kafes sisteminin doğal frekanslarını ve mod şekillerini belirleme yöntemini açıklamaktadır. Bu frekanslar ve mod şekilleri, yapısal hasar tespit işleminde temel girdiler olarak kullanılır. Bu bölümde, hesaplama sonuçlarının nasıl yorumlanacağı ve daha sonraki yapısal hasar tespit sürecinde nasıl kullanılacağı detaylı olarak açıklanmıştır.

$$A = \begin{bmatrix} 0.1318 & 0 & -0.0179 & 0.0358 & -0.1 & 0 \\ 0 & 0.1431 & 0.0358 & -0.0716 & 0 & 0 \\ -0.0179 & 0.0358 & 0.2358 & 0 & -0.0179 & -0.0358 \\ 0.0358 & -0.0716 & 0 & 0.1431 & -0.0358 & -0.0716 \\ -0.1 & 0 & -0.0179 & -0.0358 & 0.1268 & 0.0179 \\ 0 & 0 & -0.0358 & -0.0716 & 0.0719 & 0.1073 \end{bmatrix} \quad (3.15)$$

Bölüm 3.1.1. denklem (3.8)'de yer alan  $A$  matrisi bu bölümde denklem (3.15)'de gösterilmektedir.  $A$  matrisini MATLAB'taki 'eig' komutuyla kullanarak,  $x_m$  (modelin gerçek mod şekilleri) ve lamda değerleri bulunmuştur. ' $\omega^2 = \lambda$ ' bu eşitliğe göre lamda değerinin karekökü alındığında,  $w_m$  (modelin gerçek frekansı) bulunur. Hasar verilmemiş modelin ölçülmüş frekans değeri denklem (3.16)'da, ölçülmüş mod şekilleri denklem (3.17)'de ifade edilmektedir.

$$w_m = \begin{bmatrix} 0.028577 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.1830 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.3231 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.4419 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.5089 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.5277 \end{bmatrix} rad/sec \quad (3.16)$$

$$x_m = \begin{bmatrix} 0.5345 & 0.4254 & 0.0930 & 0.5298 & -0.0340 & 0.4928 \\ -0.2937 & 0.2822 & -0.6437 & 0.4893 & 0.2907 & -0.3097 \\ 0.0526 & 0.1070 & 0.3505 & -0.1262 & 0.9202 & -0.0165 \\ -0.4736 & 0.4857 & -0.1728 & -0.4197 & -0.0108 & 0.5776 \\ 0.4330 & 0.5897 & -0.0954 & -0.4427 & -0.1266 & -0.4935 \\ -0.4620 & 0.3797 & 0.6444 & 0.3028 & -0.2269 & -0.2896 \end{bmatrix} \quad (3.17)$$

Bu bölümde, 3.1.3 ve 3.1.4 bölümlerinde oluşturulan global rijitlik ve kütle matrisleri kullanarak, MATLAB'ın 'eig' fonksiyonu vasıtasıyla özdeğer ve özvektör analizi gerçekleştirilmiştir. Elde edilen özdeğerler, sistemin doğal frekanslarının karesi ( $\omega^2 = \lambda$ ) olduğundan, karekök işlemi uygulanarak doğal frekanslar belirlenmiştir. Özvektör matrisi ise, her bir doğal frekans için karşılık gelen mod şekillerini temsil etmektedir. Bu parametreler, sonraki bölümlerde uygulanacak olan hasar tespit algoritmalarının girdi verilerini oluşturmakta ve bu algoritmaların performansını doğrudan etkilemektedir.

### 3.2. Yapısal Hasar Tespiti İçin Oluşturulan Optimizasyon Probleminin Matematiksel Modeli

Yapısal hasar tespiti optimizasyon problemi, oluşturulan sayısal modelin frekans ve mod şekilleri ile gerçek modelin frekans ve mod şekilleri arasındaki farkı minimize etmek amacıyla tasarlanmıştır. Bu amaçla kullanılan matematiksel model, denklem (3.1)'deki gibi ifade edilir.

$$\min f(\alpha) = \sum_{i=1}^s ((1 - MAC(\varphi_i^m, \varphi_i^c)) + ER(\omega_i^m, \omega_i^c)) \quad (3.18)$$

Denklem (3.18), yapısal hasar tespiti optimizasyon problemi için kullanılan amaç fonksiyonunu tanımlar (Zhen Chen, 2022).  $\varphi_i^m$  ve  $\omega_i^m$  sırasıyla  $i$  numaralı ölçülmüş mod şeklini ve  $i$  numaralı ölçülmüş frekansı temsil ederken,  $\varphi_i^c$  ve  $\omega_i^c$  sırasıyla  $i$  numaralı hesaplanmış mod şeklini ve  $i$  numaralı hesaplanmış frekansı ifade eder.  $s$ , mod şekli sayısını belirtmektedir. Denklem (3.18)'de ifade edildiği gibi  $MAC$  ve  $ER$  olmak üzere iki ayrı fonksiyon yer almaktadır. Bu fonksiyonlar, simülasyonda oluşturulan modelin gerçek modele ne kadar yakın olduğunu belirtmek için kullanılır.

$$MAC(\varphi_i^m, \varphi_i^c) = \frac{|(\varphi_i^m)^T \varphi_i^c|^2}{|(\varphi_i^m)^T \varphi_i^m| |(\varphi_i^c)^T \varphi_i^c|} \quad (3.19)$$

Denklem (3.19)'de ifade edilen Modal Güvence Kriteri  $MAC(\varphi_i^m, \varphi_i^c)$ , iki farklı mod şeklinin benzerliğini karşılaştırır. Karşılaştırılan mod şekilleri (vektörlerin doğrultuları) birbirine benziyorsa sonuç 1'e yakın olur. Mod şekilleri arasındaki benzerlik azaldıkça bu değer 0'a yaklaşır.

$$ER(\omega_i^m, \omega_i^c) = \left| \frac{\omega_i^m - \omega_i^c}{\omega_i^m} \right| \times 100\% \quad (3.20)$$

Denklem (3.20)'deki  $ER(\omega_i^m, \omega_i^c)$ ,  $i$  numaralı ölçülen frekans ile  $i$  numaralı hesaplanan frekansın ne kadar benzer olduğunu yüzdesel olarak ifade eder. Frekans değerleri ne kadar yakınsa,  $ER$  değeri o kadar küçük olur.

Yapısal hasar tespiti optimizasyon problemi, denklem (3.18)'de açıklanan amaç fonksiyonu  $f(\alpha)$ 'nın minimize edilmesi ile gerçekleştirilir. Bu minimize etme işlemi, yapı elemanlarının hasar oranlarını temsil eden denklem (3.21)'deki  $\alpha$  vektörü üzerinden yapılır. Bu vektör, her bir yapı elemanının hasar durumunu 0 ile 1 arasında bir değerle ifade eder.

$$\alpha = [\alpha_1, \alpha_2, \alpha_3, \dots, \alpha_{NM}], \quad 0 < \alpha < 1 \quad (3.21)$$

Denklem (3.21)'de yer alan  $\alpha$ , yapıdaki elemanların hasar oranlarını temsil eder ve  $NM$ , eleman sayısını belirtir. Bu  $\alpha$  vektörü, denklem (3.18)'deki  $f(\alpha)$ 'ı minimize edecek en uygun  $\alpha$  değerlerini belirlemek amacıyla kullanılır.

Yapısal hasar tespitinin optimizasyonu için geliştirilen matematiksel model, simülasyonla oluşturulan modelin gerçek modele yakınlığını değerlendiren ve hasar oranlarını belirlemek amacıyla optimize edilen bir yöntem sunmaktadır. *MAC* ve *ER* fonksiyonları, mod şekilleri ve frekanslar arasındaki benzerliği ölçerek modelin doğruluğunu belirler ve her iki parametrenin minimize edilmesiyle en uygun hasar oranları elde edilir.

### 3.3. Elektrikli Yılan Balığı Avlanma Optimizasyon Algoritması

#### 3.3.1. Teori

Elektrikli yılan balığı (*Electrophorus electricus*), Güney Amerika'nın tatlı sularında yaşayan, güçlü elektrik üretebilen bir balıktır. Uzun gövdeleri ve yassı kafaları vardır ve genellikle üstleri koyu yeşil veya gri, altları ise sarımsı renktedir. Görme yetenekleri zayıf olan bu balıklar, avlarını bulmak ve gezinmek için elektriksel sinyaller kullanır. Vücutlarının büyük bir kısmını oluşturan özel hücreler olan elektrositler, sinirsel uyarılarla senkronize bir şekilde çalışarak elektrik potansiyeli üretirler. Binlerce elektrosit, seri olarak bağlanmıştır ve bu seri bağlantı, avını sersemletmek veya savunma amaçlı yüksek voltajlı şoklar üretmeyi mümkün kılar. Elektrikli yılan balığı, 600 volta kadar yüksek elektrik şokları üretebilir ve bu yüksek voltaj, avını felç etmek için kullanılır. Ayrıca, elektrikli yılan balığı 10 volttan daha düşük seviyede bir yük yayarak radar gibi kullanarak çevresini algılar ve avlarını tespit eder. Bu elektrik üretme yeteneği, avlanma, iletişim ve çevreyle etkileşim için kullanılır. Elektrikli yılan balığı avlanma optimizasyon (EEFO) algoritması, bu canlıların avlanma stratejilerinden esinlenerek geliştirilmiştir. EEFO, yılan balıklarının avlarını bulmak ve yakalamak için kullandığı karmaşık iletişim, dinlenme, göç ve avlanma davranışlarını taklit eden bir meta-sezgisel optimizasyon algoritmasıdır. Algoritma, topluluk tabanlı bir yaklaşımdır ve çözüm uzayını etkili bir şekilde aramak için hem keşif hem de sömürme stratejilerini birleştirir.

#### 3.3.2. Etkileşim Aşaması

Yılan balıkları bir balık sürüsüyle karşılaştıklarında yüzerek birbirleriyle etkileşime girer ve organize bir şekilde hareket ederler. Daha sonra birlikte büyük bir elektriksel halka oluşturarak, sürünün ortasındaki küçük balıkları tuzağa düşürmek için merkezde toplanırlar. EEFO algoritmasında, her elektrikli yılan balığı bir aday çözüm olarak değerlendirilir ve mevcut en iyi çözüm bir av olarak kabul edilir. Bu yapı, yılan balıklarının birbirlerinin konum bilgilerini kullanarak işbirliği içinde çalıştığı ve grup dinamikleriyle etkileşimde bulunduğu küresel keşif sürecini simgeler. Özellikle bir yılan balığı, diğer bireylerin pozisyonlarına göre, rastgele seçilen başka bir bireyle etkileşim kurabilir. Bu etkileşimde, yılan balığının pozisyonu, seçilen birey ile grup merkezi

arasındaki konumsal fark dikkate alınarak güncellenir. Ayrıca, yerel arama aşamasında, yılan balığı, rastgele seçilen başka bir birey ve arama alanında rastgele belirlenmiş bir nokta arasında konumunu güncelleyebilir. Yılan balıkları arasındaki bu etkileşimler, farklı yönlerde doğru rastgele hareketleri temsil eden bir yayılım mekanizmasıyla açıklanır.

$$C = n_1 \times B \quad (3.22)$$

$$n_1 \sim N(0,1) \quad (3.23)$$

$$B = [b_1, b_1, \dots b_k, \dots b_d] \quad (3.24)$$

Denklem (3.22)'deki  $C$  vektörü, hareketin büyüklüğü ve yönünü ifade eder. Denklem (3.23)'deki  $n_1$   $[0,1]$  arasında üretilen rasgele bir değerdir. Denklem (3.24)'deki  $B$  ise bu büyüklüğün hangi boyutlarda uygulanacağını belirleyen bir seçim görevi görür.

$$b(k) = \begin{cases} 1 & \text{if } k == g\{l \\ 0 & \text{else} \end{cases} \quad (3.25)$$

$$g = \text{randperm}(d) \quad (3.26)$$

Denklem (3.25),  $B$  seçim vektörünün elemanlarının nasıl doldurulacağını belirleyen temel mantıksal kuralı ifade eder. Vektörün  $k$ -inci elemanı olan  $b(k)$ 'nin değeri, rastgele elde edilen  $g$  dizisinin ilgili elemanı ile karşılaştırılmasına dayanır. Eğer  $k$  indeksi,  $g$  değerine eşitse,  $b(k)$ 'ya '1' değeri atanır; bu da ilgili boyutun değişim hareketine katılmak üzere seçildiğini gösterir. Diğer tüm durumlarda ise  $b(k)$ 'ya '0' değeri atanır.

$$l = 1, \dots, \left\lceil \left( \frac{T-t}{T} \right) \times r_1 \times (d-2) + 2 \right\rceil \quad (3.27)$$

Denklem (3.27)'deki  $l$  değişecek boyut sayısını ifade eder. Her iterasyonda rastgele seçilen belirli sayıda boyut değişime uğratılır. Bu mekanizma, keşif hareketinin arama uzayının farklı alt bölgelerinde yoğunlaşmasını sağlayarak algoritmanın arama çeşitliliğini ve etkinliğini artırır. Burada  $T$ , maksimum iterasyon sayısı,  $t$  ise mevcut iterasyon sayıdır.

$$\left\{ \begin{array}{ll} v_i(t+1) = x_j(t) + C \times (\bar{x}(t) - x_i(t)) & p_1 > 0.5 \quad \text{fit}(x_j(t)) < \text{fit}(x_i(t)) \\ v_i(t+1) = x_j(t) + C \times (x_r(t) - x_i(t)) & p_1 \leq 0.5 \quad \text{fit}(x_j(t)) < \text{fit}(x_i(t)) \\ v_i(t+1) = x_i(t) + C \times (\bar{x}(t) - x_j(t)) & p_2 > 0.5 \quad \text{fit}(x_j(t)) \geq \text{fit}(x_i(t)) \\ v_i(t+1) = x_i(t) + C \times (x_r(t) - x_i(t)) & p_2 \leq 0.5 \quad \text{fit}(x_j(t)) \geq \text{fit}(x_i(t)) \end{array} \right. \quad (3.28)$$

Denklem (3.28), EEFO algoritmasının temel keşif mekanizması olan etkileşim davranışını matematiksel olarak formüle eder. Bu denklem, her bir çözüm bireyi için potansiyel bir sonraki pozisyon  $v_i(t+1)$  üretirken, bireyin mevcut uygunluk değerine

$fit(x_j(t))$  göre adaptif bir strateji uygular. Eğer bireyin uygunluğu o anki en iyi çözümden daha kötüyse, denklem bireyi en iyi çözümün bulunduğu umut vadeden bölgeye doğru yönlendirerek yakınsamayı teşvik eder. Eğer birey zaten en iyi çözüm konumundaysa veya ona eşitse, denklem bu başarılı konumu koruyarak ve etrafında küçük değişiklikler yaparak hassas bir yerel arama gerçekleştirir. Bu yapı, hem sömürü hem de keşif arasında dinamik bir denge kurarak, algoritmanın hem global arama yeteneğini korumasını hem de yerel optimumlara takılıp kalma riskini azaltmasını sağlar.

$$\bar{x}(t) = \frac{1}{n} \sum_{i=1}^n x_j(t) \quad (3.29)$$

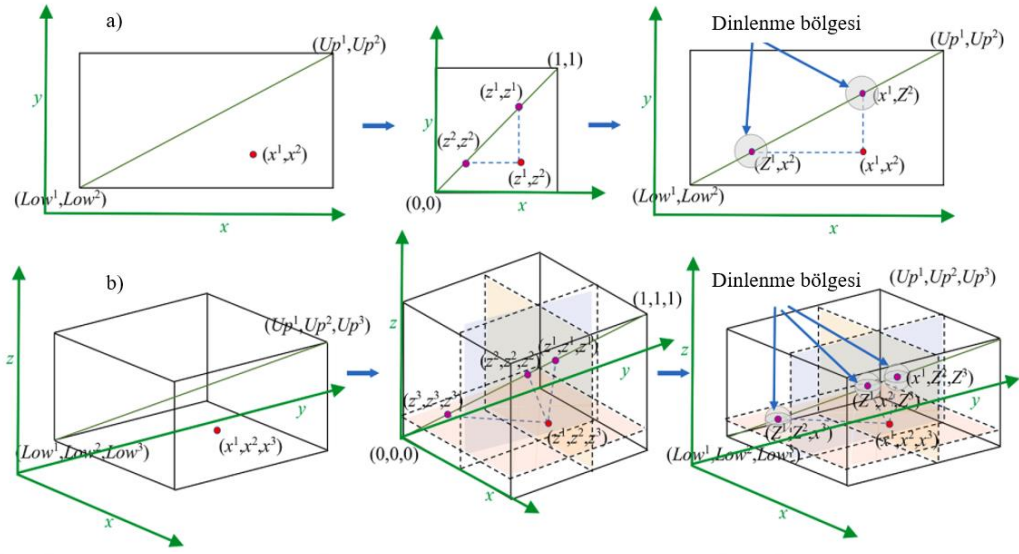
Denklem (3.29), popülasyondaki tüm bireylerin mevcut pozisyonlarının aritmetik ortalamasını alarak, sürünün o anki ağırlık merkezini veya ortalama konumunu temsil eden  $\bar{x}(t)$ 'yi hesaplar. Bu ortalama konum, denklem (3.28)'de, özellikle en iyi konumdaki bireyin popülasyonun genelinden çok fazla uzaklaşmasını engellemek ve sürü bütünlüğünü korumak amacıyla bir referans noktası olarak kullanılır.

$$x_r = Low + r \times (Up - Low) \quad (3.30)$$

Denklem (3.30)'de yer alan  $x_r$ , algoritmanın arama uzayının sınırları dışına çıkan bir bireyin pozisyonunu  $[0,1]$  aralığına geri çekerek oluşturulan yeni bireyi temsil eder. Eğer bir bireyin pozisyonu Low (alt sınır) ve Up (üst sınır) ile tanımlanan geçerli aralığın dışına taşarsa, bu denklem o birey için  $r$  rastgele vektörü kullanılarak sınırlar içinde yeni bir rastgele pozisyon üretir.

### 3.3.3. Dinlenme Aşaması

EEFO algoritmasında, elektrikli yılan balıklarının dinlenme davranışının gerçekleştirilmesinden önce, bir dinlenme alanının tanımlanması gerekmektedir. Arama etkinliğini artırmak amacıyla, yılan balığının konum vektörünün herhangi bir boyutunun, arama uzayının ana köşegeni üzerine yansıtılmasıyla bir dinlenme bölgesi oluşturulur (Şekil 3.2). Bu işlemin gerçekleştirilebilmesi için, hem arama alanı hem de yılan balığının mevcut konumu 0 ile 1 aralığına normalize edilir. Daha sonra, yılan balığının konum vektöründen rastgele seçilen bir boyut, normalize edilmiş arama uzayının ana köşegeni üzerine yansıtılarak öngörülen bir konum elde edilir. Bu konum, dinlenme bölgesinin merkezi olarak kabul edilir.



