# Kafes Sistemlerin Eşzamanlı Boyut, Şekil ve Topoloji Optimizasyonunda Bir Vekil Model Olarak Yapay Sinir Ağları

Tolga Açıkgöz

**YÜKSEK LİSANS TEZİ**

İnşaat Mühendisliği Anabilim Dalı

TEMMUZ 2021

# Artificial Neural Networks as a Surrogate Model in Simultaneous Size, Shape and Topology Optimization of Trusses

Tolga Açıkgöz

**MASTER OF SCIENCE THESIS**

Civil Engineering Department

JULY 2021

# Kafes Sistemlerin Eşzamanlı Boyut, Şekil ve Topoloji Optimizasyonunda Bir Vekil Model Olarak Yapay Sinir Ağları

Tolga Açıkgöz

Eskişehir Osmangazi Üniversitesi
Fen Bilimleri Enstitüsü
Lisansüstü Yönetmeliği Uyarınca
İnşaat Mühendisliği Anabilim Dalı
Mekanik Bilim Dalında
YÜKSEK LİSANS TEZİ
Olarak Hazırlanmıştır

Danışman: Doç. Dr. Hakan Özbaşaran

TEMMUZ 2021

# ETİK BEYAN

Eskişehir Osmangazi Üniversitesi Fen Bilimleri Enstitüsü tez yazım kılavuzuna göre, Doç. Dr. Hakan Özbaşaran danışmanlığında hazırlamış olduğum "**Kafes Sistemlerin Eşzamanlı Boyut, Şekil ve Topoloji Optimizasyonunda Bir Vekil Model Olarak Yapay Sinir Ağları**" başlıklı tezimin özgün bir çalışma olduğunu; tez çalışmamın tüm aşamalarında bilimsel etik ilke ve kurallara uygun davrandığımı; tezimde verdiğim bilgileri, verileri akademik ve bilimsel etik ilke ve kurallara uygun olarak elde ettiğimi; tez çalışmamda yararlandığım eserlerin tümüne atıf yaptığımı ve kaynak gösterdiğimi ve bilgi, belge ve sonuçları bilimsel etik ilke ve kurallara göre sunduğumu beyan ederim. 02/07/2021

Tolga Açıkgöz

# ÖZET

Metasezgisel yöntemler, basitlikleri ve gradyansız hareket operatörleri nedeniyle optimizasyon problemlerinde optimuma yakın çözümler sağlayan modern yöntemler olarak popüler hale gelmiştir. Öte yandan, optimuma yakın bir çözüm bulmak için çoğunlukla çok fazla sayıda tasarım adayının değerlendirilmesini gerektirmektedirler. Popülasyon tabanlı metasezgisellerin yapısal optimizasyonda önemli ölçüde başarılı olduğu iyi bilinmektedir. Bir yapısal optimizasyon süreci sırasında, her bir aday, dikkate alınan yapısal kısıtlamaları karşılayıp karşılamadığının belirlenebilmesi için bir yapısal analiz yazılımına gönderilir. Sonuç olarak, bu optimizasyon süreci, günümüzün gelişmiş bilgisayarlı hesaplama teknolojilerinde bile problemin boyutuna ve karmaşıklığına bağlı olarak hesaplama açısından pahalı ve zaman alıcıdır. Yapay Sinir Ağı tabanlı vekil model, karmaşık problemlerin doğrusal olmayan yapılarını yakalayabilir ve çok fazla hesaplama çabası olmadan son derece zor problemlerde bile hassas yapısal analiz tahminleri sağlayabilmektedir. Bu tez çalışmasında, yapısal optimizasyon süreçlerinin hesaplama maliyetlerini azaltmak için Yapay Sinir Ağı tabanlı vekil destekli bir optimizasyon prosedürü sunulmaktadır. Sayısal deneyler, popüler üç kafes sistem problemi üzerinde eş zamanlı boyut, şekil ve topoloji optimizasyonu üzerinden gerçekleştirilmiştir. Önerilen vekil model destekli yapısal optimizasyon prosedürü, olağan metasezgisel optimizasyon süreci ile elde edilen sonuçlarla karşılaştırılmış ve sunulan yöntemin performansı tartışılmıştır.

**Anahtar Kelimeler:** Yapısal Optimizasyon, Yapay Sinir Ağları, Vekil Model, Metasezgiseller, Kafes Sistemler

# SUMMARY

The metaheuristics have become popular as modern methods that provide near-optimum solutions to the optimization problems because of their simplicity and gradient-free movement operators. On the other hand, they mostly require evaluation of a significant number of design candidates to locate a near-optimal solution. It is well known that population-based metaheuristics are significantly successful in structural optimization. During a structural optimization process, each candidate is sent to a structural analysis software to determine whether it satisfies the considered structural constraints. As a result, these optimization process is computationally expensive and time consuming depending on the size and complexity of the problem even in today's advanced soft computing technologies. The artificial neural network based surrogate assistance is able to capture nonlinear structures of complex problems and provide sensitive estimates even in extremely difficult problems without too much computational effort. This study presents an optimization procedure with an artificial neural network-based surrogate assistance to reduce the computational cost of the structural optimization processes. Numerical experiments were performed on the popular three truss problems by simultaneous size, shape and topology optimization. The proposed surrogate model-assistant structural optimization procedure is compared with the results obtained by the pure metaheuristic optimization process and the performance of the presented method is discussed.

**Keywords:** Structural Optimization, Artificial Neural Networks, Surrogate, Metaheuristics, Trusses

# İÇİNDEKİLER

Sayfa

**ÖZET** . . . . . **vi**

**SUMMARY** . . . . . **vii**

**TEŞEKKÜR** . . . . . **viii**

**İÇİNDEKİLER** . . . . . **ix**

**ŞEKİLLER DİZİNİ** . . . . . **xi**

**ÇİZELGELER DİZİNİ** . . . . . **xiii**

**1. GİRİŞ VE AMAÇ** . . . . . **1**

**2. LİTERATÜR ARAŞTIRMASI** . . . . . **6**

**3. KAFES SİSTEMLER** . . . . . **20**

3.1. Optimum Tasarım Yaklaşımı . . . . . 20

3.2. Metasezgisel Algoritmalar ile Optimizasyon Süreçleri ve Karşılaşılan Problemler . . . . . 23

**4. YSA DESTEKLİ VEKİL MODEL KAVRAMI** . . . . . **27**

4.1. Vekil Model . . . . . 27

4.2. Yapay Sinir Ağı Örneği . . . . . 28

4.2.1. Örnek Eğitim Veri Setinin Hazırlanması . . . . . 30

4.2.2. Örnek Yapay Sinir Ağı Tasarımı . . . . . 31

4.2.3. Veri Ön İşleme Örneği . . . . . 33

4.2.4. Örnek Problem Yapay Sinir Ağı Eğitimi . . . . . 34

4.2.5. Geri Yayılım Algoritması ile Hata Dağılımı . . . . . 37

4.2.6. Geri Yayılım Algoritması ile Hata Optimizasyonu . . . . . 38

4.2.6.1. Gradyan İniş Kavramı . . . . . 40

4.2.6.2. Ara Katman ile Çıktı Katman Ağırlık Güncellemesi . . . 43

4.2.6.3. Girdi Katmanı ile Ara Katman Ağırlık Güncellemesi . . . 44

**5. ÖNERİLEN VEKİL MODEL İLE OPTİMİZASYON YÖNTEMİ** . . . . . **46**

5.1. JAYA Algoritması ile Eş Zamanlı Boyut, Şekil ve Topoloji Optimizasyonu . 46

5.1.1. Başlangıç Popülasyonun Belirlenmesi . . . . . 48

5.1.2. Sonlu Elemanlar Analizi ile Kısıt İhlallerinin Hesaplanması . . . . 49
5.1.3. JAYA ile Yeni Adayların Belirlenmesi . . . . . 51
5.1.4. Popülasyondaki Adayların Karşılaştırılması . . . . . 51
5.1.5. Optimizasyon Sürecinin Sonlandırılması . . . . . 52
5.2. Vekil Model Olarak Derin Öğrenme Stratejisi . . . . . 52
5.2.1. FAZ:1 Parametrelerin Belirlenmesi . . . . . 52
5.2.2. FAZ:1 SE Çözücüsü ile Kısıt İhlallerinin Hesaplanması ve YSA Eğitimi için Veri Toplanması . . . . . 54
5.2.3. FAZ:2'ye Hazırlık ve Ana Popülasyon Oluşturulması . . . . . 56
5.2.4. FAZ:2 Vekil Olarak Seçilen Yapay Sinir Ağı Modeli ve Eğitimi . . 56
5.2.5. FAZ:2 Kısıt İhlallerinin Vekil YSA ile Tahmini . . . . . 58
5.2.6. FAZ:2 Vekil Model Tahminlerinin SE ile Değerlendirilmesi Yaklaşımı 59
5.2.7. FAZ:2 Vekil Model ile Optimizasyon Sürecinin Sonlandırılması . . 59

**6. BULGULAR VE TARTIŞMA . . . . . 60**
6.1. Vekil Model ile Eşzamanlı Boyut, Şekil ve Topoloji Optimizasyonu Deneyleri 60
6.1.1. Test Problemleri ve Vekil Modelsiz Sonuçlar . . . . . 60
6.1.2. YSA Eğitim Deneyleri . . . . . 60
6.1.3. Vekil Model Performans Sonuçları . . . . . 62
6.2. 10 Çubuklu Kafes Sistem Deneyleri . . . . . 64
6.2.1. 10 Çubuklu Kafes Sistem Problemi . . . . . 64
6.2.1.1. 10 Çubuk Test Problemi İçin Vekil Modelsiz Sonuçlar . . 65
6.2.1.2. 10 Çubuk Test Problemi İçin YSA Eğitim Performansı . 68
6.2.1.3. 10 Çubuk Test Problemi İçin Vekil Model Performansı . . 69
6.3. 15 Çubuklu Kafes Sistem Deneyleri . . . . . 73
6.3.1. 15 Çubuklu Kafes Sistem Problemi . . . . . 73
6.3.1.1. 15 Çubuk Test Problemi İçin Vekil Modelsiz Sonuçlar . . 75
6.3.1.2. 15 Çubuk Test Problemi İçin YSA Eğitim Performansı . 77
6.3.1.3. 15 Çubuk Test Problemi İçin Vekil Model Performansı . . 78
6.4. 39 Çubuklu Kafes Sistem Deneyleri . . . . . 82
6.4.1. 39 Çubuklu Kafes Sistem Problemi . . . . . 82
6.4.1.1. 39 Çubuk Test Problemi İçin Vekil Modelsiz Sonuçlar . . 83
6.4.1.2. 39 Çubuk Test Problemi İçin YSA Eğitim Performansı . 86
6.4.1.3. 39 Çubuk Test Problemi İçin Vekil Model Performansı . . 86

**7. SONUÇ VE ÖNERİLER . . . . . 92**

**KAYNAKLAR DİZİNİ . . . . . 94**

# ŞEKİLLER DİZİNİ

**Şekil** **Sayfa**


3.1 Popülasyon Tabanlı Metasezgisellerin Genel Akış Şeması (Chong vd., 2021) . 24
3.2 On Çubuklu Kafes Sistem . . . . . 25