**Şekil 3.2.** Dinlenme alanının belirlenmesi; **a)** 2D gösterimi; **b)** 3D gösterimi (Zhao, ve diğerleri, 2024)

$$\{X | X - Z(t) \leq \alpha_0 \times |Z(t) - x_{prey}(t)|\} \quad (3.31)$$

Denklem (3.31), bir elektrikli yılan balığının dinlenme davranışını sergileyeceği arama uzayındaki hedef bölgeyi matematiksel olarak tanımlar. Bu bölge, Denklem (3.33)'de hesaplanan merkez noktası  $Z(t)$ 'ye sahip ve Denklem (3.32) ile belirlenen  $\alpha_0$  katsayısıyla ölçeklenen yarıçapı, mevcut en iyi çözüm  $x_{prey}(t)$  ile merkez noktası arasındaki mesafeye bağlı olan bir alandır. Algoritmanın bu bölgeyi tanımlaması, arama odağını mevcut en iyi çözümün yakınılarında umut vadeden bir alt bölgeye daraltarak yerel arama yeteneğini artırma stratejisinin bir parçasıdır.

$$\alpha_0 = 2 \cdot \left( e - e^{\frac{t}{T}} \right) \quad (3.32)$$

Denklem (3.32) dinlenme alanının başlangıç yarıçapını belirleyen  $\alpha_0$  ölçek faktörünü hesaplar. Euler sayısına ( $e$ ) dayanan bu formülasyon,  $\alpha_0$ 'a başlangıçta büyük bir değer atanır ve iterasyon ilerledikçe bu değeri kademeli olarak küçültür. Bu dinamik ayarlama, arama sürecinin başlarında daha geniş bir dinlenme alanı tanımlanmasına, süreç ilerledikçe ise bu alanın daraltılarak daha hassas ve odaklanmış bir yerel arama yapılmasına olanak tanır.

$$Z(t) = Low + z(t) \times (Up - Low) \quad (3.33)$$

Denklem (3.33), dinlenme alanının merkez noktasını belirlemek için kullanılan normalleştirilmiş pozisyon vektörünü hesaplar. Arama uzayının herhangi bir boyutunda rastgele seçilen bir bireyin o boyuttaki konumunu,  $[0, 1]$  aralığına çeker. Bu normalizasyon işlemi, problemin farklı boyutlardaki ölçek farklılıklarından

kaynaklanabilecek yanlışlıkları ortadan kaldırarak, dinlenme alanının merkezinin arama uzayı içinde tutarlı bir şekilde belirlenmesini sağlar.

$$z\{t = \frac{x_{rand\{n\}}^{rand\{d\}}\{t - Low^{rand\{d\}}}{Up^{rand\{d\}} - Low^{rand\{d\}}} \quad (3.34)$$

Denklem (3.34), denklem (3.33)'de hesaplanan normalleştirilmiş merkez noktasını, problemin orijinal arama uzayı sınırlarına (Low, Up) geri ölçekleyerek  $Z(t)$  nihai merkez pozisyonunu elde eder. Bu işlem, normalleştirme adımının tersidir ve dinlenme alanının merkezinin, optimizasyon probleminin gerçek koordinat sisteminde anlamlı bir konuma karşılık gelmesini sağlar. Bu  $Z(t)$  noktası, dinlenme davranışının hedef merkezi olarak kullanılır.  $\alpha_0 \times |Z(t) - x_{prey}(t)|$  terimi dinlenme alanının aralığını belirtir,  $x_{rand}^{rand}$  mevcut popülasyondan rastgele seçilen bir bireyin rastgele konum boyutudur,  $z$  normalleştirilmiş sayıdır. Böylece yılan balığının dinlenme davranışını gerçekleştirmeden önce dinlenme alanı içindeki dinlenme pozisyonu elde edilir:

$$R_i(t + 1) = Z(t) + \alpha \times |Z(t) - x_{prey}(t)| \quad (3.35)$$

Denklem (3.35), bir bireyin dinlenme alanı içindeki hedef pozisyonu olan  $R_i(t + 1)$ 'i hesaplar. Bu pozisyon, dinlenme alanının merkezi olan  $Z(t)$ 'ye, Denklem (3.36) ile hesaplanan ve zamanla küçülen bir ölçek faktörü ( $\alpha$ ) ile ayarlanmış rastgele bir değişiklik eklenerek belirlenir. Bu, bireyin doğrudan merkeze gitmek yerine, merkezin etrafındaki rastgele bir noktaya yönelmesini sağlayarak arama çeşitliliğini korur ve algoritmanın tek bir noktaya sıkışmasını engeller.  $\alpha_0 \times |Z(t) - x_{prey}(t)|$  terimi dinlenme alanının aralığını belirtir,  $x_{rand}^{rand}$  mevcut popülasyondan rastgele seçilen bir bireyin rastgele konum boyutudur,  $z$  normalleştirilmiş sayıdır.

$$\alpha = \alpha_0 \times \sin(2\pi r_2) \quad (3.36)$$

Denklem (3.36), dinlenme alanının yarıçapını her iterasyonda dinamik olarak ayarlayan  $\alpha$  ölçek faktörünü hesaplar. Başlangıç ölçeği olan  $\alpha_0$ 'ı,  $\sin(2\pi r_2)$  gibi periyodik bir fonksiyonla çarparak, dinlenme alanının boyutunun iterasyonlar boyunca hem küçülme eğilimi göstermesini hem de dalgalanmasını sağlar.

$$v_i(t + 1) = R_i(t + 1) + n_2 \times (R_i(t + 1) \text{round}(\text{rand}) \times x_i(t)) \quad (3.37)$$

$$n_2 \sim N(0,1) \quad (3.38)$$

Denklem (3.37), bireyin nihai pozisyon güncellemesini modelleyen dinlenme davranışının kendisidir. Bu son adım, bireyin hedef dinlenme pozisyonunun tam üzerine yerleşmek yerine, onun çok yakınında rastgele bir noktaya konumlanmasını sağlayarak, yerel sömürü sürecine ince bir keşif unsuru daha katar.

### 3.3.4. Avlama Aşaması

Yılan balıkları bir av tespit ettiklerinde, doğrudan saldırıya geçmek yerine, kolektif bir davranış sergileyerek geniş bir daire formasyonu oluşturacak şekilde organize biçimde yüzmeye başlarlar. Bu süreçte, düşük frekanstaki elektriksel organ deşarjları aracılığıyla birbirleriyle sürekli iletişim kurar ve koordineli bir şekilde hareket ederler. Yılan balıkları arasındaki etkileşim yoğunlaştıkça, oluşturulan elektriksel çember daralır. Nihayetinde, bu daire biçimindeki formasyon yardımıyla balık sürüsü, derin kısımdan sığ bölgelere doğru yönlendirilir ve böylece avlanma daha kolay hale gelir. Bu davranıştan yola çıkarak, elektriksel çemberin avlanma alanına dönüştüğü kabul edilir. Bu esnada av, bu sınırlı alan içerisinde hareket etmeye zorlanır; dolayısıyla bulunduğu konumdan hızla ve ardışık biçimde farklı konumlara yönelir.

$$\{X|X - x_{prey}(t)| \leq \beta_0 \times |\bar{x}(t) - x_{prey}(t)|\} \quad (3.39)$$

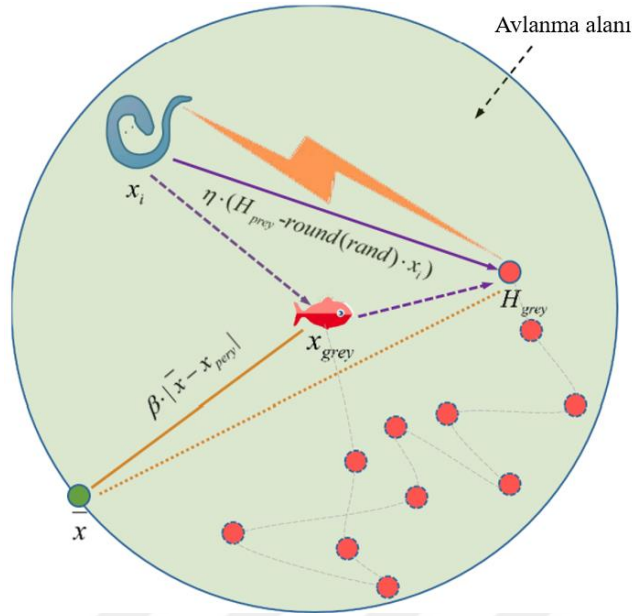
$$\beta_0 = 2 \times \left(e - e^{\frac{t}{T}}\right) \quad (3.40)$$

Denklem (3.39), elektrikli yılan balığının avlanma davranışını gerçekleştireceği hedef bölgeyi matematiksel olarak tanımlar. Bu bölgenin yarıçapı, Denklem (3.40) ile hesaplanan ve iterasyonlar ilerledikçe küçülen  $\beta_0$  başlangıç ölçek faktörüne ve popülasyonun ortalama konumu ile avın konumu arasındaki mesafeye bağlı olarak dinamik bir şekilde belirlenir. Bu alan, algoritmanın arama odağını doğrudan en iyi çözümün etrafındaki çok daha dar bir bölgeye odaklayarak, güçlü bir yerel sömürü stratejisi uygular.

$$H_{prey}(t + 1) = x_{prey}(t) + \beta \times |\bar{x}(t) - x_{prey}(t)| \quad (3.41)$$

$$\beta = \beta_0 \times \sin(2\pi r_3) \quad (3.42)$$

Denklem (3.41), avlanma alanı içinde avın (en iyi çözümün) potansiyel yeni konumunu, yani  $H_{prey}(t + 1)$ 'i hesaplar. Bu yeni konum, avın mevcut pozisyonu olan  $x_{prey}(t)$ 'ye, Denklem (3.42) ile hesaplanan ve zamanla küçülen bir ölçek ile ayarlanmış rastgele bir değişim eklenerek belirlenir. Bu, avın tamamen sabit kalmak yerine, avlanma alanı içinde küçük ve rastgele hareketler yaptığını simüle eder. Bu tahmin, yılan balığının bir sonraki adımda yöneleceği hedef noktayı oluşturur.



**Şekil 3.3.** Kıvrılma hareketi gösterimi (Zhao, ve diğerleri, 2024)

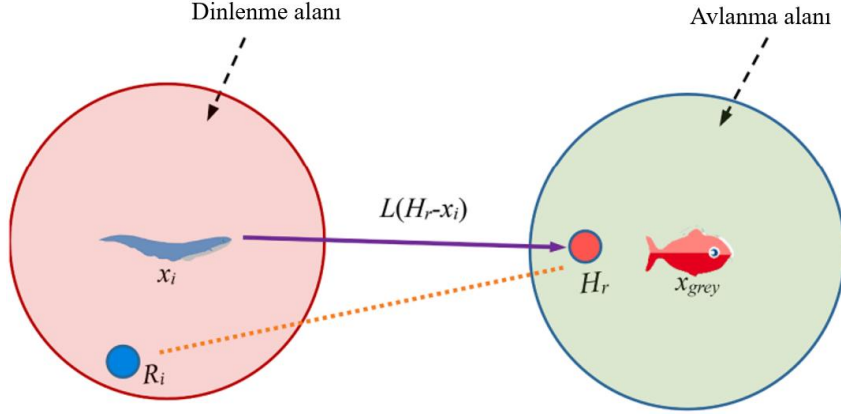
$$v_i(t+1) = H_{prey}(t+1) + \eta \times (H_{prey}(t+1) - round(rand) \times x_i(t)) \quad (3.43)$$

$$\eta = e^{\frac{r_4(1-t)}{T}} \times \cos(2\pi r_4) \quad (3.44)$$

Denklem (3.43), elektrikli yılan balığının avını yakalamak için yaptığı "kırılma (curling)" hareketini (Şekil 3.3) matematiksel olarak modeller ve bireyin nihai pozisyonunu günceller. Bireyin yeni pozisyonu, avın Denklem (3.41) ile tahmin edilen yeni konumu olan  $H_{prey}(t + 1)$ 'e doğru bir yönelimle hesaplanır. Bu yönelime ek olarak, Denklem (3.44) ile hesaplanan  $\eta$  kırılma faktörü ile ölçeklenmiş, avın tahmini pozisyonu ile bireyin mevcut konumu arasındaki farka dayalı ikinci bir terim eklenir. Bu, bireyin avın etrafında bir yay çizerek ona yaklaşmasını simüle eden ve algoritmanın yerel arama yeteneğini güçlendiren sofistike bir sömürü manevrasıdır.

### 3.3.5. Göç Aşaması

Algoritmanın göç aşaması, bir elektrikli yılan balığının daha az verimli olan dinlenme alanından, avın bulunduğu daha umut vadeden avlanma alanına doğru yaptığı stratejik hareketi modeller (Şekil 3.4). Bu davranış, bireyin pozisyonunu hem dinlenme hem de avlanma alanlarından gelen bilgilere göre günceller. Bu sürece ek olarak, yerel optimum noktalarına takılma riskini en aza indirmek ve arama kabiliyetini artırmak amacıyla, Levy Uçuşu (Levy Flight) gibi güçlü bir stokastik mekanizma da denkleme dahil edilmiştir. Bu hibrit yaklaşım, algoritmanın hem hedefe yönelik kararlı bir ilerleme kaydetmesini hem de keşif potansiyelini kaybetmeden arama uzayında beklenmedik sıçramalar yapabilmesini sağlar.



Şekil 3.4. Göç davranışının gösterimi (Zhao, ve diğerleri, 2024)

$$v_i(t+1) = r_5 \times R_i(t+1) + r_6 \times H_r(t+1) - L \times (H_r(t+1) - x_i(t)) \quad (3.45)$$

Denklem (3.45), bireyin göç davranışını matematiksel olarak formüle eder ve yeni aday pozisyonunu hesaplar. Bu formülasyon, bireyin hem dinlenme alanı hedef pozisyonuna hem de avlanma alanı hedef pozisyonuna doğru rastgele ağırlıklarla yönlendirildiği bir hibrit hareketi modeller. Bu iki hedefe ek olarak, denklemden Levy Uçuşu mekanizmasından türetilen  $L$  terimi çıkarılır.  $H_r(t+1) - x_i(t)$  terimi yılan balıklarının avlanma alanına doğru hareket ettiğini belirtir.

$$H_r(t+1) = x_{prey}(t) + \beta \times |\bar{x}(t) - x_{prey}(t)| \quad (3.46)$$

Denklem (3.46), göç davranışı sırasında kullanılan avlanma alanı hedef pozisyonunu tanımlar. Denklem (3.41) ile yapısal olarak aynı olan bu formül, bireyin sadece dinlenme alanından uzaklaşmakla kalmayıp, aynı zamanda mevcut en iyi çözümün bulunduğu çekim merkezine doğru da bir yönelim göstermesini sağlar.

$$L = 0.01 \times \left| \frac{u \cdot \sigma}{|v|^{\frac{1}{b}}} \right| \quad (3.47)$$

$$u, v \sim N(0,1) \quad (3.48)$$

Denklem (3.47), Levy Uçuşu (Viswanathan, ve diğerleri, 1996) adımının büyüklüğünü belirleyen  $L$  vektörünü hesaplar. Formül, standart normal dağılımdan ( $u$  ve  $v$ ) gelen rastgele adımların oranını, standart Gamma fonksiyonu ( $\Gamma$ ) ve  $b$  parametresi ile ölçeklenmiş bir olasılık dağılımı ile çarpılarak, hem kısa adımlardan hem de nadir fakat çok uzun sıçramalardan oluşan bir hareket deseni üretir. Bu  $L$  terimi, Denklem (3.45)'te kullanılarak, algoritmanın arama sürecine güçlü bir stokastik keşif unsuru ekler, yerel optimumlardan kaçmak için uzun ve öngörülemez sıçramalar yapma kabiliyeti kazanır, bu da algoritmanın global arama yeteneğini güçlendirir.

$$\sigma = \left( \frac{\Gamma(1+b) \times \sin\left(\frac{\pi b}{2}\right)}{\Gamma\left(\frac{1+b}{2}\right) \times b \times 2^{\frac{b-1}{2}}}\right)^{\frac{1}{b}} \quad (3.49)$$

Denklem (3.49), kullanılan Levy uçuşu fonksiyonunun adım büyüklüğünü hesaplayan standart matematiksel formülüdür. Burada yer alan  $\Gamma$  standart Gama fonksiyonudur ve  $b=1,5$ 'tir. Yılan balığı, düşük elektrik deşarjı sayesinde avının konumunu algılayabilir ve bu sayede her an kendi konumunu ayarlayabilir. Yılan balığı yiyecek arama sürecinde avın yaklaştığını hissederse aday pozisyonuna geçer; aksi takdirde yılan balıkları mevcut pozisyonunda kalır.

$$x_i(t+1) = \begin{cases} x_i(t) & \text{fit}(x_i(t)) \leq \text{fit}(v_i(t+1)) \\ v_i(t+1) & \text{fit}(x_i(t)) > \text{fit}(v_i(t+1)) \end{cases} \quad (3.50)$$

Denklem (3.50), tüm davranış modellerinden sonra uygulanan açgözlü seçim (greedy selection) mekanizmasını ifade eder. Herhangi bir davranışla üretilen yeni aday çözümün uygunluk değeri, bireyin mevcut çözümünün uygunluk değerinden daha iyiye, bireyin pozisyonu bu yeni ve daha üstün pozisyonla güncellenir. Bu kural, popülasyondaki çözümlerin kalitesinin asla kötüleşmemesini sağlayarak algoritmanın kararlı bir şekilde daha iyi sonuçlara yakınsamasını temin eder.

### 3.3.6. Keşiften Sömürüye Geçiş

EEFO'da arama davranışları, algoritmanın optimizasyon performansını iyileştirmek için keşif ve kullanım arasındaki geçişi etkili bir şekilde yönetebilen bir enerji faktörü tarafından belirlenir (Wang, 2019; Heidari, 2019). Yılan balığının enerji faktörünün değeri, keşif ve kullanım arasında seçim yapmak için kullanılır.

$$E(t) = 4 \times \sin\left(1 - \frac{t}{T}\right) \times \ln \frac{1}{r_7} \quad (3.51)$$

Denklem (3.51), algoritmanın davranış modunu belirleyen Enerji Faktörü'nü formüle eder. Bu faktör, üç temel bileşenin birleşiminden oluşur:  $\left(1 - \frac{t}{T}\right)$  terimi,  $t$  arttıkça maksimum iterasyon sayısına oranla azalan bir eğilim yaratarak algoritmayı genel olarak keşif fazından sömürü fazına yönlendirir.  $\sin(\dots)$  fonksiyonu, bu azalışa salınımlı bir karakter kazandırarak, geçişin doğrusal olmasını engeller ve algoritmanın ileri iterasyonlarda dahi yerel optimumdan kaçma şansını sürdürmesine olanak tanır. Son olarak,  $\ln \frac{1}{r_7}$  terimi,  $r_7$  rastgele sayısı aracılığıyla enerji değerine rastgele bir unsur ekleyerek arama yollarının öngörülebilirliğini azaltır ve arama kabiliyetini güçlendirir.

Enerji faktörü 1'den büyük olduğunda, yılan balıkları etkileşimli davranış yoluyla tüm değişken uzayda küresel arama gerçekleştirir ve keşifle sonuçlanır. Enerji faktörü 1'den küçük veya 1'e eşit olduğunda, yılan balıkları göç etme, dinlenme veya avlanma

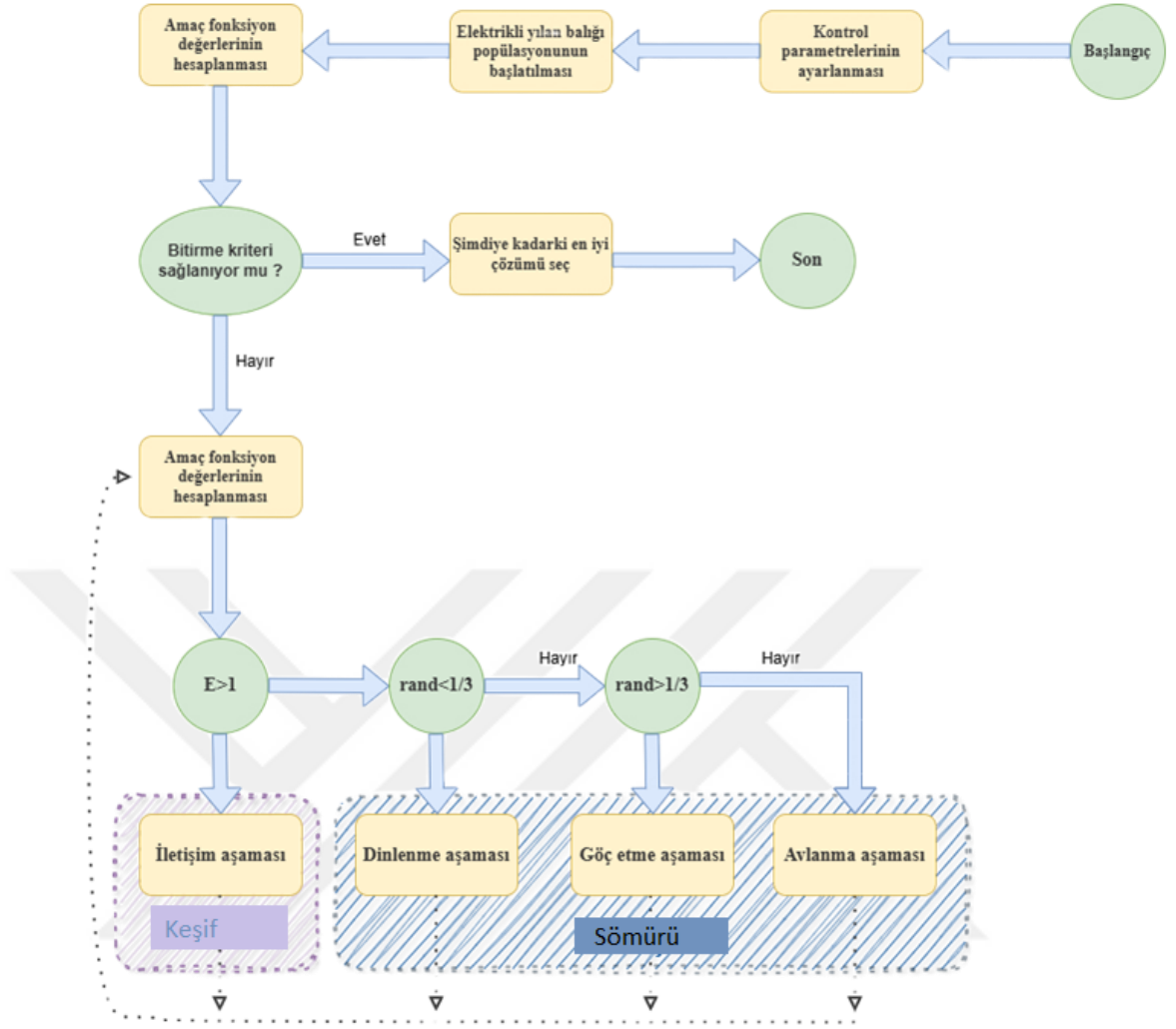
davranışları yoluyla gelecek vaat eden bir alt bölgede yerel arama yapma eğilimindedir ve bu da sömürüyle sonuçlanır. Yinelemelerin ilk yarısında keşif daha yüksek olasılıkla gerçekleşirken, yinelemelerin ikinci yarısında yararlanma daha yüksek olasılıkla gerçekleşir. EEFO'nun arama davranışını incelemek için, optimizasyon süreci boyunca  $E > 1$  olasılığı değerlendirilir.

$$P\{E(t) > 1\} = \frac{\int_0^1 \int_0^e \frac{1}{4 \sin(\theta)} dr d\theta}{1} = - \int_{-\infty}^{\frac{1}{1-4 \sin(1)}} \frac{e^x dx}{x \sqrt{16x^2 - 1}} \approx 0.5035 \quad (3.52)$$

Denklem (3.52)'deki sonuca göre optimizasyon sürecinde keşif veya kullanım arasında gerçekleşme olasılığı yaklaşık %50'dir. Keşif ve sömürünün dengelenmesine büyük katkı sağlar. Bu matematiksel bulgu, EEFO algoritmasının tasarım gereği, çalışma süresinin yaklaşık olarak yarısını keşif ( $E > 1$ ) modunda, diğer yarısını ise sömürü ( $E \leq 1$ ) modunda geçirdiğini kanıtlamaktadır. Bu durum, algoritmanın küresel arama ile yerel arama arasında dengeli bir strateji izlediğini ve bu sayede hem yeni ve umut vadeden çözüm bölgelerini etkin bir şekilde arayabildiğini hem de bulunan iyi çözümleri hassas bir şekilde iyileştirebildiğini göstermesi açısından büyük önem taşımaktadır.

### 3.3.7. Yöntemin Algoritması

EEFO algoritması, popülasyon büyüklüğü ve maksimum iterasyon sayısı gibi kontrol parametrelerinin belirlenmesiyle başlar. Bu aşamada, rastgele bir yılan balığı popülasyonu oluşturulur. Her yinelemede, enerji faktörü  $E > 1$  olduğunda, yılan balıkları etkileşim davranışıyla keşif yapar. Enerji faktörü  $E \leq 1$  olduğunda ise, yılan balıkları dinlenme, göç veya avlanma davranışlarından birini eşit olasılıkla seçerek sömürü sürecini gerçekleştirir. Her durumda, tüm yılan balıkları yeni aday çözümler üretir ve bu çözümler mevcut olanlarla karşılaştırılır. Bu süreçte, bulunan en iyi çözüm güncellenir. Yinelemeler ilerledikçe  $E$  değeri azalır ve bu, yılan balıklarının keşiften sömürüye geçişini sağlar. İşlem, belirlenen durma kriteri karşılanana kadar devam eder ve sonunda en iyi çözüm kaydedilir. Algoritmanın akış şeması Şekil (3.5)'de sunulmaktadır.



Şekil 3.5. EEFO algoritma şeması

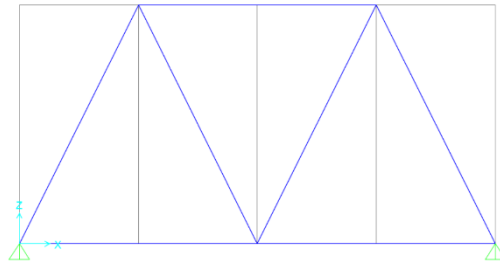
#### 4. BULGULAR VE TARTIŞMA

Bu bölümde, önceki bölümlerde teorik altyapısı ve metodolojisi açıklanan sayısal analizlerin sonuçları detaylı bir şekilde sunulmaktadır. İlk olarak, bu çalışmada geliştirilen MATLAB tabanlı sonlu eleman modelinin doğruluğu ve güvenilirliği, ticari bir yazılım olan SAP2000 ve literatürdeki referans çalışmalar ile karşılaştırılarak teyit edilmiştir. Modelin doğrulanmasının ardından, çalışmanın ana odağı olan Elektrikli Yılan Balığı Avlanma Optimizasyonu (EEFO) algoritmasının yapısal hasar tespitindeki performansı, farklı karmaşıklık seviyelerini temsil eden iki referans problem üzerinde incelenmiştir: 10 elemanlı düzlemsel kafes ve 72 elemanlı uzaysal kafes. Her bir problem için hem tekli hem de çoklu hasar senaryoları ele alınmış; algoritmaların etkinliği, literatürde yaygın olarak kullanılan HS (Harmoni Arama) ve ABC (Yapay Arı Kolonisi) algoritmaları ile kıyaslanmıştır. Sonuçlar, algoritmaların yakınsama davranışları, nihai hasar oranı tahminleri ve farklı popülasyon büyüklükleri altındaki istatistiksel performansları üzerinden kapsamlı bir şekilde değerlendirilmiştir.

##### 4.1. Sap2000 ve MATLAB Doğrulaması

###### 4.1.1. 7 Elemanlı Düzlemsel Kafes Doğrulaması

Doğrulama amacıyla kullanılan Şekil 4.1'deki 7 elemanlı modelin tüm girdilerinin detaylı bir açıklaması, Bölüm 3.1.3'de sunulmaktadır. Örnek model, hem MATLAB ortamında hem de SAP2000 yazılımında oluşturulmuştur. İki platformda elde edilen frekans değerlerinin uyumlu olması hedeflenmiş ve bu doğrultuda gerçekleştirilen analizler, kodun doğruluğunu ve güvenilirliğini ortaya koymaktadır. 7 elemanlı örnek kafesin SAP2000'de modellenirken kullanılan eleman özellikleri Bölüm 3.1.3'de açıklanmaktadır. Karşılaştırma sonuçları ilgili çizelgelerde sunulmuş olup, sayısal verilerin sunumunda hassasiyeti ve okunabilirliği sağlamak amacıyla, değerler virgülden sonra üç basamağa yuvarlanarak verilmiştir.



Şekil 4.1. 7 elemanlı düzlemsel kafesin SAP2000 modeli

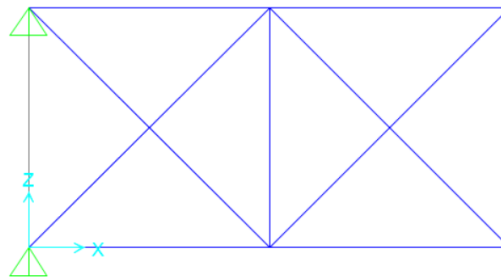
**Çizelge 4.1.** 7 elemanlı örneğin Matlab ve SAP2000 ile hesaplanan doğal frekansların karşılaştırılması

| Mod | Matlab Frekansı (Hz) | SAP2000 Frekansı (Hz) | Fark (Hz) | Yüzde Hata (%) |
|-----|----------------------|-----------------------|-----------|----------------|
| 1   | 0,028                | 0,028                 | 0,000     | 0%             |
| 2   | 0,029                | 0,029                 | 0,000     | 0%             |
| 3   | 0,055                | 0,055                 | 0,000     | 0%             |
| 4   | 0,072                | 0,072                 | 0,000     | 0%             |
| 5   | 0,081                | 0,081                 | 0,000     | 0%             |
| 6   | 0,084                | 0,084                 | 0,000     | 0%             |

Çizelge 4.1'de, incelenen yapının ilk altı moduna ait doğal frekans değerleri MATLAB ve SAP2000 programları kullanılarak hesaplanmıştır. Elde edilen frekans değerleri, hem büyüklük hem de yüzde hata açısından karşılaştırılmıştır. Her iki yöntemle elde edilen sonuçların tamamen örtüştüğü, frekans farkının 0 Hz olduğu ve yüzde hatanın %0 olduğu görülmektedir. Bu durum, MATLAB ile geliştirilen analiz yönteminin doğruluğunu ve güvenilirliğini ortaya koymaktadır.

#### 4.1.2. 10 Elemanlı Düzlemsel Kafes Doğrulaması

Şekil 4.2'de sunulan model, yapı dinamiği alanında yaygın olarak kullanılan ve literatürde sıklıkla referans gösterilen örneklerden biridir (Minh, Khatir, Abdel Wahab, & Cuong-Le, 2021). Bu modelin tercih edilmesinin temel nedeni, genel geçer mühendislik özelliklerine sahip olması ve geniş bir uygulama alanında kendini kanıtlamış olmasıdır. Modelin geçerliliği ve güvenilirliği, çok sayıda akademik çalışmada test edilmiş ve farklı araştırmalarla desteklenmiştir. Bu tür standart ve yaygın kabul görmüş modellerin kullanımı, yapılan analizlerin literatürdeki diğer çalışmalarla karşılaştırılabilirliğini artırmakta; dolayısıyla elde edilen sonuçların güvenilirliğini ve geçerliliğini pekiştirmektedir. Bu bağlamda, mevcut çalışmada söz konusu modelin kullanılması, doğrulama süreçlerinin etkin ve tutarlı bir şekilde yürütülmesine olanak sağlamıştır.



**Şekil 4.2.** 10 elemanlı düzlemsel kafesin SAP2000 modeli

Kullanılan yapı modelinde (Şekil 4.2), elemanların elastisite modülü (E)  $2 \times 10^8 \text{ kN/m}^2$ , malzeme yoğunluğu ( $\rho$ )  $76,98 \text{ kN/m}^3$ , olarak belirlenmiştir. Yapı elemanlarının kesit özellikleri: yükseklik 300 mm, üst ve alt flanş genişliği 125 mm, üst ve alt flanş kalınlığı 8 mm ve gövde kalınlığı 8 mm olarak seçilmiştir. Eleman uzunluğu ise 9,14 m'dir. Örnek model, Bölüm 3.1.2.'de detaylı olarak açıklandığı üzere, MATLAB programı kullanılarak hazırlanmış, gerekli girdiler oluşturulmuş ve analiz gerçekleştirilmiştir.

Bu aşamada MATLAB tabanlı analiz aracının doğruluğunu test etmek amacıyla, aynı yapısal sistem SAP2000 programı kullanılarak da modellenmiş ve serbest titreşim analizleri gerçekleştirilmiştir. Her iki yazılım ile elde edilen hasarsız doğal frekans değerleri makale değerleri ile karşılaştırılmıştır. Bu karşılaştırmalar, Tablo 4.2'de sunulmuştur.

**Çizelge 4.2.** 10 elemanlı örneğin SAP2000, MATLAB ve literatürdeki frekans değerlerinin karşılaştırılması

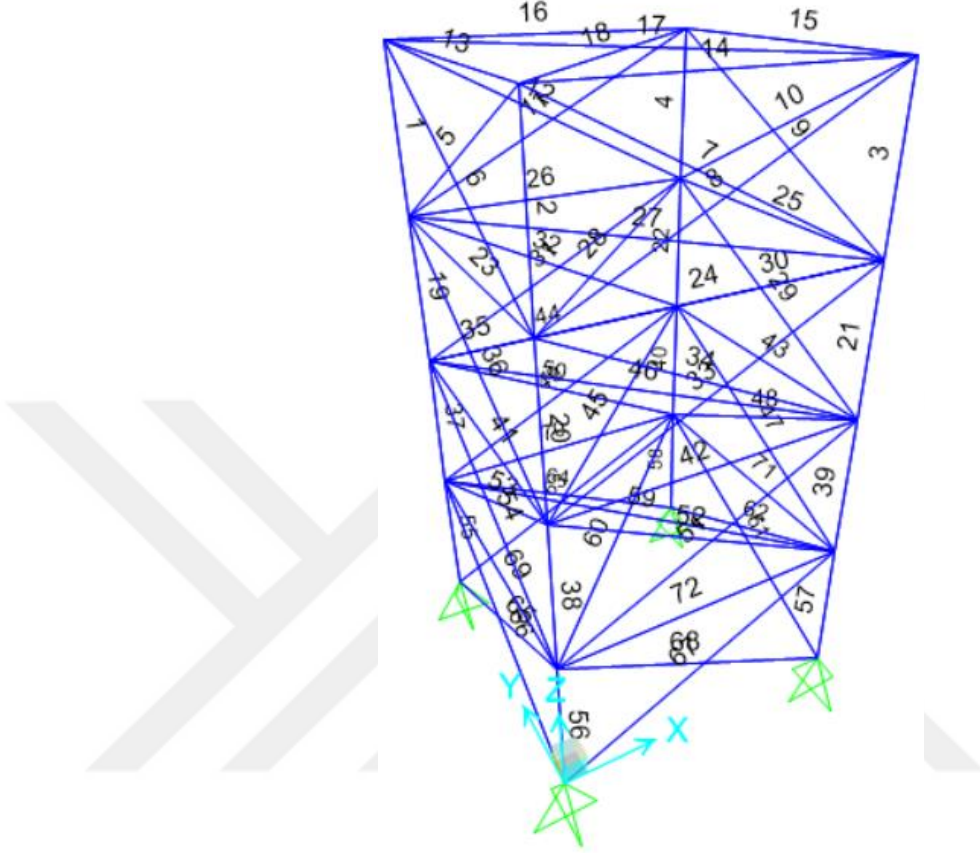
| Mod | Matlab Frekansı (Hz) | SAP2000 Frekansı (Hz) | (Minh, Khatir, Abdel Wahab, & Cuong-Le, 2021) | Fark (Hz) | Yüzde Hata (%) |
|-----|----------------------|-----------------------|-----------------------------------------------|-----------|----------------|
| 1   | 14,39                | 14,39                 | 14,39                                         | 0,00      | 0,02%          |
| 2   | 40,36                | 40,36                 | 40,39                                         | 0,03      | 0,08%          |
| 3   | 42,10                | 42,10                 | 42,09                                         | 0,00      | 0,02%          |
| 4   | 76,81                | 76,81                 | 76,86                                         | 0,05      | 0,07%          |
| 5   | 83,04                | 83,04                 | 83,07                                         | 0,03      | 0,04%          |
| 6   | 95,08                | 95,08                 | 95,07                                         | 0,01      | 0,01%          |

Çizelge 4.2'de görüldüğü üzere, MATLAB ve SAP2000 programlarından elde edilen frekans değerleri literatürde bildirilen frekans değerleriyle de tutarlı sonuçlar elde edilmiştir. Bu durum, geliştirilen MATLAB kodunun doğruluğunu ve analizlerde güvenle kullanılabilirliğini göstermektedir. Çizelgedeki tüm sayısal veriler, virgülden sonra iki basamak hassasiyetinde verilmiştir.

#### 4.1.3. 72 Elemanlı Uzaysal Kafes Doğrulaması

Yapısal hasar tespit algoritmalarının performansını daha karmaşık ve gerçekçi bir yapı üzerinde değerlendirmek amacıyla, literatürde kullanılan bir referans problem olan 72 elemanlı uzaysal kafes sistemi Şekil 4.3'teki gibi modellenerek incelenmiştir (Kaveh & Zolghadr, 2015). Kafes sisteminin taban boyutları her iki yönde de 1.524 metre olup, her bir kat yüksekliği 3.048 metredir. Yapıyı oluşturan tüm elemanlar için elastisite modülü (E)  $69800000 \text{ kN/m}^2$ , yoğunluk ( $\rho$ )  $27.1737 \text{ kN/m}^3$  ve kesit alanı (A)  $0.0025 \text{ m}^2$

olarak tanımlanmıştır. Ayrıca, yapının dinamik davranışını etkilemek üzere en üst katta yer alan 1, 2, 3 ve 4 numaralı düğüm noktalarına, global X, Y ve Z yönlerinde her biri 2.27 tonluk düğüm kütleler tanımlanmıştır.



Şekil 4.3. 72 elemanlı uzaysal kafesin SAP2000 modeli

**Çizelge 4.3.** İki farklı hasar senaryosu için elde edilen doğal frekans değerlerinin SAP2000, MATLAB ve referans makale sonuçları ile karşılaştırılması

| Mod | Senaryo 1    |             |                           |           | Senaryo 2    |             |                           |           |
|-----|--------------|-------------|---------------------------|-----------|--------------|-------------|---------------------------|-----------|
|     | Sap2000 (Hz) | Matlab (Hz) | (Kaveh & Zolghadr , 2015) | Fark (Hz) | Sap2000 (Hz) | Matlab (Hz) | (Kaveh & Zolghadr , 2015) | Fark (Hz) |
| 1   | 5,95         | 5,95        | 5,95                      | 0,01      | 5,94         | 5,94        | 5,95                      | 0,01      |
| 2   | 6,04         | 6,04        | 6,04                      | 0,01      | 6,04         | 6,04        | 6,04                      | 0,01      |
| 3   | 10,36        | 10,36       | 10,46                     | 0,11      | 10,36        | 10,36       | 10,46                     | 0,11      |
| 4   | 18,13        | 18,13       | 18,14                     | 0,01      | 18,08        | 18,08       | 18,09                     | 0,01      |
| 5   | 25,28        | 25,29       | 25,44                     | 0,16      | 25,04        | 25,04       | 25,20                     | 0,16      |
| 6   | 25,29        | 25,28       | 25,45                     | 0,17      | 25,29        | 25,29       | 25,45                     | 0,16      |
| 7   | 26,35        | 26,35       | 26,52                     | 0,17      | 26,28        | 26,28       | 26,45                     | 0,17      |
| 8   | 37,86        | 37,86       | 38,08                     | 0,22      | 37,77        | 37,77       | 37,99                     | 0,22      |

Çizelge 4.3'te görüldüğü gibi, her iki hasar senaryosu için geliştirilen sayısal modellerin sonuçları, referans makaledeki (Kaveh & Zolghadr, 2015) değerler ile karşılaştırılarak modelin doğruluğu teyit edilmiştir. İlk olarak, SAP2000 ve MATLAB ortamında oluşturulan modellerin birbirleriyle yüksek derecede uyumlu olduğu ve her iki senaryo için de neredeyse aynı doğal frekans değerlerini ürettiği gözlemlenmektedir. Bu durum, kullanılan sayısal modellerin iç tutarlılığını göstermektedir. Çizelgedeki tüm sayısal veriler, virgülden sonra iki basamak hassasiyetinde verilmiştir.

Referans makale ile yapılan karşılaştırmada ise, sonuçlar arasında genel bir uyum olduğu açıkça görülmektedir. Özellikle düşük modlarda (Mod 1-4) farklar oldukça küçükken, yüksek modlara doğru bu farkın bir miktar arttığı ve maksimum 0,22 Hz seviyesine ulaştığı tespit edilmiştir. Bu küçük farklılıkların, modellemede kullanılan yazılımların sayısal çözümleyici algoritmalarından veya referans çalışmada kullanılan parametrelerdeki olası küçük yuvarlamalardan kaynaklanabileceği düşünülmektedir. Sonuç olarak, elde edilen frekans değerlerinin referans çalışma ile gösterdiği yüksek uyum, oluşturulan sayısal modelin güvenilirliğini kanıtlamakta ve sonraki aşamalarda yapılacak hasar tespit analizleri için geçerli bir temel oluşturduğunu göstermektedir.