4.1 Çok Katmanlı YSA Örneği . . . . . 31
4.2 YSA Nöron Detayı . . . . . 32
4.3 Sigmoid Aktivasyon Fonksiyonu Grafiği . . . . . 33
4.4 Birinci Aşama YSA modeli . . . . . 35
4.5 Birinci Aşama YSA Giriş Katmanı . . . . . 35
4.6 Birinci Aşama YSA Girdi-Ara Katman Detayı . . . . . 36
4.7 Birinci Aşama YSA Girdi-Ara Katman Detayı . . . . . 37
4.8 YSA Geri Yayılım Hata Şeması . . . . . 39
4.9 Gradyan İniş Örneği . . . . . 40
4.10 Gradyan İniş Detay . . . . . 41

5.1 JAYA Optimizasyon Akış Şeması . . . . . 49
5.2 Sunulan Vekil Model Akış Şeması . . . . . 53
5.3 Sunulan Vekil Model YSA Mimarisi . . . . . 57

6.1 On Çubuklu Kafes Sistem . . . . . 65
6.2 On Çubuklu Kafes Sistem: İzin Verilen Analiz Sayısı - Vekil Modelsiz Optimum Sonuçlar (a) . . . . . 66
6.3 On Çubuklu Kafes Sistem: On bin Analiz Sonucu Optimum Şekil ve Topoloji 67
6.4 On Çubuklu Kafes Sistem: İzin Verilen Analiz Sayısı - Vekil Modelsiz Optimum Sonuçlar (b) . . . . . 67
6.5 On Çubuklu Kafes Sistem: İzin Verilen Analiz Sayısı - Toplanan Veri Yüzdesi . 69
6.6 On Çubuklu Kafes Sistem: İzin Verilen Analiz Sayısı - İterasyon Sayısı . . . . 70
6.7 On Çubuklu Kafes Sistem: İzin Verilen Analiz Sayısı - Eğitim OKH . . . . . 70
6.8 On Çubuklu Kafes Sistem: İzin Verilen Analiz Sayısı - Test OKH . . . . . 71
6.9 On Çubuklu Kafes Sistem: İzin Verilen Analiz Sayısı - Test / Eğitim OKH . . . 71
6.10 On Çubuklu Kafes Sistem: Vekil Model Performans Sonuçları . . . . . 72
6.11 On beş Çubuklu Kafes Sistem . . . . . 74
6.12 On Beş Çubuklu Kafes Sistem: İzin Verilen Analiz Sayısı - Vekil Modelsiz Optimum Sonuçlar (a) . . . . . 75
6.13 On beş Çubuklu Kafes Sistem: Bulunan Optimum Şekil ve Topoloji . . . . . 76

# ŞEKİLLER DİZİNİ (devam)

**Şekil** **Sayfa**

6.14 On Beş Çubuklu Kafes Sistem: İzin Verilen Analiz Sayısı - Vekil Modelsiz Optimum Sonuçlar (b) . . . . . 77
6.15 On beş Çubuklu Kafes Sistem: İzin Verilen Analiz Sayısı - Toplanan Veri Yüzdesi 78
6.16 On beş Çubuklu Kafes Sistem: İzin Verilen Analiz Sayısı - İterasyon Sayısı . . 79
6.17 On beş Çubuklu Kafes Sistem: İzin Verilen Analiz Sayısı - Eğitim OKH . . . . 79
6.18 On beş Çubuklu Kafes Sistem: İzin Verilen Analiz Sayısı - Test OKH . . . . . 80
6.19 On beş Çubuklu Kafes Sistem: İzin Verilen Analiz Sayısı - Test / Eğitim OKH . 80
6.20 On beş Çubuklu Kafes Sistem: Vekil Model Performans Sonuçları . . . . . 81
6.21 Otuz dokuz Çubuklu Kafes Sistem . . . . . 83
6.22 Otuz Dokuz Çubuklu Kafes Sistem: İzin Verilen Analiz Sayısı - Vekil Modelsiz Optimum Sonuçlar (a) . . . . . 84
6.23 Otuz dokuz Çubuklu Kafes Sistem: Bulunan Optimum Şekil ve Topoloji . . . 85
6.24 Otuz Dokuz Çubuklu Kafes Sistem: İzin Verilen Analiz Sayısı - Vekil Modelsiz Optimum Sonuçlar (b) . . . . . 85
6.25 Otuz dokuz Çubuklu Kafes Sistem: İzin Verilen Analiz Sayısı - Toplanan Veri Yüzdesi . . . . . 87
6.26 Otuz dokuz Çubuklu Kafes Sistem: İzin Verilen Analiz Sayısı - İterasyon Sayısı 88
6.27 Otuz dokuz Çubuklu Kafes Sistem: İzin Verilen Analiz Sayısı - Eğitim OKH . 88
6.28 Otuz dokuz Çubuklu Kafes Sistem: İzin Verilen Analiz Sayısı - Test OKH . . . 89
6.29 Otuz dokuz Çubuklu Kafes Sistem: İzin Verilen Analiz Sayısı - Test / Eğitim OKH 89
6.30 Otuz dokuz Çubuklu Kafes Sistem: Vekil Model Performans Sonuçları . . . . . 90

# ÇİZELGELER DİZİNİ

**Çizelge** **Sayfa**

3.1 On Çubuklu Kafes Metasezgisel Optimizasyon Sonuçları (Sonmez, 2018) . . . 26

4.1 Örnek Eğitim Veri Seti . . . 30
4.2 Ölçeklenmiş Eğitim Veri Seti . . . 34
4.3 Örnek Problem YSA Hataları . . . 39

5.1 Vekil Model için Kullanıcı Tarafından Tanımlanacak Parametreler . . . 54

6.1 On Çubuklu Kafes:Vekil YSA Aktivasyon Fonksiyonları (Özbaşaran, 2021) . . 62
6.2 On Çubuklu Kafes Sistem Tasarım Parametreleri . . . 64
6.3 On Çubuklu Kafes Boyut, Şekil ve Topoloji Optimizasyonu için Referans Sonuçlar 66
6.4 On beş Çubuklu Kafes Sistem Tasarım Parametreleri . . . 74
6.5 On beş Çubuklu Kafes Boyut, Şekil ve Topoloji Optimizasyonu için Referans Sonuçlar . . . 75
6.6 Otuz dokuz Çubuklu Kafes Sistem Tasarım Parametreleri . . . 83
6.7 Otuz Çubuklu Kafes Boyut, Şekil ve Topoloji Optimizasyonu için Referans Sonuçlar . . . 84

# 1. GİRİŞ VE AMAÇ

Genel olarak mühendislik uygulamalarında tasarımların mümkün olduğunca yüksek performanslı ve maliyetlerin olabildiğince az olması amaçlanmaktadır. Günümüz mühendislik tasarımlarının genelinde olduğu gibi İnşaat Mühendisliği'nin alt disiplinlerinden biri olan Yapısal Mühendislik tasarımlarında da nihai amaç, mümkün olan optimum tasarıma ulaşmaya çalışmaktır. Yapı Mühendisliği, hesaplama yöntemleri ve tasarım yaklaşımları göz önüne alındığında çok fazla deneme gerektiren ve zaman alan karmaşık bir süreci kapsamakta, problemlerin çözümünde mühendis tarafından benimsenen geleneksel yaklaşımlar, ancak birkaç sınırlı deneme sonucunda sunulan, çözümlerin en iyi olduğu varsayımına dayanmaktadır. Tasarlanacak yapı karmaşıklaştıkça gelenekçi deneme-yanılma yöntemleri, mühendisi ekonomik çözümlerden beklenildiği üzere bir hayli uzaklaştırmaktadır. Diğer mühendislik disiplinlerinde de olduğu gibi Yapı Mühendisliği, geleneksel hesaplama tekniklerinin yetersiz kaldığı karmaşık problemlerle doludur. Gerçek dünyada yapı tasarım problemleri, geniş arama uzayları, birçok tasarım değişkenleri içermesi, doğrusal olmayan amaç ve kısıt fonksiyonlarından kaynaklı olarak, problemlerin optimum tasarımlarının bulunması zor ve karmaşık bir hale gelmektedir. Optimuma yakın sayılabilecek, uygun çözümler, çok fazla deneme-yanılma gerektiren ve çoğu zaman analitik olarak bir model oluşturulabilse bile, tekrarlı ve karmaşık matematiksel modellerin analitik olarak hesaplanmasını gerektirmektedir. Hesaplamalı yöntemler, fiziksel olayları tanımlayan matematiksel modelleri sayısal olarak çözmek için kullanılan bilgisayar tabanlı yöntemlerdir. Günümüzde çoğunlukla, yapıların tasarım problemlerinin çözümünde (genellikle sezgisel analitik çözümlerin mevcut olmadığı durumlarda) farklı koşullar altında karmaşık yapısal sistemlerin davranış tahminlerini yapmak için hesaplamalı yöntemlerden biri olan Sonlu Elemanlar Analizi kullanılmaktadır.

Yapısal problemlerin optimal çözümlerinin aranması adına sıklıkla optimallik ilkesi ve matematiksel programlama teknikleri kullanılmıştır. Optimallik ilkesi, optimal tasarımın önceden belirlenen koşulların karşılanması gerektiğini varsayar. Optimallik ilkesinin aksine, bilgisayarlı hesaplama teknolojisi ve paralel programlamadaki gelişmeler göz önüne alındığında, matematiksel programlama teknikleri, bu problemlerin çözümünde daha etkili araçlar olarak ön plana çıkmaktadır. Genel olarak literatür incelendiğinde her iki yöntem de daha çok basit tasarım senaryoları üzerinde denenmiştir. Çoklu yük dağılımları, tasarım spesifikasyonları ve birçok kısıt göz önüne alındığında yapısal tasarım problemleri karmaşıklaştıkça bu yöntemler optimum tasarımların ortaya çıkabilmesi için yetersiz kalmaktadır (Ahrari vd., 2015). Gerçek dünya yapısal mühendislik problemlerinde çok

sayıda tasarım değişkeni bulunmakta ve doğrusal olmayan amaç fonksiyonları ve kısıtların yanı sıra yüksek boyutluluk, multimodalite ve dışbükey olmayan bölgeler problemlerin karmaşıklığını arttırmaktadır (Charalampakis ve Tsiatas, 2019). Emniyetten taviz verilmeden, ekonomik yapı tasarımlarının ortaya çıkabilmesi "Yapısal Optimizasyon" disiplininin konusu olmakla birlikte, bu disiplin son yıllarda genel olarak, gelişmiş bilgisayarlı hesaplama yöntemlerinin kullanıldığı, yapısal optimizasyon algoritmaları ve yeni optimizasyon tekniklerinin geliştirilmesi ile ilgilenir. Tasarım ve hesaplama yöntemlerinin karmaşıklığı göz önüne alındığında, hata yapma olasılığının ve hesaplama maliyetlerinin azaltılması, ekonomik tasarımların ortaya çıkarılabilmesinin sağlanması için bilgisayar destekli hesaplama yöntemlerinin gelişimine olan gereksinimin giderek arttığı bir sürece girdiğimiz açıktır. Bu süreç, bilgisayar teknolojisinde yapılan büyük atılımların sonucunda, bilgisayarlı hesaplama yöntemlerinin gelişmesi ile, Yapı Mühendisliği alanının çok disiplinli karakterini araştıran ve tasarım süreçlerini destekleyen yeni metotları içermektedir.