## 4.2. Hassaslık Analizleri

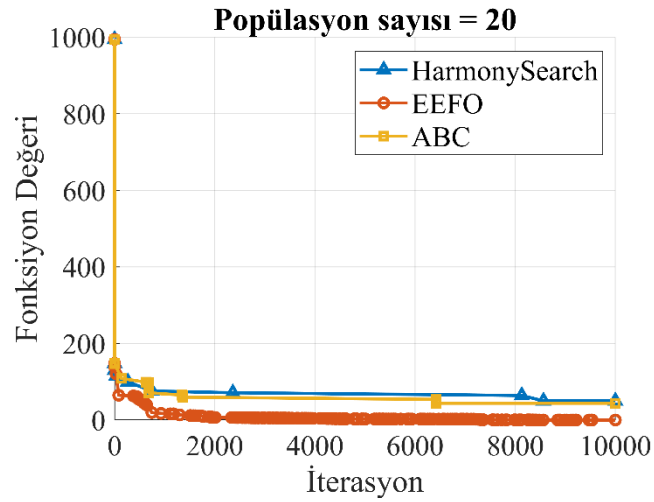
Bu bölümde, seçilen iki farklı düzlemsel kafes sistem örneği için yapılan hassaslık analizlerinin sonuçları sunulmaktadır. Analizlerde, üç farklı optimizasyon algoritması (HS, EEFO ve ABC) kullanılmış ve her algoritma için dört farklı popülasyon büyüklüğü (20, 50, 75 ve 100) ayrı ayrı değerlendirilmiştir. Amaç, algoritmaların farklı popülasyon değerleri altında gösterdiği performansı gözlemlemek ve popülasyon büyüklüğünün arama süreci üzerindeki etkisini belirlemektir. Her örnek için yapılan analizler kapsamında, algoritmaların iterasyonlara bağlı uygunluk (fitness) değerlerindeki değişimlerini gösteren çizgi grafiklerin yanı sıra, her popülasyon değeri için eleman bazında tespit edilen hasar oranlarının karşılaştırmalı olarak sunulduğu çubuk grafikler oluşturulmuştur. Bu grafiklerde, HS, EEFO ve ABC algoritmaları tarafından bulunan hasar oranları ile yapının gerçek hasar durumu birlikte gösterilmiş; böylece her algoritmanın eleman düzeyinde yaptığı tespitlerin doğruluğu ve dağılımı görsel olarak sunulmuştur. Bu çalışmada, her örnek için 10.000 iterasyon ve 10 tekrar yapılmaktadır.

### 4.2.1. 10 Elemanlı Düzlemsel Kafes

Bu kısımda, 10 elemanlı düzlemsel bir kafes yapı üzerinde gerçekleştirilen hassaslık analizlerine ait bulgular sunulmaktadır. Analizlerde, farklı meta-sezgisel yaklaşımlar kullanan üç optimizasyon algoritması dört ayrı popülasyon büyüklüğü ile değerlendirilmiştir. Bu kapsamda, literatürde sıklıkla referans gösterilen çalışmalarla uyumlu olarak iki farklı hasar senaryosu tanımlanmıştır. Birinci senaryoda, yalnızca 2 numaralı elemana %20 oranında hasar verilmiştir. İkinci senaryoda ise 7, 8, 9 ve 10 numaralı elemanlara sırasıyla %15, %20, %20 ve %25 oranlarında hasar uygulanmıştır. Bu senaryolar aracılığıyla, algoritmaların hem tekil hem de çoklu hasar durumlarındaki tespit yetenekleri incelenmiş; uygunluk fonksiyonu değerlerinin iterasyonlar boyunca değişimi ve nihai hasar oranlarının gerçek değerlerle olan uyumu grafiksel olarak sunulmuştur.

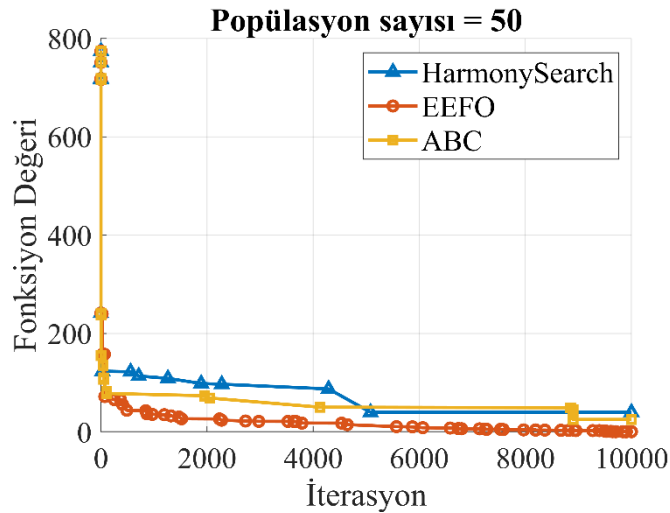
- Tekli Hasar Senaryosu

10 elemanlı düzlemsel kafes yapısında 2. elemanın %20 hasarlı olduğu durum incelenmiştir. Üç farklı optimizasyon algoritması kullanılarak, farklı popülasyon büyüklüklerinde iterasyon boyunca fitness değerlerinin değişimi ve algoritmalar tarafından tespit edilen hasar oranları karşılaştırılmıştır.



**Şekil 4.4.** Popülasyon 20 – 10 elemanlı kafesin fonksiyon değeri değişimi

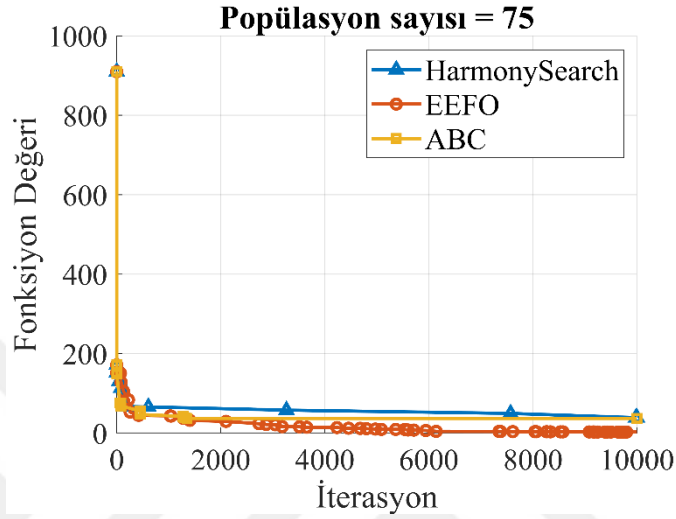
Şekil 4.4, popülasyon sayısı 20 olarak ayarlandığında üç algoritmanın yakınsama performansını göstermektedir. Her üç algoritma da başlangıç aşamasında, yaklaşık 1000'li fonksiyon değerlerinden yaklaşık 180'e kadar son derece hızlı bir düşüş sergilemiştir. Algoritmaların benzer adımlarla ilerlediği bu ilk evrenin ardından, yaklaşık 800. iterasyondan itibaren davranışlarında belirgin farklılıklar gözlemlenmektedir. HS ve ABC algoritmaları benzer hareketler sergileyerek, yaklaşık 2000. iterasyondan sonra yaptıkları birkaç küçük güncellemenin ardından neredeyse tamamen durmuştur. Bu iki algoritmanın aksine, EEFO algoritması bu noktadan sonra fonksiyon değerini kademeli olarak düşürmeye devam etmiş ve 10000 iterasyon sonunda sıfıra çok yakın bir değere ulaşmıştır.



**Şekil 4.5.** Popülasyon 50 – 10 elemanlı kafesin fonksiyon değeri değişimi

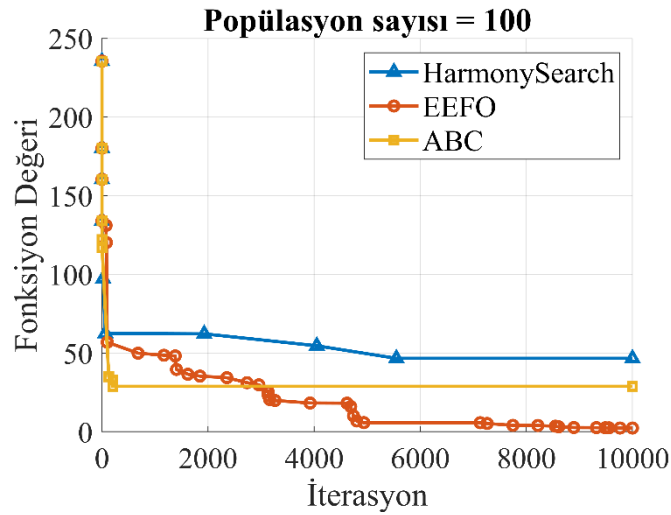
Şekil 4.5, popülasyon sayısı 50 olarak ayarlandığında üç algoritmanın yakınsama performansını sergilemektedir. ABC algoritmasının yaklaşık 4000. iterasyona kadar sık

güncellemeler yaptığı, sonrasında ise birkaç küçük güncellemenin ardından durduğu gözlemlenmektedir. Benzer bir davranışla HS algoritması da yaklaşık 5000. iterasyondan sonra yaklaşık 50'ye ulaşarak belirgin bir takılma durumu sergilemektedir. Bu iki algoritmanın aksine, EEFO algoritması baştan sona kademeli bir azalma eğilimi göstererek yerel optimuma takılmamış ve 10000 iterasyon sonunda fonksiyon değerini sıfıra çok yakın bir seviyeye indirmiştir.



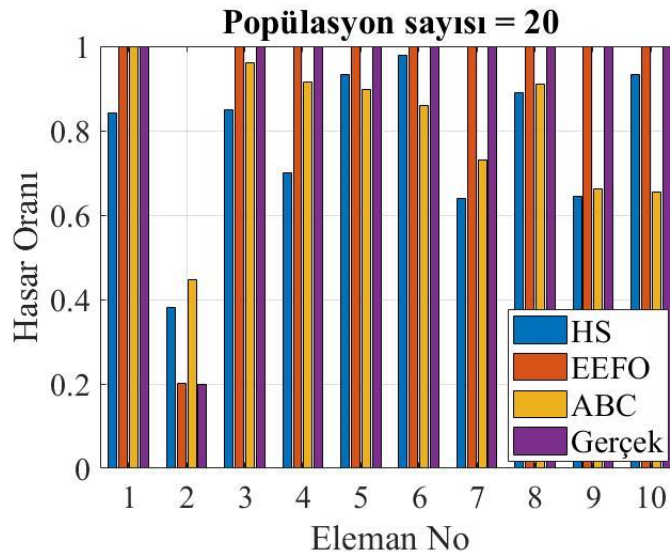
**Şekil 4.6.** Popülasyon 75 – 10 elemanlı kafesin fonksiyon değeri değişimi

Şekil 4.6, popülasyon sayısı 75 olarak ayarlandığında üç algoritmanın yakınsama performansını göstermektedir. Her üç algoritma da, başlangıçtaki yaklaşık 900 fonksiyon değerinden ilk birkaç yüz iterasyon içerisinde ani bir düşüşle oldukça düşük değerlere hızla yakınsamıştır. Bu ilk hızlı yakınsamanın ardından, HS ve ABC algoritmalarının sırasıyla yaklaşık 60 ve 40 fonksiyon değerlerine ulaşarak güncellemesini durdurduğu gözlemlenmektedir. Bu iki algoritmanın aksine, EEFO algoritması yerel optimuma takılmamış ve iterasyon süreci boyunca düşüş eğilimini sürdürmüştür.



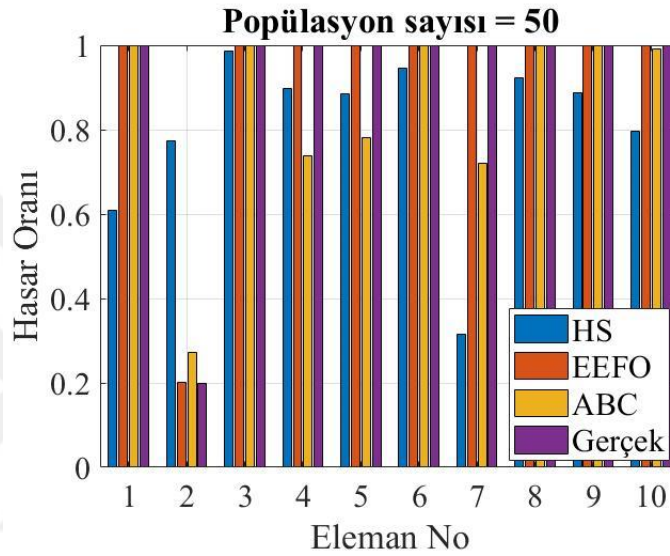
**Şekil 4.7.** Popülasyon 100 – 10 elemanlı kafesin fonksiyon değeri değişimi

Şekil 4.7, popülasyon sayısı 100 olarak ayarlandığında üç algoritmanın yakınsama performansını göstermektedir. Her üç algoritma da ilk birkaç yüz iterasyon içerisinde ani bir düşüş sergilemektedir; özellikle ABC, yaklaşık 240 fonksiyon değerinden 30'lu seviyelere dikey bir iniş yapmıştır. Bu ilk hızlı düşüşün ardından ABC algoritması, yaklaşık 30 fonksiyon değerine ulaşarak erken bir takılma sergilemektedir. HS algoritması da benzer şekilde, birkaç kademeli düşüşün ardından yaklaşık 5500. iterasyondan sonra 50'ye yakın bir değerde sabitlenmiştir. EEFO algoritması yerel optimuma takılmamış ve özellikle 4000. ile 5500. iterasyonlar arasında yaptığı belirgin düşüşle fonksiyon değerini sıfıra çok yakın bir seviyeye indirerek iterasyonlar boyunca kademeli bir düşüş göstermiştir.



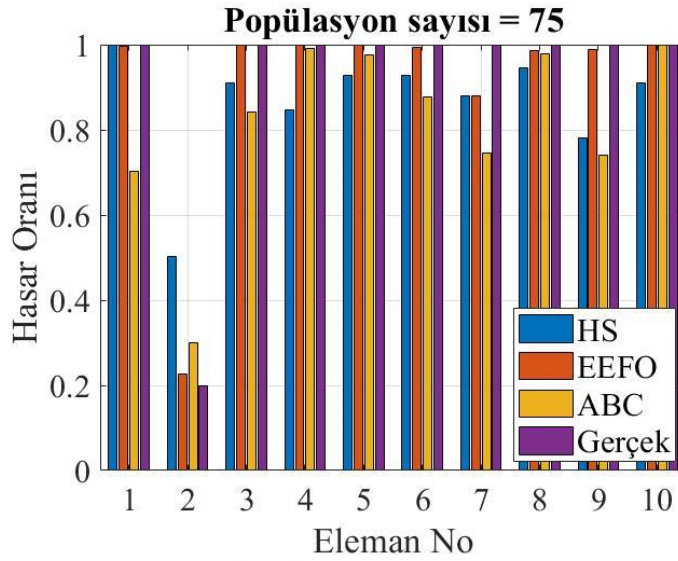
**Şekil 4.8.** Popülasyon 20 – 10 elemanlı kafes örneği için algoritmaların hasar oranı tahminleri ve gerçek değerlerin karşılaştırılması

Şekil 4.8, popülasyon sayısı 20 olarak ayarlandığında 10 elemanlı kafes örneği için algoritmaların hasar oranı tahminlerini ve gerçek değerleri karşılaştırmaktadır. EEFO algoritmasının tahminlerinin neredeyse tüm elemanlarda gerçek hasar oranlarıyla birebir örtüştüğü gözlemlenmektedir; bu durum, hem hasarlı hem de hasarsız durumlar için geçerlidir. Diğer yandan, HS ve ABC algoritmaları belirli elemanlarda gerçek değerlerden belirgin sapmalar sergilemektedir. Özellikle 2 numaralı elemanda her iki algoritma da gerçek değer üzerinde bir hasar oranı tahmin ederken; 7, 9 ve 10 numaralı elemanlar gibi tam hasarlı durumlarda ise gerçek değerin oldukça altında kalan tahminler ürettikleri tespit edilmiştir.



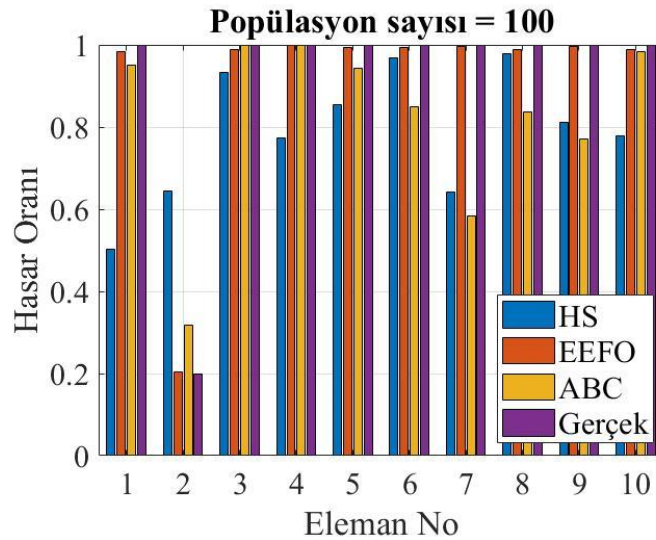
**Şekil 4.9.** Popülasyon 50 - 10 elemanlı kafes örneği için algoritmaların hasar oranı tahminleri ve gerçek değerlerin karşılaştırılması

Şekil 4.9, popülasyon sayısı 50 olarak ayarlandığında, algoritmaların hasar durumu tespit performansını gerçek değerlerle karşılaştırmaktadır. Bu senaryoda, EEFO algoritmasının tahminlerinin, hasarlı ve hasarsız durumlarda gerçek değerlerle neredeyse tam bir uyum sergilediği görülmektedir. Diğer yandan HS ve ABC algoritmaları, özellikle tam hasarlı olması gereken elemanlarda belirgin sapmalar göstermektedir. HS algoritması 2 numaralı elemanın hasarını önemli ölçüde düşük tahmin ederken, ABC algoritması ise 4 ve 5 numaralı tam hasarlı elemanlarda hasarı tespit edemeyerek 0,75'in üzerinde oranlar tahmin etmiştir.



**Şekil 4.10.** Popülasyon 75 - 10 elemanlı kafes örneği için algoritmaların hasar oranı tahminleri ve gerçek değerlerin karşılaştırılması

Şekil 4.10, popülasyon sayısı 75 olduğu durumdaki, algoritmaların hasar durumu tespit performansını incelemektedir. Özellikle hasarsız elemanların tespitinde algoritmalar arasında belirgin farklılıklar ortaya çıkmıştır. 9 numaralı hasarsız elemenda hem HS hem de ABC algoritmaları 0,8'in altında oranlar üreterek hatalı bir şekilde hasar varlığına işaret etmiştir. Benzer şekilde, 7 numaralı hasarsız elemenda HS ve EEFO algoritmaları yaklaşık 0,9 oranında hasar tespiti yapmıştır. %20 oranında hasarlı olan 2 numaralı elemenda EEFO, gerçek değere en yakın tahmini sunmuş, HS ve ABC ise hasar miktarını olduğundan daha az olarak raporlamıştır.