Kafes sistemler, sürekli yükler altında düğümlere bağlanan, basma veya çekme gerilmelerine maruz kalan doğrusal elemanlardan oluşan iki veya üç boyutlu yapılardır. Kafes sistemler köprüler, uzun açıklıklı çatılar ve iletim kuleleri gibi daha birçok yapıda yaygın olarak kullanılan ve mekanik özellikleri, imalat kolaylıkları gibi artılarından dolayı yapı mühendisliğinde yaygın olarak tercih edilmektedirler. Geleneksel olarak kafes yapı optimizasyonu, belirli kısıtların sağlanması şartı ile, ağırlık, uygunluk ve yer değiştirme gibi amaç fonksiyon değerlerinin optimuma yakınsaması için uygun yapısal tasarım değişkenlerinin aranması (arama uzayı) sürecini kapsamaktadır. Literatürdeki kafes yapı optimizasyonu için ortak kısıtlar, kinematik stabilite, gerilme, yer değiştirme ve burkulma yüküdür, ancak frekans kısıtlamalarını da dikkate alan bazı çalışmalar yapılmıştır (Techasen vd., 2018). Bir kafes sistem yapısında kullanılan malzeme miktarının, yapının ağırlığı ve maliyeti ile doğrudan ilişkili olduğu varsayımından yola çıkarak, kafes sistemin tasarım verimliliği için kafes yapı ağırlığının minimize edilmesi yaklaşımı kullanılabilir (Camp ve Bichon, 2004).

Tüm bu kafes yapı tasarım problemlerinin kendileriyle ilişkili belirli bir karmaşıklığı vardır. Bu, mevcut değişkenlerin sayısına (arama uzayı) ve amaç fonksiyonunun karmaşıklığı ve kısıtlar nedeniyle önerilen bir çözümün uygun olmama olasılığına bağlı olabilir. Yapısal optimizasyonun ilgili alanlarından biri olan kafes yapıların, farklı tasarım durumları için 1960 yılından bu yana birçok optimizasyon yöntemi geliştirilmiş ve yaygın olarak kullanılmıştır (Sonmez, 2018). Kafes yapılar, tasarım değişkenlerine göre boyut, şekil ve topoloji optimizasyonu olarak ele alınabilir. Kısaca topoloji optimizasyonu gereksiz doğrusal elemanları ve düğüm noktalarının sistemden kaldırılması veya gerekli olanların muhafaza edilmesi, boyut optimizasyonu elemanların

kesit alanları yani boyutlandırılması, şekil optimizasyonu ise doğrusal elemanların bağlandığı düğüm noktalarının uygun topolojiyi oluşturacak biçimde belirli koordinatlarda olması ile ilgilenir. Özellikle topoloji optimizasyonu hesaplama açısından çok maliyetli bir süreç olarak değerlendirilir çünkü topolojideki küçük bir varyasyon bile eleman kuvvetlerinde önemli değişikliklere neden olabilir. Ayrıca arama sırasında kinematik olarak kararsız birçok yapı üretilebilir. Kinematik olarak stabil olmayan kararsız ve geçersiz yapılar, topoloji optimizasyonun önündeki en temel engellerdir ve bu nedenle çok sayıda istenmeyen sonlu elemanlar analizinden kaçınmak için topolojinin iyi bir şekilde tanımlanması ve verimli bir şekilde optimizasyon sürecinde kullanılması gerekmektedir (Savsani vd., 2016). Literatürde geleneksel olarak kafes sistemlerin optimizasyonları, çeşitli teknikler veya yaklaşımlarla belirli gruplamalar veya sıralarla yapılmıştır. Yine de bir tasarımcının eş zamanlı olarak üç tasarım değişkeni türünü (topoloji, şekil ve boyut) optimize edebilmesinin daha etkili olduğu bulunmuştur (Techasen vd., 2018). Eşzamanlı topoloji, şekil ve boyut optimizasyonu, tek aşamalı (eş zamanlı) bir süreç olduğundan daha optimuma yakın sonuçlar vermesine rağmen, çok aşamalı optimizasyon yaklaşımlarından daha fazla hesaplama çabası gerektirir.

Kafes yapıların tasarım problemlerinin optimum sonuçlarına ulaşabilmek için gradyan tabanlı geleneksel optimizasyon yöntemleri yetersiz kalmakta, çoğu zaman lokal bölgelerde takılarak optimum değerlerden uzak sonuçlar vermektedir (Pholdee ve Bureerat, 2014). Son yıllarda önerilen popülasyon tabanlı metasezgisel algoritmalar global çözümlere yakınsamada, gradyan tabanlı optimizasyon metotlarına kıyasla, çok daha başarılıdır (Souza vd., 2016). Basit olasılık ve istatistiksel yöntemlerle zenginleştirilmiş evrim ilkeleri, sürü zekâsı veya diğer fiziksel fenomenlerden esinlenilen metasezgisel algoritmaların, gerçek dünya mühendislik problemlerinin optimal çözümünü aramada modelleme kolaylığı sağladığı ve iyi sonuçlar verdiği bilinmektedir (Charalampakis ve Tsiatas, 2019). Çeşitli karmaşık optimizasyon problemlerinin çözümünde metasezgiseller, gradyan tabanlı yöntemlerden farklı olarak, deterministik olmaması, türev ve gradyan bilgisi gerektirmemesi gibi özelliklerinden dolayı popüler hale gelmiştir (Degertekin vd., 2019). Metasezgisel algoritmalar, her durumda farklı bir nihai çözüme ulaşılmasını sağlayan olasılıksal geçiş ilkelerini kullanmakta olup bu durum çok güçlü ve sağlam arama mekanizmalarına yol açmaktadır (Sonmez, 2018).