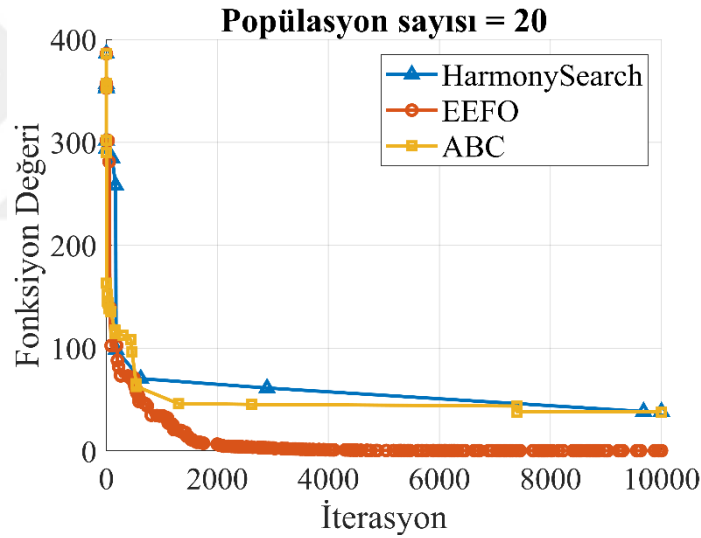


**Şekil 4.11.** Popülasyon 100 - 10 elemanlı kafes örneği için algoritmaların hasar oranı tahminleri ve gerçek değerlerin karşılaştırılması

Şekil 4.11, popülasyon sayısı 100 olarak ayarlandığında algoritmaların hasar durumu tespit performansını göstermektedir. Bu senaryoda, EEFO algoritmasının tahminleri hem %20 hasarlı olan 2 numaralı elemenda hem de hasarsız elemanların çoğunda gerçek değerlerle yüksek bir uyum içerisindedir. Buna karşılık, HS ve ABC algoritmaları özellikle hasarsız olması beklenen elemanlarda sapmalar göstermiştir. Örneğin, HS algoritması 1, 4 ve 7 numaralı hasarsız elemanlarda 0.8'in altında hasar oranları üretmiş; ABC algoritması ise 7, 8 ve 9 numaralı hasarsız elemanlarda çeşitli oranlarda hasar tespiti yapmıştır. Ayrıca, 2 numaralı hasarlı elemenda EEFO'nun tahmini gerçek değere oldukça yakınken, HS ve ABC algoritmalarının tahminleri hasar oranını olduğundan daha yüksek göstermiştir.

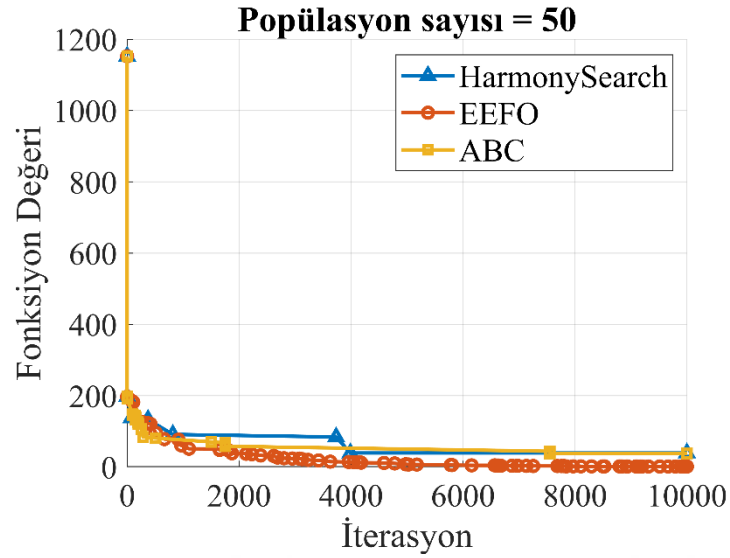
- Çoklu Hasar Senaryosu

Bu bölümde, tekli hasar durumundan farklı olarak, 10 elemanlı aynı düzlemsel kafes yapısında birden fazla elemanın hasar gördüğü daha karmaşık bir senaryo ele alınmaktadır: 7, 8, 9 ve 10. elemanlar sırasıyla %15, %20, %20 ve %25 oranlarında hasarlıdır.



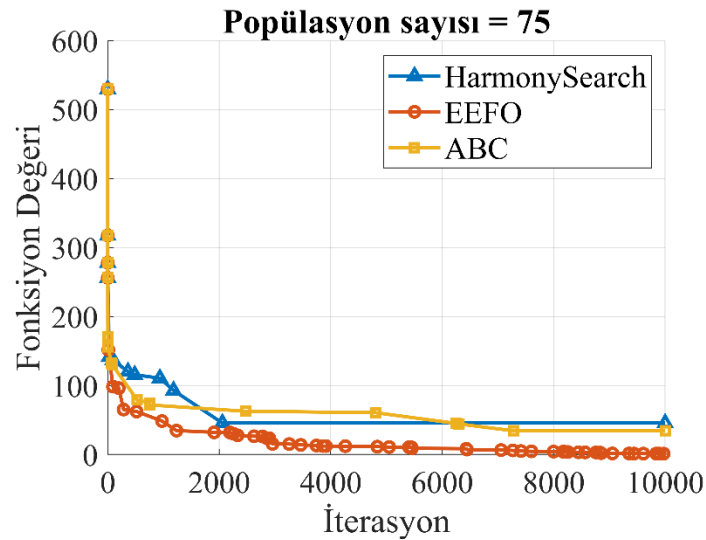
**Şekil 4.12.** Popülasyon 20 – 10 elemanlı kafesin fonksiyon değeri değişimi

Şekil 4.12'de, popülasyon sayısı 20 olarak ayarlandığında, üç algoritmanın yakınsama davranışları belirgin farklılıklar göstermektedir. Üç algoritma da yaklaşık 300 değerinden yaklaşık 100 değerine ani düşüş yapmışlardır. HS ve ABC algoritmaları yaklaşık 50 fonksiyon değerinde takılı kalmışlardır. Buna karşın EEFO algoritması, ilk keskin düşüşten sonra yerel optimuma takılmamış ve 10000 iterasyon boyunca yavaş fakat istikrarlı ve kademeli bir düşüş eğilimini sürdürerek diğer iki algoritmadan daha düşük bir fonksiyon değerine ulaşmıştır.



**Şekil 4.13.** Popülasyon 50 – 10 elemanlı kafesin fonksiyon değeri değişimi

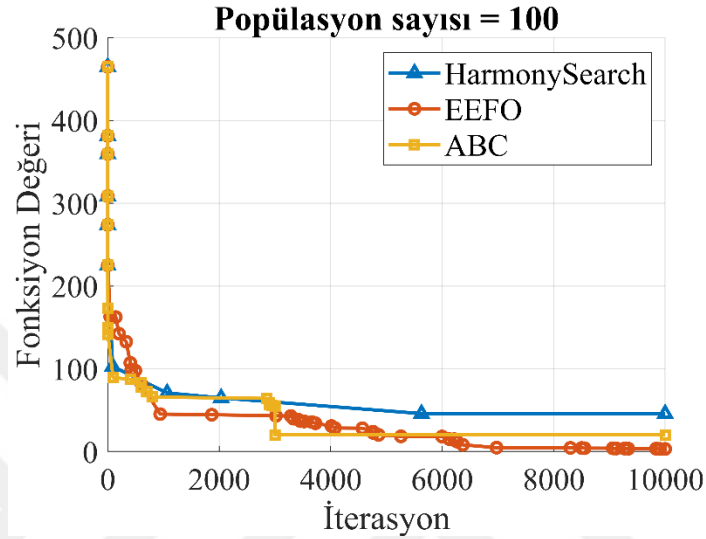
Popülasyon büyüklüğünün 50'ye çıkarıldığı Şekil 4.13'te, tüm algoritmaların başlangıç fonksiyon değerleri artmıştır. Her üç algoritma da ilk 500 iterasyon içerisinde son derece hızlı bir yakınsama göstermiştir. Özellikle HS ve EEFO, fonksiyon değerini 100'ün altına çok hızlı bir şekilde indirmiştir. İlk hızlı düşüşün ardından HS ve ABC algoritması 50 civarında bir değere ulaşarak belirgin bir takılma göstermiştir. Buna karşın, EEFO algoritması ilk hızlı düşüşten sonra yavaş ve istikrarlı bir şekilde güncellemeye devam etmiş ve iterasyon sonunda yaklaşık 25 gibi bir değerle en düşük fonksiyon değerini elde etmiştir.



**Şekil 4.14.** Popülasyon 75 – 10 elemanlı kafesin fonksiyon değeri değişimi

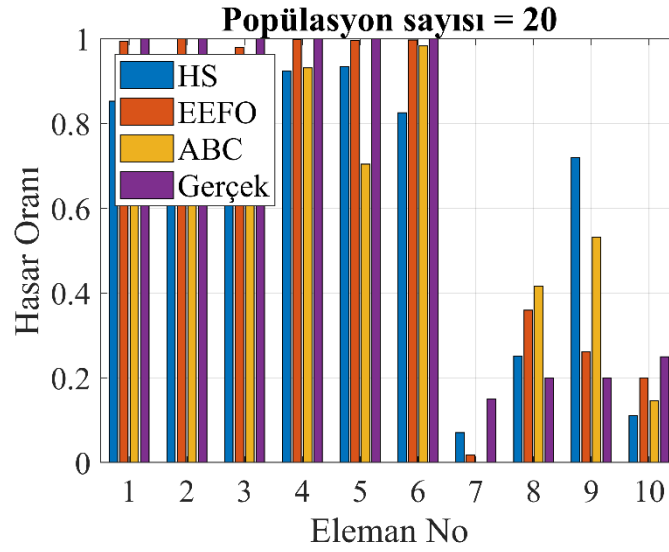
Şekil 4.14'te, popülasyon sayısının 75 olduğu bu senaryoda, her üç algoritma da yaklaşık 550 gibi yüksek bir fonksiyon değerinden başlayarak ilk 2000 iterasyon

içerisinde keskin bir düşüş sergilemektedir. Bu ilk hızlı yakınsama evresinin ardından algoritmaların davranışları belirgin bir şekilde farklılaşmaktadır. HS ve ABC algoritmaları, sırasıyla yaklaşık 2000. ve 6000. iterasyonlardan sonra 50 civarında bir fonksiyon değerine ulaşarak takılı kalma durumu göstermiştir. Bu iki algoritmanın aksine, EEFO algoritması ilk ani düşüşün ardından yerel optimuma takılmadan, kademeli bir şekilde fonksiyon değerini düşürmeye devam etmiştir.



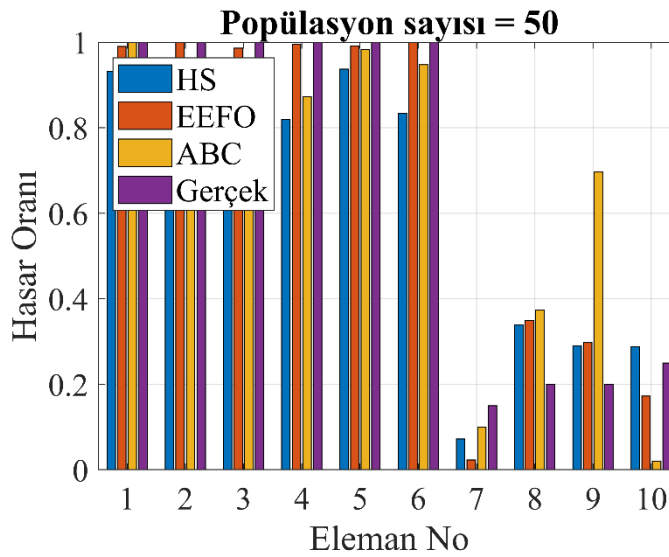
**Şekil 4.15.** Popülasyon 100 – 10 elemanlı kafesin fonksiyon değeri değişimi

Popülasyon sayısının 100 olarak belirlendiği Şekil 4.15'te, üç algoritma da yaklaşık 480 fonksiyon değerinden başlayarak ilk 1000 iterasyon içerisinde ani bir düşüş sergilemiştir. Bu hızlı yakınsamanın ardından, ABC algoritması 3000. iterasyon civarında yaklaşık 20 fonksiyon değerine ulaşmış ve sonrasında değerini sabit tutmuştur. HS algoritması ise 2000. iterasyondan sonra sınırlı sayıda güncelleme yapmış ve fonksiyon değeri daha yüksek seviyelerde kalmıştır. EEFO algoritması, 1000. iterasyondan itibaren güncellenmeye devam etmiş ve iterasyonlar ilerledikçe fonksiyon değerinde düzenli bir azalma gözlemlenmiştir.



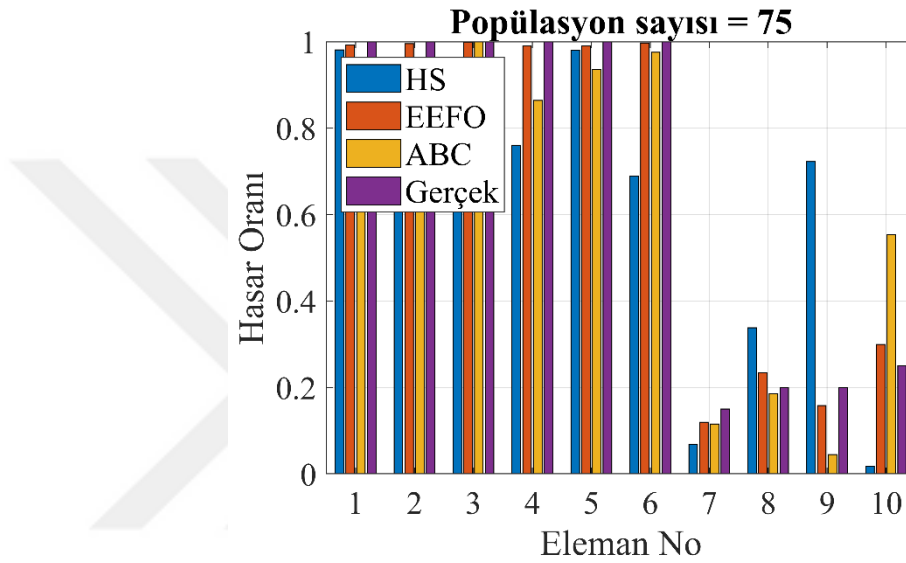
**Şekil 4.16.** Popülasyon 20 - 10 elemanlı kafes örneği için algoritmaların hasar oranı tahminleri ve gerçek değerlerin karşılaştırılması

Şekil 4.16'da, popülasyon sayısı 20 olarak ayarlandığında, algoritmaların 10 farklı eleman için hasar oranı tahminleri sunulmaktadır. 1'den 6'ya kadar olan elemanlarda her üç algoritma da genel olarak gerçek hasar oranlarına yakın tahminler üretmiştir. 7. elemandan itibaren ise algoritmaların performansları farklılaşmaktadır; 8. elemanda tüm algoritmalar gerçek değerin üzerinde tahmin yaparken, en belirgin sapma 9. elemanda gözlemlenmiştir. Bu elemanda gerçek hasar oranı yaklaşık 0,2 iken, HS yaklaşık 0,73'lük bir tahminle en büyük sapmayı göstermiş, EEFO ise gerçek değere en yakın sonucu üretmiştir. 10. elemanda ise EEFO'nun tahmini gerçek değere en yakınken, diğer iki algoritmanın tahminleri daha düşük seviyelerde kalmıştır.



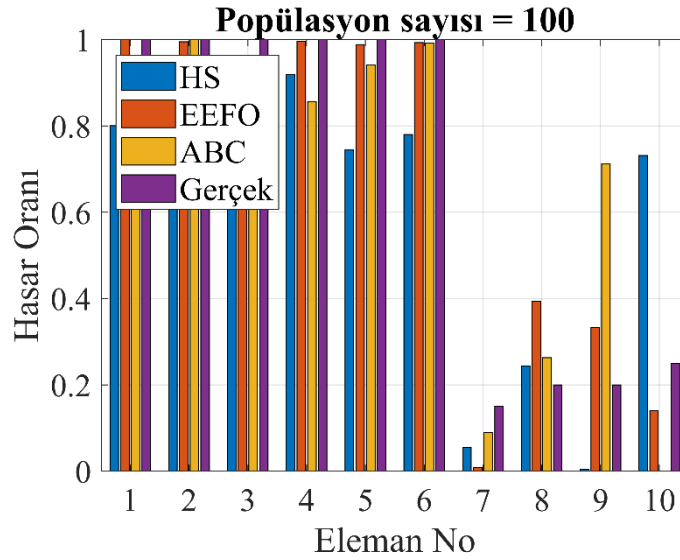
**Şekil 4.17.** Popülasyon 50 - 10 elemanlı kafes örneği için algoritmaların hasar oranı tahminleri ve gerçek değerlerin karşılaştırılması

Şekil 4.17'de, popülasyon sayısının 50 olarak ayarlandığı durum göstermektedir. Hasarsız elemanlarda her üç algoritma gerçek hasar oranlarına yakın tahminler sunmaktadır. 7. elemanda tüm algoritma tahminleri gerçek değerin altında kalmışken, 8. elemanlarda tüm algoritmaların tahminleri gerçek değerin üzerinde kalmıştır. En belirgin sapma 9. elemanda gözlemlenmektedir; bu elemanda gerçek hasar oranı 0,2 iken, HS ve EEFO'nun tahminleri bu değere yaklaşmıştır, ABC'nin tahmini yaklaşık 0,7 gibi oldukça yüksek bir değerle büyük bir farklılık sergilemektedir. 10. elemanda ise HS ve EEFO gerçek değere daha yakın, ABC ise gerçek değerden uzak kalarak sıfıra yakın bir değer tahmin etmiştir.



**Şekil 4.18.** Popülasyon 75 - 10 elemanlı kafes örneği için algoritmaların hasar oranı tahminleri ve gerçek değerlerin karşılaştırılması

Şekil 4.18 incelendiğinde, popülasyon sayısı 75 olarak belirlendiğinde, önceki popülasyon sayılarına benzer şekilde hasarsız elemanlar için tüm algoritmalar gerçek hasar oranlarına yakın tahminler üretmiştir. Ancak 7. elemandan itibaren algoritmaların tahminleri arasında farklılaşma gözlemlenmektedir. En belirgin sapma 9. elemanda ortaya çıkmaktadır. Bu noktada, gerçek hasar oranı yaklaşık 0,2 iken, HS algoritması yaklaşık 0,7 değerinde bir tahmin yapmış, EEFO algoritması gerçek değere yakın bir sonuç üretmiş, ABC algoritmasının tahmini ise sıfıra yakın bir seviyede kalmıştır. Benzer bir durum 10. eleman için de geçerlidir; bu elemanda ABC algoritmasının tahmini gerçek değerin üzerinde kalırken, HS algoritmasının tahmini sıfıra yakın gerçekleşmiştir.



**Şekil 4.19.** Popülasyon 100 - 10 elemanlı kafes örneği için algoritmaların hasar oranı tahminleri ve gerçek değerlerin karşılaştırılması

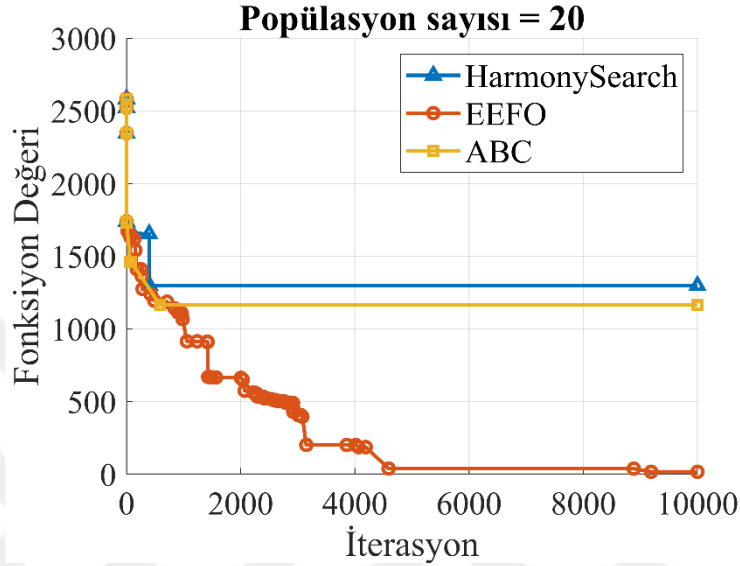
Şekil 4.19, popülasyon sayısı 100 olarak ayarlandığında algoritmaların 10 farklı eleman için sunduğu hasar oranı tahminlerini gerçek değerlerle karşılaştırmaktadır. Hasar almayan elemanlardaki tahminler önceki popülasyon sayılarında olduğu gibi çok uzak bulunmamıştır. Ancak diğer elemanlarda algoritmaların performansları belirgin şekilde farklılaşmaktadır. Özellikle 9. ve 10. elemanlarda büyük sapmalar gözlemlenmektedir; 9. elemanda gerçek hasar oranı yaklaşık 0,2 iken ABC'nin tahmini 0,7'nin üzerinde bir değerle, 10. elemanda ise yine gerçek oran yaklaşık 0,25 iken HS'nin tahmini 0,7'nin üzerinde bir değerle büyük bir farklılık sergilemektedir. 8. elemanda da EEFO'nun tahmini gerçek değerın yaklaşık iki katı seviyesindedir.

#### 4.2.2. 72 Elemanlı Uzaysal Kafes

Geliştirilen hasar tespit yönteminin etkinliğini ve sınırlarını daha kapsamlı bir şekilde incelemek için, bu bölümde literatürde kullanılan 72 elemanlı uzaysal kafes yapısı başka bir örnek olarak seçilmiştir. Önceki düzlemsel örneğin aksine, bu üç boyutlu ve çok elemanlı yapı, optimizasyon algoritmalarının yüksek boyutlu arama uzaylarındaki performansını ve gerçeğe daha yakın mühendislik problemlerini çözme kapasitesini test etmek için ideal bir platform sunmaktadır. Bu amaçla, yapı üzerinde, literatürdeki çalışmaya paralel olarak iki farklı hasar senaryosu ele alınacaktır: Birincisi, sadece 55 numaralı elemanda %15 oranında hasarın bulunduğu tekli hasar durumu; ikincisi ise 4 numaralı elemanda %10 ve 58 numaralı elemanda %15 oranlarında hasarın mevcut olduğu çoklu hasar durumudur. Daha önce incelenen optimizasyon algoritmaları, bu iki hasar senaryosu için uygulanarak performansları ve hasar tespit yetenekleri değerlendirilecektir.

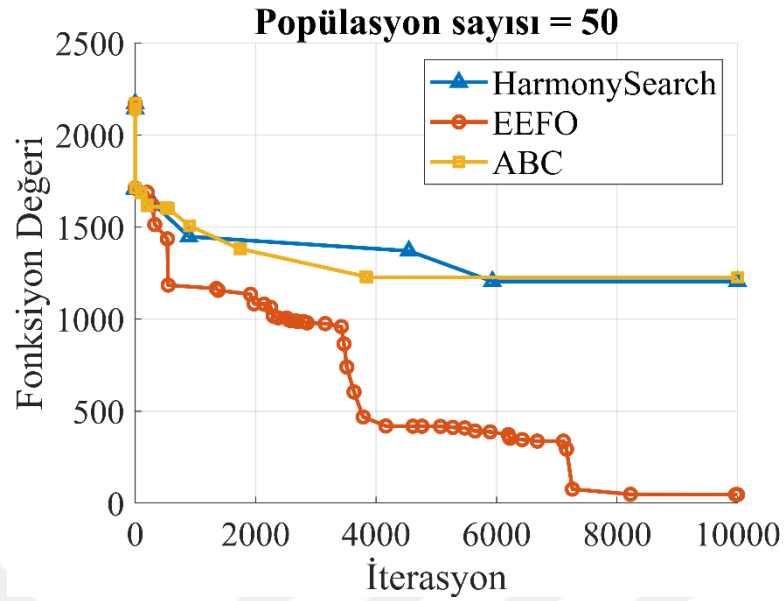
- Tekli Hasar Senaryosu

Bu bölümde, 72 elemanlı uzaysal kafes sistemi için tanımlanan ilk hasar durumu olan tekli hasar senaryosu incelenmektedir. Bu kapsamda, 55 numaralı elemanda %15 oranında hasarın bulunduğu durum için optimizasyon algoritmalarının iterasyon boyunca fitness değerlerindeki değişim ve hasar tespit yetenekleri analiz edilecektir.



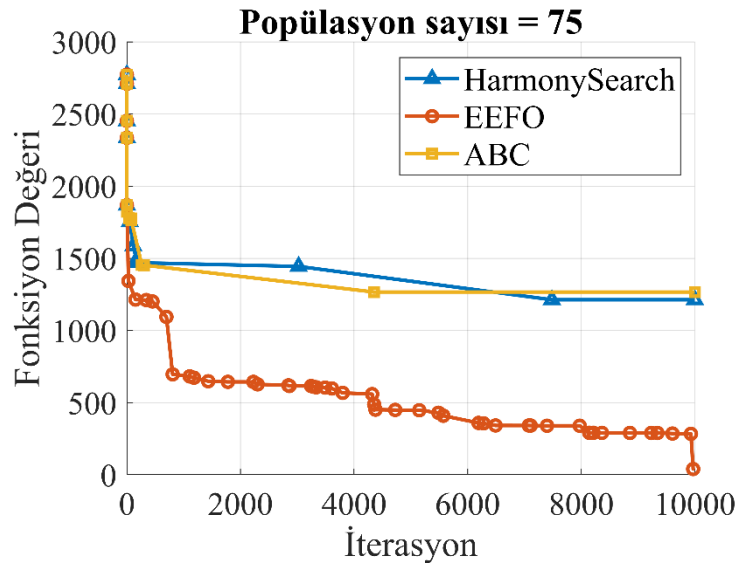
**Şekil 4.20.** Popülasyon 25 – 72 elemanlı kafesin fonksiyon değeri değişimi

Şekil 4.20, 72 elemanlı kafes problemi için üç algoritmanın yakınsama performansını sergilemektedir. Her üç algoritma da başlangıçta hızlı bir düşüş gösterse de, bu ilk evrenin hemen ardından davranışları belirgin bir şekilde farklılaşmaktadır. HS ve ABC algoritmaları, ilk birkaç yüz iterasyondan sonra sırasıyla yaklaşık 1300 ve 1150 fonksiyon değerlerine ulaşarak çok erken bir aşamada belirgin bir duraksamaya uğramaktadır. Bu iki algoritmanın aksine, EEFO algoritması kademeli ve basamaklı bir düşüş profili sergilemektedir. Fonksiyon değeri yaklaşık 4500 ile 9000. iterasyonlar arasında bir süre sabit kalmış, ancak bu duraksamanın ardından gerçekleşen güncellemelerle düşüş eğilimini sürdürmüştür.



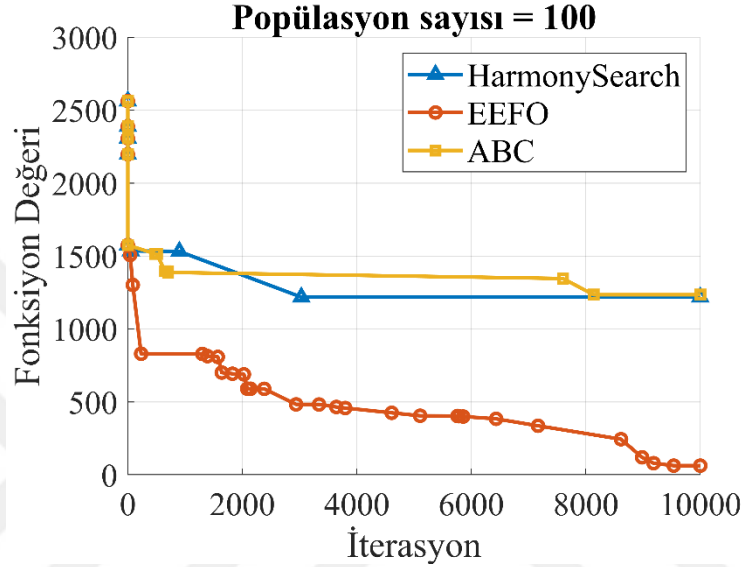
**Şekil 4.21.** Popülasyon 50 – 72 elemanlı kafesin fonksiyon değeri değişimi

Şekil 4.21, popülasyon sayısı 50 olarak ayarlandığında 72 elemanlı kafes problemi için üç algoritmanın yakınsama performansını göstermektedir. HS ve ABC algoritmaları, sırasıyla yaklaşık 6000. ve 4000. iterasyonlardan sonra 1200 civarında yüksek bir fonksiyon değerinde takılarak, yerel optimumda sıkışma eğilimi göstermiştir. Bu iki algoritmanın aksine, EEFO algoritması kademeli ve basamaklı bir düşüş profili sergilemektedir. Özellikle 3500. ve 7500. iterasyonlar arasında sergilediği ani ve keskin düşüşlerle yerel optimumlardan kurtulmayı başaran EEFO, fonksiyon değerini sıfıra çok yakın bir seviyeye indirerek diğer iki algoritmadan önemli ölçüde daha iyi bir yakınsama performansı ortaya koymuştur.



**Şekil 4.22.** Popülasyon 75 – 72 elemanlı kafesin fonksiyon değeri değişimi

Şekil 4.22, popülasyon sayısı 75 olarak ayarlandığında 72 elemanlı kafes problemi için üç algoritmanın yakınsama performansını göstermektedir. HS ve ABC algoritmaları, sırasıyla yaklaşık 1450 ve 1250 civarındaki fonksiyon değerlerine ulaştıktan sonra ilerleme kaydedememiş ve çözüm uzayında durağan bir seyir izlemiştir. EEFO'nun yaklaşık 10000. iterasyonda 300 civarındaki bir fonksiyon değerinde belirgin bir takılma yaşadığı görülmektedir. Algoritma, arama sürecinin sonuna doğru, 10000. iterasyonda ani bir düşüşle kurtularak fonksiyon değerini sıfıra çok yakın bir seviyeye indirmiştir.

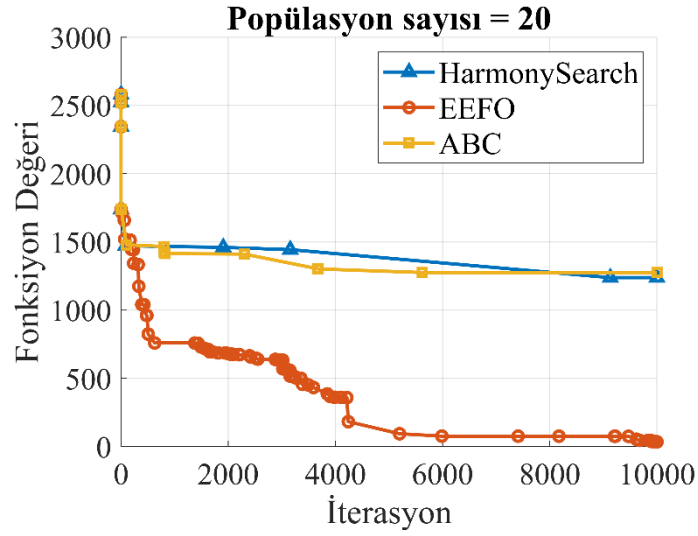


Şekil 4.23. Popülasyon 100 – 72 elemanlı kafesin fonksiyon değeri değişimi

Şekil 4.23, popülasyon sayısı 100 olarak ayarlandığında 72 elemanlı kafes problemi için üç algoritmanın yakınsama performansını göstermektedir. HS ve ABC algoritmaları, başlangıçtaki düşüşten sonra sırasıyla yaklaşık 1250 ve 1400 fonksiyon değeri civarında yerel minimumlarda takılı kalmıştır. EEFO yaklaşık 500. iterasyondan 1800. iterasyona kadar fonksiyon değerinin 850 civarında sabitlendiği geçici bir takılma evresi yaşamıştır. Bu platodan kurtulduktan sonra, iterasyon süreci boyunca basamaklı bir düşüş yapıyla ilerlemeye devam etmiş, özellikle 6000. iterasyondan sonra daha belirgin bir ivme kazanarak fonksiyon değerini sıfıra çok yakın bir seviyeye indirmiştir.

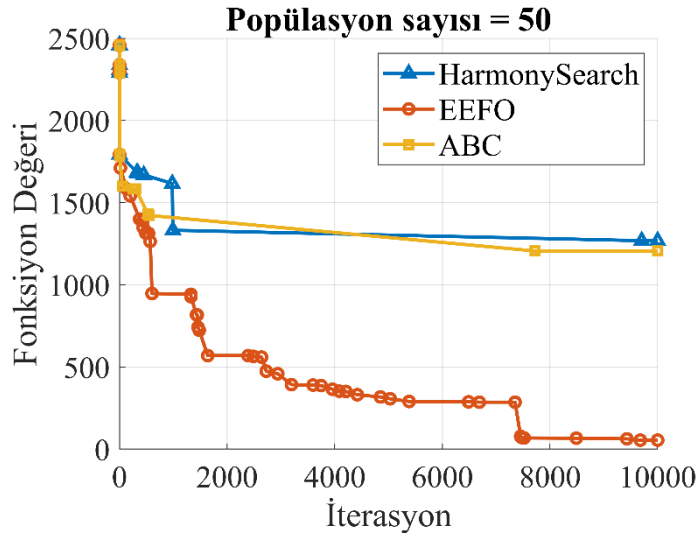
- Çoklu Hasar Senaryosu

72 elemanlı uzaysal kafes sistemi için ikinci hasar durumu olarak, yapıda birden fazla elemanın hasar gördüğü çoklu hasar senaryosu bu alt bölümde incelenmektedir. Bu senaryo kapsamında, 4 numaralı elemanda %10 ve 58 numaralı elemanda %15 oranlarında hasarın bulunduğu durum için optimizasyon algoritmalarının iterasyon boyunca fonksiyon değerlerindeki değişim ve hasar tespit yetenekleri analiz edilmiştir.



**Şekil 4.24.** Popülasyon 20 – 72 elemanlı kafesin fonksiyon değeri değişimi

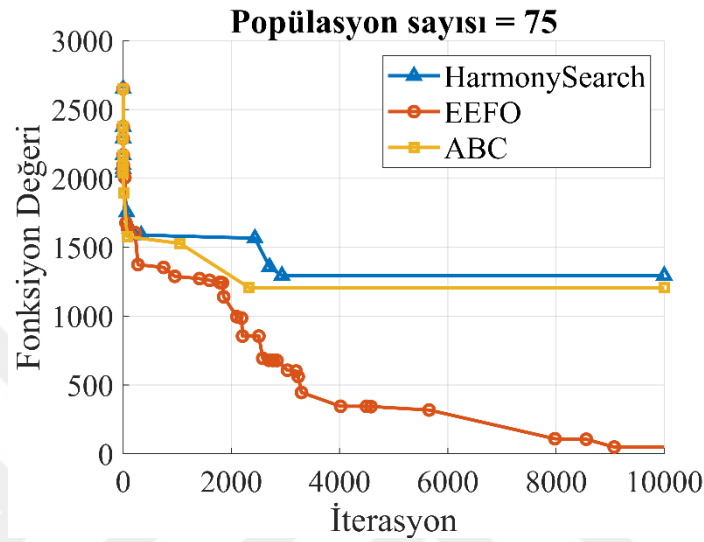
Şekil 4.24, popülasyon sayısı 20 olarak ayarlandığında 72 elemanlı kafes problemi için üç algoritmanın yakınsama performansını göstermektedir. HS ve ABC algoritmaları, sırasıyla yaklaşık 1450 ve 1300 fonksiyon değerlerine ulaşarak erken bir aşamada takılarak oldukça uzak değerler elde etmiştir. EEFO, özellikle 4000. iterasyona kadar sergilediği belirgin düşüşlerle fonksiyon değerini sıfıra çok yakın bir seviyeye indirmiş ve diğer iki algoritmadan önemli ölçüde daha iyi bir yakınsama performansı ortaya koymuştur.



**Şekil 4.25.** Popülasyon 50 – 72 elemanlı kafesin fonksiyon değeri değişimi

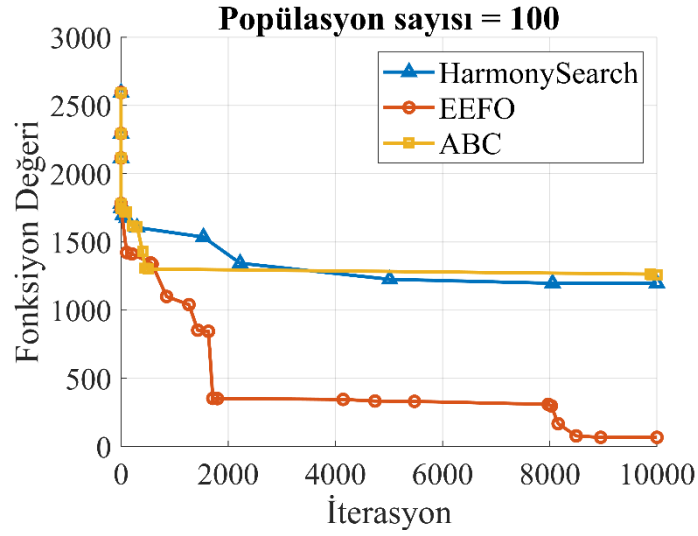
Şekil 4.25, popülasyon sayısı 50 olarak ayarlandığında 72 elemanlı kafes problemi için üç algoritmanın yakınsama performansını göstermektedir. HS ve ABC algoritmaları, sırasıyla yaklaşık 1300 ve 1200 fonksiyon değerlerine ulaşmıştır. Yaklaşık 1000 iterasyondan sonra belirgin güncellemeler yapamadan iterasyon bitene kadar sabit

kalmıştır. EEFO ilk hızlı düşüşün ardından yaklaşık 500. ile 1500. iterasyonlar arasında 950 civarındaki bir fonksiyon değerinde geçici bir takılma yaşamaktadır. Bu noktadan kurtulduktan sonra, yaklaşık 7000. iterasyona kadar yavaş ve kademeli bir düşüşle 300'lü seviyelere inmektedir. Bu uzun yavaşlama evresinin ardından, özellikle 7500. iterasyon civarında sergilediği ani ve keskin bir düşüşle fonksiyon değerini sıfıra çok yakın bir seviyeye indirmiştir.



**Şekil 4.26.** Popülasyon 75 – 72 elemanlı kafesin fonksiyon değeri değişimi

Şekil 4.26, popülasyon sayısı 75 olarak ayarlandığında 72 elemanlı kafes problemi için üç algoritmanın yakınsama performansını göstermektedir. HS algoritması yaklaşık 1300 değerinden sonra yaklaşık 3000 iterasyondan sonra güncellemesini durdurmuştur. ABC algoritması, sırasıyla 1200 fonksiyon değerlerin ulaşarak erken bir aşamada belirgin bir takılmaya uğramaktadır. EEFO iterasyon süreci boyunca yerel optimumlardan kaçınarak sürekli fonksiyon değerini sıfıra çok yakın bir seviyeye indirmiştir.



**Şekil 4.27.** Popülasyon 100 – 72 elemanlı kafesin fonksiyon değeri değişimi

Şekil 4.27, popülasyon sayısının 100 olarak belirlendiği 72 elemanlı kafes sistemine ait yakınsama performanslarını göstermektedir. HS ve ABC algoritmaları, başlangıçta hızlı düşüş sergilemelerine rağmen, sırasıyla yaklaşık 1200 ve 1250 fonksiyon değerlerinde durağanlaşmış ve ilerleyen iterasyonlarda kayda değer bir iyileşme sağlayamamıştır. Buna karşılık, EEFO algoritması yaklaşık 1800. iterasyonda ani bir iyileşme ile fonksiyon değerini 350 seviyelerine kadar düşürmüştür. Yaklaşık 8000. iterasyona kadar sınırlı değişim göstermiştir. Sonrasında ise kalan iterasyonlar boyunca düşüş eğilimini sürdürerek fonksiyon değerini sıfıra oldukça yakın bir seviyeye kadar indirmiştir.

### 4.3. İstatistiksel Sonuçların Karşılaştırılması

Algoritmaların yakınsama performanslarının ötesinde, etkinliklerini ve tutarlılıklarını nicel olarak ölçmek amacıyla bu bölümde istatistiksel sonuçlar karşılaştırılacaktır. 10 tekrarlı çalıştırmalardan elde edilen ortalama, standart sapma ve en iyi amaç fonksiyonu değerleri kullanılarak algoritmaların genel performansı değerlendirilecektir. Bu kapsamda, öncelikle 10 elemanlı düzlemsel kafes, ardından ise 72 elemanlı uzaysal kafes problemleri için elde edilen istatistiksel veriler ilgili alt başlıklar altında sunulup analiz edilecektir. Veri sunumunda tutarlılığı sağlamak amacıyla, bu bölümde yer alan çizelgelerdeki sayısal değerlerin hassasiyeti verinin türüne göre belirlenmiştir. Algoritmaların genel performansını özetleyen ve Çizelge 4.4, 4.6, 4.11 ile 4.13'te sunulan istatistiksel değerler, virgülden sonra iki basamak hassasiyetle sunulurken; eleman bazında daha hassas bir ölçüm gerektiren ve Çizelge 4.5, 4.7–4.10, 4.12 ile 4.14'te gösterilen hasar oranları, virgülden sonra üç basamak hassasiyetle verilmiştir.