Metasezgisel algoritmaların en uygun tasarım çözümleri vasıtası ile aranması popülasyon olarak adlandırılır. Optimizasyon süreci, belirli bir adayı veya adayları içeren ilk popülasyonun rastgele olarak atanmasıyla başlar. Kafes sistemlerin, metasezgisel optimizasyon algoritmaları ile optimum çözümlere ulaşılması sürecinde, istenilen kısıtların belirlenebilmesi için popülasyondaki aday çözümler, statik analizlerin yapılacağı bir sonlu elemanlar çözücüsüne gönderilmektedir. Sonrasında süreç tasarım çözümlerinin rastgele bir

sıra ile güncellenmesini ve iyileştirilmesini içermektedir. Belirli bir iterasyon sayısına ulaşıldığında veya artık herhangi bir gelişme gözlenmediğinde optimizasyon süreci durur.

Popülasyonun iyileştirilmesi sürecinde, her bir iterasyonda adayların kısıtları sağlayıp sağlanmadığının kontrolü amacıyla defalarca Sonlu Elemanlar çözücüsü çağırıldığından dolayı, yapılan analiz sayıları ile birlikte hesaplama maliyetlerinin büyük ölçüde arttığı görülmektedir. Optimizasyon rutini sırasında gerekli olan amaç fonksiyon değerlendirmelerinin miktarına karşılık gelen sonlu eleman analizlerinin sayısı hesaplama süresini etkilediği için özellikle hesaplama maliyetlerini yüksek oranda arttıran bir faktördür (Müller ve Klashorst, 2017). Yapısal bir problemin karmaşıklığını etkileyen bir diğer faktör, farklı değişkenlerin karışımıdır (Ahrari vd., 2015). Bunlar, ayrık, sürekli ve mantıksal değişkenleri içerir. Kafes yapıların tasarım durumları ve problemin türüne göre, geniş arama uzayları ve çok sayıda tasarım değişkenleri de barındırmaları sebebiyle hesaplama maliyetleri bir hayli yüksektir (Talatahari vd., 2020). Son yıllarda gelişen bilgisayar teknolojilerine paralel olarak, bilgisayarlı hesaplama metotlarının hız ve performansında büyük gelişim sağlanmasına rağmen, bu kadar çok adayın değerlendirildiği bu süreç, optimize edilen yapıya bağlı olarak aylar sürebilmektedir.

Hesaplama maliyetlerinin bu denli fazla olması optimal tasarımların ortaya konmasında büyük bir engel teşkil etmektedir. Tasarım süreçlerinde ekonomik çözümlere ulaşmak bir hayli zaman almakta, ekonomik çözümlerden uzaklaşılmaktadır. Bu hesaplama yüklerinin azaltılması adına deterministik sonlu elemanlar hesaplama modelini simüle eden, model girdilerine bağlı olarak verilen çıktı değerlerine tutarlı tahminsel cevaplar üreten, vekil model olarak bilinen bir kestirim modeli sunulmuştur. Önerilen vekil model çok sayıda iterasyondan oluşan, hesaplama maliyetleri fazla olan optimizasyon süreçlerindeki kısıtların belirlenmesi için popülasyondaki her bir adayın gönderildiği, sonlu elemanlar analiz aracının verdiği çıktıları kabul edilebilir bir şekilde tahmin etmektedir. Optimal çözümlere yakınsamak için çok fazla adayın bu sürece girmesi ile birlikte sürecin gerçekleşeceği bilgisayarlar için çok fazla bellek ve işlemci gücü gerektirdiğinden dolayı bu analiz sayılarını azaltacak bir vekil model tekniği önerilmiştir.

Vekil olarak, Yapay Sinir Ağları ile oluşturulmuş doğrusal olmayan bir regresyon modeli tasarlanmıştır. Yapay Sinir Ağları, insan beyninde öğrenmeyi sağlayan biyolojik sinir sisteminin çalışma prensibinin simüle edildiği bir modeldir (Ballal ve Sher, 2003). Yapay Zekâ'nın alt dallarından olan Makine Öğrenmesi yöntemlerinden biri olan Yapay Sinir Ağları ile oluşturulmuş Regresyon tahmin modellerinin, Yapı Mühendisliği'ndeki karmaşık problemlerin, doğrusal olmayan yapılarını yakalayabildiği ve son derece güç tahminleri bile hassas bir şekilde sağlayabildiği kanıtlanmıştır (Chou vd., 2016). Yapay Sinir Ağları tabanlı modellerin herhangi bir mühendislik problemini çözmedeki başarısında,

eğitim için sunulan veri örnekleriyle öğrenmesi ve girdi ve çıktılar arasındaki ilişkiler bilinmediğinde (Kara Kutu) veya fiziksel anlamın açıklanmasının zor olduğu durumlarda bile veriler arasındaki ince fonksiyonel ilişkileri yakalayarak, çıktı olarak kabul edilebilir yaklaşık tahminsel değerler üretmesi yatmaktadır (Prayogo vd., 2020).