**Çizelge 4.4.** 10 Elemanlı Kafes (Senaryo 1): Algoritma performans istatistikleri

| Pop. | HS       |            |        | EEFO     |            |        | ABC      |            |        |
|------|----------|------------|--------|----------|------------|--------|----------|------------|--------|
|      | Ortalama | Std. Sapma | En iyi | Ortalama | Std. Sapma | En iyi | Ortalama | Std. Sapma | En iyi |
| 20   | 59,87    | 8,65       | 50,01  | 10,00    | 10,99      | 0,08   | 51,15    | 4,53       | 43,89  |
| 50   | 53,79    | 11,37      | 39,91  | 4,81     | 4,50       | 0,43   | 48,29    | 10,36      | 25,09  |
| 75   | 60,09    | 11,04      | 38,88  | 11,35    | 9,47       | 2,80   | 50,16    | 10,07      | 36,97  |
| 100  | 63,71    | 9,45       | 46,71  | 8,32     | 5,49       | 2,44   | 47,74    | 10,58      | 28,92  |

Çizelge 4.4'te, senaryo 1 kapsamında, farklı popülasyon boyutları altında HS, EEFO ve ABC algoritmalarının performans istatistiklerini özetlemektedir. Tablodaki veriler incelendiğinde, EEFO algoritmasının ortalama fonksiyon değerleri (4,81 ile 11,35 arasında) ve en iyi fonksiyon değerleri ile diğer iki algortmadan belirgin şekilde daha düşük sonuçlar elde ettiği görülmektedir. Buna karşın, HS ve ABC algoritmaları genel olarak birbirlerine daha yakın performans sergilemekte olup, ortalama değerleri çoğunlukla 47 ile 64 aralığında seyretmektedir. Standart sapma değerleri ise tüm algoritmalar için popülasyon büyüklüğüne göre değişkenlik göstermekte ve belirli bir algoritmanın tüm senaryolarda tutarlı olarak en düşük sapmayı sergilediği gözlemlenmemektedir.

**Çizelge 4.5.** 10 Elemanlı Kafes (Senaryo 1): Algoritmaların hasar durumları

| Pop. | Gerçek Hasar Oranı | Ölçülen Hasar Oranı |       |       |
|------|--------------------|---------------------|-------|-------|
|      |                    | HS                  | EEFO  | ABC   |
| 20   | 0,200              | 0,381               | 0,200 | 0,445 |
| 50   | 0,200              | 0,772               | 0,200 | 0,271 |
| 75   | 0,200              | 0,501               | 0,227 | 0,300 |
| 100  | 0,200              | 0,644               | 0,204 | 0,318 |

Çizelge 4.5, senaryo 1 kapsamında, farklı popülasyon boyutlarında algoritmaların 0,2'lik gerçek hasar oranını tahmin etme performansını göstermektedir. EEFO algoritması tüm popülasyon senaryolarında gerçek hasar oranına son derece yakın veya birebir aynı sonuçlar üretmiştir. Buna karşın, HS ve ABC algoritmaları, incelenen tüm popülasyon boyutlarında gerçek değeri belirgin şekilde üzerinde tahmin etmiştir. Özellikle HS algoritması, 50 popülasyonlu senaryoda 0,772 gibi bir değerle en büyük sapmayı gösterirken, ABC algoritmasının tahminleri de 0,271 ile 0,445 arasında değişerek gerçek değerden sapmıştır.

**Çizelge 4.6.** 10 Elemanlı Kafes (Senaryo 2): Algoritma performans istatistikleri

| Pop. | HS       |            |        | EEFO     |            |        | ABC      |            |        |
|------|----------|------------|--------|----------|------------|--------|----------|------------|--------|
|      | Ortalama | Std. Sapma | En iyi | Ortalama | Std. Sapma | En iyi | Ortalama | Std. Sapma | En iyi |
| 20   | 62,16    | 13,51      | 38,52  | 6,02     | 11,63      | 0,50   | 52,26    | 7,58       | 38,09  |
| 50   | 62,75    | 14,35      | 39,16  | 3,40     | 1,10       | 1,20   | 50,22    | 8,24       | 37,41  |
| 75   | 59,89    | 9,10       | 45,95  | 3,20     | 1,28       | 1,43   | 44,78    | 7,77       | 34,65  |
| 100  | 60,20    | 9,32       | 46,01  | 5,89     | 2,07       | 3,35   | 52,90    | 13,86      | 20,28  |

Çizelge 4.6, 10 elemanlı yapının çoklu hasar senaryosuna ait istatistiksel sonuçları özetlemektedir. İncelenen tüm popülasyon senaryolarında EEFO algoritmasının ortalama fonksiyon değerleri (3,20 ile 6,02 arasında) ve en iyi fonksiyon değerleri (0,50 ile 3,35 arasında), HS ve ABC algoritmalarının sonuçlarından belirgin ölçüde daha düşüktür. Standart sapma değerleri incelendiğinde ise EEFO algoritması, 20 popülasyonlu senaryo hariç, diğer popülasyon boyutlarında HS ve ABC'ye kıyasla çok daha düşük değerler sergilemekte; bu durum, ilgili popülasyonlarda EEFO'nun daha tutarlı sonuçlar ürettiğini göstermektedir. HS ve ABC algoritmaları ise genel olarak birbirlerine daha yakın ortalama değerler sunarken, ABC'nin 100 popülasyon boyutunda elde ettiği 20,28'lik en iyi değer, bu iki algoritma arasındaki en düşük sonuçtur.

**Çizelge 4.7.** 10 Elemanlı Kafes (Senaryo 2, Popülasyon 20): Algoritmaların hasar durumları

| Eleman No | Gerçek Hasar Oranı | Ölçülen Hasar Oranı |       |       |
|-----------|--------------------|---------------------|-------|-------|
|           |                    | HS                  | EEFO  | ABC   |
| 7         | 0,150              | 0,071               | 0,017 | 0,000 |
| 8         | 0,200              | 0,251               | 0,359 | 0,416 |
| 9         | 0,200              | 0,719               | 0,261 | 0,532 |
| 10        | 0,250              | 0,111               | 0,199 | 0,146 |

Çizelge 4.7'de popülasyon sayısı 20 olduğunda, çoklu hasar senaryosunda algoritmaların hasar şiddetini tahmin etme yeteneklerini ortaya koymaktadır. Her üç algoritmanın da en düşük hasara sahip olan 7. elemanda hasarı tespit edememiş ve sıfıra çok yakın değerler bulmuştur. Diğer hasarlı elemanlarda (8, 9 ve 10) ise, algoritmalar hasarın varlığını belirlemiş ancak miktarını tahmin etmede değişken bir performans sergilemiştir. 8. elemanda, her üç algoritma da gerçek değer üzerinde tahminler sunmuş, en büyük sapmayı 0,416 değeriyle ABC göstermiştir. En belirgin farklılık 9. elemanda gözlemlenmektedir; burada gerçek değer 0,2 iken, HS ve ABC oldukça yüksek tahminler yaparken, EEFO gerçek değere en yakın sonucu vermiştir. Son olarak, 10. elemanda da EEFO'nun tahmini, gerçek değere diğer algoritmalarından daha yakın bir değer olarak kaydedilmiştir.

**Çizelge 4.8.** 10 Elemanlı Kafes (Senaryo 2, Popülasyon 50): Algoritmaların hasar durumları

| Eleman No | Gerçek Hasar Oranı | Ölçülen Hasar Oranı |       |       |
|-----------|--------------------|---------------------|-------|-------|
|           |                    | HS                  | EEFO  | ABC   |
| 7         | 0,150              | 0,072               | 0,023 | 0,100 |
| 8         | 0,200              | 0,339               | 0,349 | 0,374 |
| 9         | 0,200              | 0,290               | 0,297 | 0,696 |
| 10        | 0,250              | 0,287               | 0,172 | 0,020 |

Çizelge 4.8, popülasyon sayısı 50 olarak ayarlandığında, belirtilen elemanlar için algoritmaların ürettiği hasar oranı tahminlerini gerçek değerlerle karşılaştırmaktadır. 7. elemanda tüm algoritmalar gerçek hasar oranını olduğundan düşük tahmin etmiş, ABC ise bu değere en yakın tahmini sunmuştur. 8. elemanda, her üç algoritma da gerçek değer üzerinde ve birbirine yakın tahminlerde bulunmuştur. En belirgin sapma 9. elemanda gözlemlenmektedir; burada gerçek değer 0,2 iken, HS ve EEFO'nun tahminleri 0,3 civarında iken ABC'nin tahmini 0,696 ile oldukça yüksek bir değer sergilemektedir. 10. elemanda ise HS'nin tahmini gerçek değere en yakın sonuç olurken, ABC ise gerçek orandan belirgin şekilde düşük bir tahminde bulunmuştur.

**Çizelge 4.9.** 10 Elemanlı Kafes (Senaryo 2, Popülasyon 75): Algoritmaların hasar durumları

| Eleman No | Gerçek Hasar Oranı | Ölçülen Hasar Oranı |       |       |
|-----------|--------------------|---------------------|-------|-------|
|           |                    | HS                  | EEFO  | ABC   |
| 7         | 0,150              | 0,068               | 0,119 | 0,115 |
| 8         | 0,200              | 0,338               | 0,233 | 0,185 |
| 9         | 0,200              | 0,722               | 0,158 | 0,045 |
| 10        | 0,250              | 0,017               | 0,299 | 0,553 |

Çizelge 4.9, popülasyon sayısı 75 olarak ayarlandığında, belirtilen elemanlar için algoritmaların ürettiği hasar oranı tahminlerini gerçek değerlerle karşılaştırmaktadır. Algoritmaların tahminleri elemana göre belirgin farklılıklar göstermektedir. En büyük sapma 9. elemanda kaydedilmiştir; burada gerçek değer 0,2 iken, HS 0,722 ile oldukça yüksek bir tahminle büyük bir sapma göstermekte, EEFO 0,158 ile gerçek değere en yakın sonucu üretmekte ve ABC ise 0,045 ile değeri belirgin şekilde düşük tahmin etmektedir. Benzer şekilde 10. elemanda da EEFO gerçek değere en yakın tahmini sunarken, ABC değeri olduğundan yüksek, HS ise sıfıra yakın bir seviyede tahmin etmiştir. 8. elemanda ABC'nin tahmini gerçek değere en yakınken, 7. elemanda tüm algoritmalar gerçek değer altında kalmıştır.

**Çizelge 4.10.** 10 Elemanlı Kafes (Senaryo 2, Popülasyon 100): Algoritmaların hasar durumları

| Eleman No | Gerçek Hasar Oranı | Ölçülen Hasar Oranı |       |       |
|-----------|--------------------|---------------------|-------|-------|
|           |                    | HS                  | EEFO  | ABC   |
| 7         | 0,150              | 0,055               | 0,008 | 0,089 |
| 8         | 0,200              | 0,244               | 0,393 | 0,262 |
| 9         | 0,200              | 0,003               | 0,332 | 0,711 |
| 10        | 0,250              | 0,731               | 0,139 | 0,000 |

Çizelge 4.10, popülasyon sayısı 100 olarak ayarlandığında, algoritmaların ürettiği hasar oranı tahminlerini gerçek değerlerle karşılaştırmaktadır. En büyük sapmalar 9. ve 10. elemanlarda gözlemlenmektedir. 10. elemanda, gerçek hasar oranı 0,25 iken HS 0,731 ile oldukça yüksek bir tahmin sunarken, EEFO değeri düşük tahmin etmiş, ABC ise bu elemanda herhangi bir hasar tespit edememiştir. Benzer şekilde 9. elemanda, gerçek değer 0,2 iken ABC'nin tahmini 0,711 ile oldukça yüksek bir sapma gösterirken, HS'nin tahmini sıfıra yakın bir değerde kalmıştır. 8. elemanda tüm algoritmalar gerçek değer üzerinde tahminler sunmuş, en büyük sapmayı 0,393 değeriyle EEFO göstermiştir. 7. elemanda ise tüm algoritmalar gerçek hasar oranını olduğundan düşük tahmin etmiştir.

**Çizelge 4.11.** 72 Elemanlı Kafes (Senaryo 1): Algoritma performans istatistikleri

| Pop. | HS       |            |         | EEFO     |            |        | ABC      |            |         |
|------|----------|------------|---------|----------|------------|--------|----------|------------|---------|
|      | Ortalama | Std. Sapma | En iyi  | Ortalama | Std. Sapma | En iyi | Ortalama | Std. Sapma | En iyi  |
| 20   | 1372,30  | 40,80      | 1297,24 | 136,18   | 95,01      | 16,63  | 1295,53  | 60,40      | 1163,89 |
| 50   | 1352,86  | 66,81      | 1204,74 | 125,32   | 67,08      | 47,05  | 1304,81  | 44,95      | 1227,02 |
| 75   | 1343,64  | 60,02      | 1213,16 | 215,24   | 102,53     | 40,47  | 1338,56  | 35,21      | 1266,50 |
| 100  | 1317,20  | 62,14      | 1219,20 | 220,16   | 105,81     | 63,56  | 1325,97  | 50,44      | 1235,35 |

Çizelge 4.11, 72 elemanlı kafes problemi için farklı popülasyon boyutları altında üç algoritmanın performans istatistiklerini özetlemektedir. Veriler, EEFO algoritmasının ortalama ve en iyi fonksiyon değerlerinin, tüm popülasyon senaryolarında HS ve ABC algoritmalarının sonuçlarından belirgin ölçüde daha düşük olduğunu ortaya koymaktadır. Buna karşın, HS ve ABC algoritmaları genel olarak birbirlerine daha yakın performans sergilemekte olup, ortalama ve en iyi değerleri çoğunlukla 1200 ile 1400 aralığında kaydedilmiştir. Standart sapma değerleri incelendiğinde ise, belirli bir algoritmanın tüm senaryolarda tutarlı olarak en düşük sapmayı gösterdiği gözlemlenmemektedir; EEFO'nun standart sapma değerleri, özellikle popülasyon 75 ve 100'de, diğer iki algoritmanın değerlerinden daha yüksek olarak kaydedilmiştir.

**Çizelge 4.12.** 72 Elemanlı Kafes (Senaryo 1): Algoritmaların hasar durumları

| Pop. | Ölçülen Hasar Oranı |
|------|---------------------|
|------|---------------------|

|     | Gerçek Hasar Oranı | HS    | EEFO  | ABC   |
|-----|--------------------|-------|-------|-------|
| 20  | 0,850              | 0,597 | 0,998 | 0,592 |
| 50  | 0,850              | 0,574 | 0,985 | 0,562 |
| 75  | 0,850              | 0,929 | 0,984 | 0,795 |
| 100 | 0,850              | 0,774 | 0,975 | 0,515 |

Çizelge 4.12, 72 elemanlı kafes problemi için farklı popülasyon boyutlarında, algoritmaların 0,85'lik gerçek hasar oranını tahmin etme performansını göstermektedir. EEFO algoritması incelenen tüm popülasyon senaryolarında gerçek hasar oranını belirgin şekilde üzerinde tahmin etmiştir. Buna karşın, HS ve ABC algoritmaları, Popülasyon 75'teki HS tahmini hariç, tüm durumlarda gerçek değer altında kalan tahminlerde bulunmuştur. Bu tahminler arasında, ABC algoritmasının Popülasyon 75 senaryosunda ürettiği 0,795'lik değer, 0,85 olan gerçek hasar oranına en yakın sonuç olarak kaydedilmiştir.

**Çizelge 4.13.** 72 Elemanlı Kafes (Senaryo 2): Algoritma performans istatistikleri

| Pop. | HS       |            |         | EEFO     |            |        | ABC      |            |         |
|------|----------|------------|---------|----------|------------|--------|----------|------------|---------|
|      | Ortalama | Std. Sapma | En iyi  | Ortalama | Std. Sapma | En iyi | Ortalama | Std. Sapma | En iyi  |
| 20   | 1357,22  | 55,83      | 1238,37 | 119,72   | 76,23      | 35,58  | 1321,07  | 41,38      | 1275,61 |
| 50   | 1353,31  | 67,97      | 1269,13 | 122,32   | 66,04      | 55,78  | 1326,66  | 54,11      | 1206,49 |
| 75   | 1387,18  | 44,56      | 1294,84 | 188,81   | 106,32     | 49,51  | 1304,84  | 44,94      | 1206,59 |
| 100  | 1330,62  | 72,74      | 1196,95 | 161,94   | 72,89      | 67,93  | 1326,21  | 45,35      | 1259,42 |

Çizelge 3.13, 72 elemanlı uzaysal kafes sisteminin daha karmaşık olan çoklu hasar senaryosuna ait 10'ar tekrarlı analizlerin istatistiksel özetini sunmaktadır. EEFO algoritmasının ortalama ve en iyi fonksiyon değerleri açısından diğer iki algoritmadan belirgin ölçüde daha düşük sonuçlar elde ettiğini ortaya koymaktadır. Tüm popülasyon senaryolarında, EEFO'nun ortalama değerleri 200'ün altında kalırken, en iyi değerleri ise 70'in altındadır. Buna karşın, HS ve ABC algoritmaları genel olarak birbirlerine yakın performans sergilemekte olup, hem ortalama hem de en iyi değerleri çoğunlukla 1200'ün üzerinde kaydedilmiştir. Standart sapma değerleri incelendiğinde ise, belirli bir algoritmanın tüm senaryolarda tutarlı olarak en düşük sapmayı gösterdiği gözlemlenmemekte, EEFO'nun standart sapma değerleri bazı durumlarda diğerlerinden yüksek olabilmektedir.

**Çizelge 4.14.** 72 Elemanlı Kafes (Senaryo 2): Algoritmaların hasar durumları

| Popülasyon | Eleman No | Gerçek Hasar Oranı | Ölçülen Hasar Oranı |      |     |
|------------|-----------|--------------------|---------------------|------|-----|
|            |           |                    | HS                  | EEFO | ABC |

|            |    |       |       |       |       |
|------------|----|-------|-------|-------|-------|
| <b>20</b>  | 4  | 0,900 | 0,229 | 0,996 | 0,946 |
|            | 58 | 0,850 | 0,810 | 0,996 | 0,789 |
| <b>50</b>  | 4  | 0,900 | 0,679 | 0,993 | 0,666 |
|            | 58 | 0,850 | 0,239 | 0,977 | 0,541 |
| <b>75</b>  | 4  | 0,900 | 0,388 | 0,799 | 0,554 |
|            | 58 | 0,850 | 0,267 | 0,989 | 0,795 |
| <b>100</b> | 4  | 0,900 | 0,444 | 0,986 | 0,948 |
|            | 58 | 0,850 | 0,587 | 0,985 | 0,068 |

Çizelge 4.14'te sunulan verilere göre, 72 elemanlı kafes sistemindeki 4 ve 58 numaralı hasarlı elemanlar için farklı popülasyon büyüklüklerinde elde edilen tahminler değerlendirildiğinde, EEFO algoritmasının genel olarak gerçek hasar oranlarına en yakın sonuçları ürettiği görülmektedir. Özellikle 58 numaralı elemanda EEFO, tüm senaryolarda 0.977 ile 0.996 arasında değişen tahminlerle istikrarlı bir performans sergilemiştir. 4 numaralı elemanda ise popülasyon 75 için düşük bir değer üretmiş olsa da diğer popülasyon boyutlarında yüksek doğruluk sağlamıştır. Buna karşılık HS ve ABC algoritmalarının tahminleri hem popülasyon sayısına hem de eleman numarasına bağlı olarak değişkenlik göstermiştir. HS algoritması bazı durumlarda (örneğin popülasyon 20'de 4 numaralı eleman için 0.229) oldukça düşük tahminlerde bulunurken, ABC algoritması özellikle popülasyon 100'de 58 numaralı eleman için yalnızca 0.068 değerinde bir tahmin üreterek en büyük sapmayı göstermiştir. Bu bulgular, EEFO'nun genel olarak daha tutarlı sonuçlar sunduğunu ancak hiçbir algoritmanın tüm senaryolarda en doğru tahmini yapamadığını ortaya koymaktadır.