Yapay Sinir Ağlarının vekil olarak kullanıldığı bu model sayesinde popülasyon tabanlı Metasezgisel algoritmaların hesaplama maliyetlerinin düşürülerek, optimizasyon süreçlerinin kısaltılması ve ilgili alanlarda kullanılmasıyla birlikte daha karmaşık problemlerin daha az zaman ve düşük maliyetlerle çözülebilmesi amaçlanmıştır. Önerilen vekil model ile, Sonlu Elemanlar analizi gibi hesaplama metodojilerinin simüle edildiği bu ve bunun gibi tüm alanlardaki optimizasyon problemleri için kullanılabilir olması ve metasezgisellerin hesaplama maliyetlerinin düşürülmesi ile hem zamandan büyük oranda tasarruf elde edilmesi, hem de ülkemize ekonomik ve teknolojik anlamda katkı sağlanması amaçlanmaktadır.

Hesaplama çabasının azaltılması ile mevcut işlemci ve hesaplama gücü ortalama olan kişisel bilgisayarlarda bile bu süreçlerin makul sürelere düşürülmesi amaçlanmaktadır. Bu sayede optimizasyon alanındaki çalışmaların ilgili herkesin yapabileceği bir sürece dönüştürülerek, ekonomik ve güvenilir çözümler geliştirilebilmesinin önünün açılabilmesi sağlanacaktır. Yapı mühendisliği için geliştirilen bilgisayar yazılımlarında kayda değer bir optimizasyon modülüne rastlanmamaktadır. Sunulan vekil modelin istenilen hedeflere ulaşması halinde vekil model ile Yapı mühendisliği için hazırlanan bilgisayar destekli tasarım yazılımlarına optimizasyon modülü eklenebileceği fikri oluşmaktadır. Böylece, ülkemizde optimal tasarımların yaygınlaşacağı ve ülke ekonomisine önemli bir katkıda bulunulacağı düşünülmektedir. Ucuz ve emniyetli konut, işyeri vb. yapıların inşa edilebilmesini sağlayarak ülkemiz ve dünya ekonomisine gözle görülür katkılar sağlanabilecektir.

İlgili tez çalışmasında verim ve performans karşılaştırılmalarının yapılabilmesi adına çeşitli kafes sistem problemleri üzerinde, önerilen Yapay Sinir Ağı tabanlı vekil model ile eş zamanlı boyut, şekil ve topoloji optimizasyonu yapılmış ve deney sonuçları karşılaştırılarak, tartışılmıştır. Deney sonuçlarının karmaşık optimizasyon problemleri için uygulanabilirliği irdelenmiş ve optimizasyon hesaplama maliyetlerinin ve sürelerinin azaltılması adına nispeten başarılı sayılabilecek sonuçlara ulaşılmıştır.

# 2. LİTERATÜR ARAŞTIRMASI

Papadrakakis vd. (1998) 'Evrim Stratejilerini ve Sinir Ağlarını Kullanarak Yapısal Optimizasyon' başlıklı çalışmalarında, seçilen test problemleri üzerinde Metasezgisel olarak Evrimsel Strateji algoritmasını uygulayarak tekrarlı optimizasyon prosedürlerinin hesaplama yüklerini azaltmak için Sonlu Elemanlar Analizleri'nin tahminlerini yapan Yapay Sinir Ağı tabanlı bir Regresyon modeli sunar. Boyut ve şekil optimizasyonunu ayrı olarak ele alır. Şekil optimizasyonunu, merkezi kesme sorunu olan kare plak problemi üzerinde devamlı değişkenlerle, boyut optimizasyonunu ise altı ve yirmi altı katlı uzay çerçeve problemleri üzerinde karşılaştırmalı olarak incelemiştir. Yapay Sinir Ağı modelinin eğitimi ve testi için veri setinin arama uzayından nasıl seçilmesi gerektiğine dair verimli bir yöntem de önerir. Problemlerdeki kısıtların belirlenmesi amacı ile Sonlu Elemanlar analizinin tahminlerini yapacak Yapay Sinir Ağı model ve parametrelerinin problem türüne göre spesifik olarak seçildiği görülür. Papadrakakis ve Lagaros (2003) 'Yapısal optimizasyon için yumuşak hesaplama metodolojileri' başlıklı çalışmaları Papadrakakis vd. (1998)'deki çalışmanın devamı şeklinde görülebilir. Yapay Sinir Ağlarının vekil olarak kullanıldığı Evrimsel Strateji algoritmasını tekrarlı deterministik optimizasyon prosedürlerinin hesaplama yüklerini azaltmak için daha önce sundukları yaklaşımı geliştirerek, yapının dayanımı ile ilgili ek olasılık kısıtlamalarını da dikkate alarak güvenilirlik tabanlı yapısal optimizasyon problemleri için de bir yaklaşım geliştirirler. Deterministik kısıtlama kontrolünü ya da güvenilirliğe dayalı optimizasyon durumlarında hem deterministik hem de olasılıksal kısıtlama kontrollerini gerçekleştirmek için eğitimli bir Yapay Sinir Ağı kullanılır. Yapay Sinir Ağı, güvenilirliğe dayalı yapısal optimizasyon süreçlerinde, güvenilirlik analizinin doğasında bulunan yaklaşımlar ve Monte Carlo Simülasyonu'nun (MCS) gerektirdiği zaman alıcı tekrarlı analizlerin tutarlı tahminlerini yapmak üzere bir vekil model olarak görev yapar. Her Monte Carlo Simülasyonu için Yapay Sinir Ağı, seçilen geleneksel elastoplastik analizlerden elde edilen mevcut bilgiler kullanılarak eğitilir. Sınır durum analizi verileri, Yapay Sinir Ağı'nı eğitmek için kullanılan giriş ve çıkış çiftlerini elde etmek için işlenir. Eğitilen Yapay Sinir Ağı daha sonra farklı temel rastgele değişken kümeleri nedeniyle kritik yük faktörünü tahmin etmek için kullanılır. Test problemlerinin sonuçları incelendiğinde önerilen her iki prosedürün doğasından dolayı tahminler ve kesin analiz değerleri arasında hata içermesinden dolayı kısıt ihlallerinin önüne geçmek ve Yapay Sinir Ağını tahminlerinin etkinliğinin arttırılması adına test verilerindeki hatalar üzerinden bir düzeltme yaklaşımı uygulandığı dikkat çekicidir.