## 5. SONUÇLAR

Bu tez çalışmasında, yapısal hasar tespiti problemine yönelik olarak, doğadan esinlenen yeni bir meta-sezgisel yöntem olan Elektrikli Yılan Balığı Avlanma Optimizasyonu (EEFO) algoritmasının etkinliği incelenmiş ve performansı, literatürde yaygın olarak kullanılan Harmoni Arama (HS) ve Yapay Arı Kolonisi (ABC) algoritmalarıyla karşılaştırmalı olarak değerlendirilmiştir. Analizler, farklı karmaşıklık seviyelerini temsil eden 10 elemanlı düzlemsel ve 72 elemanlı uzaysal kafes sistemleri üzerinde, hem tekli hem de çoklu hasar senaryoları ve farklı popülasyon büyüklükleri altında yürütülmüştür. Çalışmanın temel amacı, EEFO'nun doğruluk, yakınsama hızı, çözüm kararlılığı ve zorlu problemlerdeki uygulama kapasitesini objektif olarak ortaya koymaktır.

Analizler sonucunda elde edilen en temel bulgu, EEFO algoritmasının incelenen tüm problemler ve senaryolar genelinde HS ve ABC algoritmalarına kıyasla belirgin bir üstünlük sergilediğidir. Yakınsama grafiklerinin incelenmesi, bu üstünlüğün temel nedenini ortaya koymaktadır: HS ve ABC algoritmaları, özellikle arama sürecinin erken aşamalarında yerel optimum değerlere takılma eğilimi göstermiştir. Bu durum, bu iki algoritmanın arama uzayının tamamını etkin bir şekilde keşfetmek yerine, buldukları ilk iyi çözümlerin etrafında sıkışıp kalmalarına neden olmuştur. Buna karşın EEFO, keşif ve sömürü aşamaları arasındaki dinamik geçiş mekanizması sayesinde, basamaklı ve kademeli bir düşüş profili sergilemiştir. Bu davranış, EEFO'nun geçici platolarda takılıp kalsa dahi bu durumlardan kurtularak daha iyi çözümler aramaya devam ettiğini ve küresel optimuma daha başarılı bir şekilde yakınsadığını göstermektedir. Bu tespit, tüm popülasyon senaryolarında EEFO'nun istatistiksel olarak daha düşük ortalama ve en iyi fonksiyon değerleri elde etmesiyle de nicel olarak kanıtlanmıştır.

EEFO'nun üstünlüğü, problem karmaşıklığı arttıkça daha da belirginleşmiştir. Nispeten basit olan 10 elemanlı kafes probleminde dahi EEFO, diğer algoritmalar takılı kalırken fonksiyon değerini sıfıra yakın seviyelere indirmeyi başarmıştır. Ancak asıl performans farkı, 72 elemanlı uzaysal kafes gibi daha yüksek boyutlu ve karmaşık bir problemde ortaya çıkmıştır. Bu zorlu problemde, HS ve ABC algoritmaları neredeyse tüm popülasyon senaryolarında 1200'ün üzerindeki yüksek fonksiyon değerlerinde takılı kalırken, EEFO fonksiyon değerini istikrarlı bir şekilde 100'ün altına, hatta sıfıra yakın değerlere kadar düşürmüştür. Bu durum, EEFO'nun ölçeklenebilirliğinin ve daha gerçekçi mühendislik problemlerini çözme kapasitesinin rakiplerinden önemli ölçüde daha yüksek olduğunu göstermektedir. Benzer şekilde, çoklu hasar senaryoları da EEFO'nun avantajını pekiştirmiştir.

Algoritmaların performansının popülasyon büyüklüğüne olan duyarlılığı, bu çalışmanın önemli bulgularından bir diğerini oluşturmaktadır. EEFO'nun göreceli başarısı, özellikle 20 ve 50 gibi düşük popülasyon büyüklüklerinde dikkat çekicidir. Bu kısıtlı koşullarda, HS ve ABC'nin yerel optimum tuzaklarına düşme riski belirgin şekilde

artarken, EEFO'nun daha az sayıda çözüm adayıyla dahi arama uzayını daha verimli bir şekilde tarayabildiği gözlemlenmiştir. Ancak, optimum popülasyon büyüklüğünün seçimi, problemin karmaşıklığına bağlıdır. Daha basit olan 10 elemanlı kafes probleminde, 50 kişilik bir popülasyon, hesaplama maliyeti ve performans arasında optimal bir denge sunarken, popülasyonu 75 veya 100'e çıkarmak sadece marjinal bir fayda sağlamıştır. Buna karşın, 72 elemanlı uzaysal kafes gibi yüksek boyutlu ve karmaşık bir problemde, çözümün kararlılığını ve global optimuma ulaşma güvencesini artırmak için 75 ve 100 gibi daha büyük popülasyonların kullanılması gerekli ve anlamlı hale gelmektedir. Bu zorlu problemde daha fazla çözüm adayına sahip olmak, algoritmanın geçici platolardan daha hızlı kurtulmasını ve arama uzayını daha güvenilir bir şekilde kapsamasını sağlamıştır. Bu değerlendirme, EEFO'nun esnek bir yapıya sahip olduğunu, ancak en iyi performans için problemin boyutuna uygun bir popülasyon büyüklüğünün seçilmesinin kritik olduğunu ortaya koymaktadır.

Grafiksel bulgular da bu sonuçları desteklemektedir. Fonksiyon değerinin iterasyonlara göre değişimini gösteren grafiklerde EEFO'nun daha kısa sürede daha düşük fonksiyon değerlerine ulaştığı, özellikle 75 ve 100 bireyli popülasyonlarda minimuma yakın değerlerde stabil şekilde ilerlediği görülmüştür. Hasar oranı tahmin grafikleri ise EEFO'nun gerçek hasar değerlerine daha yakın tahminler sunduğunu, sapma değerlerinin istatistiksel olarak daha düşük olduğunu göstermektedir. Hasar oranı analizlerinde de EEFO'nun varyansının daha dar olduğu, yani sonuçlarının daha tutarlı olduğu açıkça anlaşılmaktadır.

Çalışmanın önemli bir katkısı da, algoritmaların hem doğruluk hem de kararlılık yönünden performanslarının istatistiksel olarak karşılaştırılmış olmasıdır. Ortalama, standart sapma gibi temel istatistiklerle birlikte her senaryo için çoklu deneyler yapılması, sonuçların güvenilirliğini artırmıştır. Ayrıca hem düzlemsel hem de uzaysal sistemlerde test edilmesi, önerilen yöntemin ölçeklenebilirliğini ve farklı sistem türlerine uygulanabilirliğini göstermesi açısından değerlidir.

Bu çalışmanın ortaya koyduğu umut verici sonuçlar, EEFO algoritmasının potansiyelini daha ileri seviyeye taşımak için çeşitli gelecek araştırma yönlerine işaret etmektedir. Öncelikle, algoritmanın etkinliğinin, bu tezde incelenen kafes sistemlerin ötesinde, kabuk, plak ve rijit diyaframlı yapılar gibi farklı dinamik davranışlar sergileyen daha karmaşık sistemler üzerinde de test edilmesi, yöntemin genellenebilirliğini artıracaktır. Bununla birlikte, gerçek dünya uygulamalarına geçişin önemli bir adımı olarak, algoritmanın gürültüye karşı dayanıklılığı, sensör verilerindeki olası belirsizlikleri simüle etmek amacıyla ölçüm verilerine yapay gürültü eklenerek kapsamlı bir şekilde incelenmelidir. Teorik ve sayısal bulguların ötesinde, laboratuvar ortamında fiziksel bir model üzerinde yapılacak deneysel çalışmalarla pratik doğrulanması yapılmalıdır. Son olarak, metodolojik bir gelişim olarak, EEFO'nun güçlü küresel keşif yeteneği, yerel arama (sömürü) kabiliyeti yüksek başka algoritmalarla birleştirilerek hibrit modellerin geliştirilmesi, potansiyel olarak yakınsama hızını daha da artırabilir ve daha rafine

çözümler sunabilir. sağlamakta hem de afet risklerinin azaltılması konusunda pratik faydalar sunmaktadır.



## 6. KAYNAKLAR

- Alkayem, N., Cao, M., Bayat, M. ve Su, Z. 2018. Structural damage detection using finite element model updating with evolutionary algorithms: A survey. *Neural Computing and Applications*, 389–411.
- Barman, S. K., Mishra, M., Maiti, D. K. ve Maity, D. 2021. Vibration-based damage detection of structures employing Bayesian data fusion coupled with TLBO optimization algorithm. *Structural and Multidisciplinary Optimization*.
- Cha, Y.-J. ve Büyüköztürk, O. 2015. Structural Damage Detection Using Modal Strain Energy and Hybrid Multiobjective Optimization. *Computer-Aided Civil and Infrastructure Engineering*, 347–358.
- Chen, Z. ve L. Y. 2018. A new structural damage detection strategy of hybrid PSO with Monte Carlo. *Measurement*.
- Chen, Z., Zhang, K., Chan, T. H., Li, X. ve Zhao, S. 2022. A Novel Hybrid Whale-Chimp Optimization Algorithm for Structural Damage Detection. *Applied Sciences*.
- Dinh-Cong, D. ve Nguyen-Thoi, T. 2024. Simultaneous identification of structural joint and member damage in semi-rigid frames using expanded FRF data. *Structures*.
- Du, D., Pham, D. ve Nguyen-Thoi, T. 2018. Damage Detection of 2D Frame Structures Using Incomplete Measurements by Improved Finite Element Model Reduction Technique and Teaching-Learning-Based Optimization. *Journal of Advanced Engineering and Computation (JAEC)*, 164–173.
- Elaziz, M. A., Dahou, A., Ewees, A. A., Al-Betar, M. A., Shrahili, M., Rehab, I. A. ve Al-qaness, M. A. 2024. Optimized quantum LSTM using modified electric Eel foraging optimization for real-world intelligence engineering systems. *Ain Shams Engineering Journal*.
- Farhadi, M., Ghiasi, R. ve Torkzadeh, P. 2024. Damage detection of truss structures using meta-heuristic algorithms and optimized group method of data handling surrogate model. *Structures*.
- Ghannadi, P. ve Kourehli, S. S. 2019. Model updating and damage detection in multi-story shear frames using Salp Swarm Algorithm. *Earthquakes and Structures*, 63–73.
- Ghannadi, P. ve Kourehli, S. S. 2019. Structural damage detection based on MAC flexibility and frequency using moth-flame algorithm. *Structural Engineering and Mechanics*, 649–659.

- Ghannadi, P. ve Kourehli, S. S. 2020. Multiverse optimizer for structural damage detection. *Journal of Civil Structural Health Monitoring*, 459-474.
- Ghannadi, P., Kourehli, S. S., Noori, M. ve Altabey, W. A. 2020. Efficiency of grey wolf optimization algorithm for damage detection of skeletal structures via expanded mode shapes. *Advances in Structural Engineering*.
- Ghiasi, R., Fathnejat, H. ve Torkzadeh, P. 2019. A three-stage damage detection method for large-scale space structures using forward substructuring approach and enhanced bat optimization algorithm. *Engineering with Computers*, 857–874.
- Gökdağ, H. ve Yildiz, A. R. 2012. Structural Damage Detection Using Modal Parameters and Particle Swarm Optimization. *Materialpruefung/Materials Testing*.
- Hao, H. ve Xia, Y. 2002. Vibration-based damage detection of structures by genetic algorithm. *Journal of Computing in Civil Engineering*, 222–229.
- Hassani, S., Dackermann, U., Mousavi, M. ve Li, J. 2024. Enhanced damage detection for noisy input signals using improved reptile search algorithm and data analytics techniques. *Computers & Structures*.
- Heidari, A. A. 2019. Harris hawks optimization: Algorithm and applications. *Future Generation Computer Systems*, 849–872.
- Holland, J. H. 1975. *Adaptation in Natural and Artificial Systems*. University of Michigan Press.
- Huang, M., Cheng, X. ve Lei, Y. 2021. Structural damage identification based on substructure method and improved whale optimization algorithm. *Journal of Civil Structural Health Monitoring*, 351–380.
- Kang, F., Li, J.-j. ve Xu, Q. 2012. Damage detection based on improved particle swarm optimization using. *Applied Soft Computing*.
- Kaveh, A. ve Maniat, M. 2014. Damage detection in skeletal structures based on charged system search optimization using incomplete modal data. *International Journal of Civil Engineering*, 291–298.
- Kaveh, A. ve Zolghadr, A. 2015. An improved CSS for damage detection of truss structures using changes in natural frequencies and mode shapes. *Advances in Engineering Software*, 93-100.
- Kaveh, A., Hosseini, S. M. ve Zaerreza, A. 2021. Boundary Strategy for Optimization-based Structural Damage Detection Problem using Metaheuristic Algorithms. *Periodica Polytechnica Civil Engineering*, 150–167.

- Kennedy, J. ve Eberhart, R. 1995. Particle swarm optimization. *Proceedings of ICNN'95 - International Conference on Neural Networks*.
- Khanahmadi, M., Mirzaei, B., Ghodrati Amiri, G., Gholhaki, M. ve Rezaifar, O. 2023. Vibration-based damage localization in 3D sandwich panels using an irregularity detection index (IDI) based on signal processing. *Measurement*.
- Kim, N.-I., Kim, S. ve Lee, J. 2019. Vibration-based damage detection of planar and space trusses using differential evolution algorithm. *Applied Acoustics*, 308-321.
- Li, S. ve Lu, Z. R. 2015. Multi-swarm fruit fly optimization algorithm for structural damage identification. *Structural Engineering and Mechanics*, 409–422.
- Li, Y., Minh, H.-L., Cao, M.-S., Qian, X. ve Wahab, M. A. 2024. An integrated surrogate model-driven and improved termite life cycle optimizer for damage identification in dams. *Mechanical Systems and Signal Processing*.
- Maged, N. A., Hasanien, H. M. ve Alharbi, M. 2024. Electric eel foraging algorithm-based optimal control for low voltage ride through capability improvement of Grid-Connected photovoltaic power plants. *Ain Shams Engineering Journal*.
- Majumdar, A., Nanda, B., Maiti, D. K. ve Maity, D. 2014. Structural damage detection based on modal parameters using continuous ant colony optimization. *Advances in Civil Engineering*.
- Makiabadi, M. H., Maheri, M. R. ve Sarcheshmehpour, M. 2024. Damage detection of span bridge structures under moving loads using the hybrid enhanced SOS-SA algorithm. *Iranian Journal of Science and Technology, Transactions of Civil Engineering*.
- Maurya, P., Tiwari, P. ve Pratap, A. 2024. Electric eel foraging optimization algorithm for distribution network reconfiguration with distributed generation for power system performance enhancement considerations different load models. *Computers and Electrical Engineering*.
- Minh, H.-L., Khatir, S., Abdel Wahab, M. ve Cuong-Le, T. 2021. An Enhancing Particle Swarm Optimization Algorithm (EHVPSO) for damage identification in 3D transmission tower. *Engineering Structures*.
- Mishra, M., Barman, S. K., Maity, D. ve Maiti, D. K. 2019. Ant lion optimisation algorithm for structural damage detection using vibration data. *Journal of Civil Structural Health Monitoring*, 117–136.
- Mohebian, P., Aval, S. B., Noori, M., Lu, N. ve Altabey, W. A. 2022. Visible Particle Series Search Algorithm and Its Application in Structural Damage Identification. *Sensors*.

- Moslem, K. ve Nafaspour, R. 2002. Structural damage detection by genetic algorithms. *AIAA Journal (American Institute of Aeronautics and Astronautics)*.
- Nabavi, S., Gholampour, S. ve Haji, M. S. 2021. Damage detection in frame elements using Grasshopper Optimization Algorithm (GOA) and time-domain responses of the structure. *Evolving Systems*.
- Nabavi, S., Gholampour, S. ve Haji, M. S. 2022. Damage identification in structure elements by grasshopper optimization algorithm using dynamic structure behaviors. *Evolutionary Intelligence*.
- Nhamagea, I. A., Lopez, R. H. ve Miguel, L. F. 2016. An improved hybrid optimization algorithm for vibration based-damage detection. *Advances in Engineering Software*, 47–64.
- Rao, R. V., Savsani, V. J. ve Vakharia, D. P. 2011. Teaching-learning-based optimization: A novel method for constrained mechanical design optimization problems. *Computer-Aided Design*, 303-315.
- Sandesh, S. ve Shankar, K. 2010. Application of a hybrid of particle swarm and genetic algorithm for structural damage detection. *Inverse Problems in Science and Engineering*.
- Saremi, S., Mirjalili, S. ve Lewis, A. 2017. Grasshopper Optimisation Algorithm: Theory and application. *Advances in Engineering Software*.
- Sarjamei, S., Massoudi, M. S. ve Sarafraz, M. E. 2022. Damage Detection of Truss Structures Via Gold Rush Optimization Algorithm. *International Journal of Optimization in Civil Engineering*, 69–89.
- Seyedpoor, S. M. 2012. A two stage method for structural damage detection using a modal strain. *International Journal of Non-Linear Mechanics*.
- Seyedpoor, S. M., Shahbandeh, S. ve Yazdanpanah, O. 2015. An efficient method for structural damage detection using a differential evolution algorithm-based optimisation approach. *Civil Engineering and Environmental Systems*, 230–250.
- Seyedpoor, S., Ahmadi, A. ve Pahnabi, N. 2019. damage detection using time domain responses and an optimization method. *Inverse Problems in Science and Engineering*, 669–688.
- Storn, R. ve Price, K. 1997. Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. *Journal of Global Optimization*, 341–359.

- Su, Y., Liu, L. ve Lei, Y. 2021. Structural Damage Identification Using a Modified Directional Bat Algorithm. *Applied Sciences*.
- Tran, H. V., Nguyen, T. C., Le, T. X., Nguyen, N. T., Ngo, T. V. ve Tran, H. N. 2024. Enhancing Salp Swarm Optimization with Orthogonal Diagonalization Transformation for Damage Detection in Truss Bridge. *KSCE Journal of Civil Engineering*.
- Viswanathan, G., Afanasyev, V., Buldyrev, S., Murphy, E., Prince, P. ve Stanley, H. 1996. Lévy flight search patterns of wandering albatrosses. *Nature*.
- Wang, G. G. 2019. Monarch butterfly optimization. *Neural Computing and Applications*, 1995–2014.
- Wei, Z., Liu, J. ve Lu, Z. 2018. Structural damage detection using improved particle swarm optimization. *Inverse Problems in Science and Engineering*.
- Yang, X.-S. 2010. A New Metaheuristic Bat-Inspired Algorithm. N. Krasnogor, P. Lanzi, D. Pelta, ve G. Nicosia içinde, *Nature Inspired Cooperative Strategies for Optimization* (s. 65-74). Berlin: Springer.
- Zhang, Y., Zhou, K., ve Tang, J. 2024. Harnessing collaborative learning automata to guide multi-objective optimization based inverse analysis for structural damage identification. *Applied Soft Computing*.
- Zhao, W., Wang, L., Zhang, Z., Fan, H., Zhang, J., Mirjalili, S., Khodadadi, N. ve Cao, Q. 2024. Electric eel foraging optimization: A new bio-inspired optimizer for engineering applications . *Expert Systems with Applications*.

## ÖZGEÇMİŞ

Aleyna ZIVALIOĞLU

### ÖĞRENİM BİLGİLERİ

|               |                                                                     |
|---------------|---------------------------------------------------------------------|
| Yüksek Lisans | Akdeniz Üniversitesi                                                |
| 2023-2025     | Fen Bilimleri Enstitüsü, İnşaat Mühendisliği Anabilim Dalı, Antalya |
| Lisans        | Antalya Bilim Üniversitesi                                          |
| 2016-2021     | Mühendislik Fakültesi, İnşaat Mühendisliği Bölümü, Antalya          |