Nguyen ve Vu (2020) ‘Hesaplamalı Yapısal Optimizasyon için Derin Öğrenme’ başlıklı çalışmalarında yapısal optimizasyonda uygun modellenmiş Yapay Sinir Ağları’nın Sonlu Elemanlar analizinin yerine çok tutarlı tahminsel değerler üretebileceğini ve kullanılabileceğini göstermişlerdir. Yapılan çalışmada Yapay Sinir Ağları ile oluşturulan tek katmanlı ve çok katmanlı derin öğrenme modeli için uygun mimari, çeşitli aktivasyon fonksiyonlarını içeren eğitim parametreleri ve model optimize edicilerle, 10 çubuklu,25 çubuklu, 52 çubuklu ve 200 çubuklu test problemlerinde uygulanmış ve sonuçlar Sonlu Elemanlar analizleri yerine Yapay Sinir Ağları’nın kullanılabilirliğini göstermiştir. Çalışmada kullanılan Yapay Sinir Ağı modelinin eğitim ve test verilerinin sabit toplam tekniği kullanılarak üretilmesi, Sonlu Elemanlar analizinin çok defa çağrılmasını içermekte ve hesaplama maliyetlerini arttırmasından kaynaklı olarak önerilen modelin amacına ters düşmektedir.

Lee vd. (2017) ’Yapısal Mühendislik için Derin Öğrenme’nin Arka Plan Bilgileri’ başlıklı çalışmalarında Yapısal Mühendislik’te Sonlu Elemanlar analizlerini simüle etmek için kullanılacak Derin Öğrenme modelinin tahminsel değerlerinin tutarlılığını ve model performansının arttırılmasına yönelik 10 çubuklu kafes sistem üzerinde çeşitli deneyler gerçekleştirmişlerdir. Belirli eğitim parametreleri ve iyileştirmeler sunulmuş, çeşitli Aktivasyon Fonksiyonları, optimizasyon teknikleri ve farklı ara katman sayıları ile elde etikleri sonuçları karşılaştırmalı olarak paylaşmışlardır. Yapısal Mühendislik’te, Derin Öğrenme modeli üzerinde yapılacak çeşitli teknikler ve yaklaşımlarla çeşitli yapısal problemler üzerinde daha etkili bir şekilde kullanılabileceğini göstermişlerdir. Ancak ilgili çalışmada önerilen modeller basit bir kafes sistem test problemi olan 10 Çubuklu Kafes Sistem üzerinde test edilmiş, farklı test problemleri üzerinde modellerin etkinliğine veya genelleştirilebileceğine dair bir bilgiye rastlanmamıştır.

Mote ve Kumar (2019) ‘Çelik Yapılar’ın İlk Tasarımı için Yapay Sinir Ağı Kullanımı’ başlıklı çalışmalarında tasarım ve analiz döngülerinin sayısını önemli ölçüde azaltmak ve pratik hususlar ve diğer tasarım kısıtlamaları ile uygulanabilir bir çözüme ulaşmak için Çelik Yapılar’ın ilk tasarımında Yapay Sinir Ağları kullanmaktırlar. Yapay Sinir Ağları tarafından tahmin edilen çıktılar, tasarımda bir başlangıç noktası olarak kullanılabilmektedir. Çalışmanın kapsamı, bir ayaktan yüklenen basınç altındaki bir açı kesitinin ilk tasarımını tahmin etmek için bir Yapay Sinir Ağları geliştirmeyi içermektedir. Yapay Sinir Ağı mimarisi bir girdi, bir ara katman ve bir çıktı katmanından oluşmaktadır. Giriş katmanında sırasıyla tasarım genişliği, tasarım yükü ve limit narinlik oranı değeri ($KL/r$) olarak üç girdi değeri bulunmaktadır, kesitin genişlik ($b$) ve kalınlık ($t$) çıktı değerleri olarak alınmıştır. Yapay Sinir Ağları çıktı verileri sürekli değişkenler veya ayrık değişkenler olarak ele alınarak problem için açı kesitinin boyutunu tahmin etmek üzere ayrı ayrı düşünülmüş, bu sebeple Regresyon ve sınıflandırma yaklaşımları için iki farklı model

belirlenmiştir. Sürekli değişkenler için Regresyon ve ayrık değişkenler için Sınıflandırma modeli düşünülmüş ve iki modelin performans karşılaştırmaları yapılmıştır. Yapay Sinir Ağları'nın optimal çözümleri yansıtan arama uzayından elde edilen veriler kullanılarak eğitilmiş olması koşuluyla, global optimum tasarım çözümlerine yakın çözümler bulabilmek için Yapay Sinir Ağları'nın kullanılabileceği sonucuna varmışlardır.

Krempser vd. (2012) 'Yapısal Optimizasyon Problemleri için Vekil Modeller Tarafından Desteklenen Diferansiyel Evrim' başlıklı çalışmalarında optimizasyon problemlerinde popüler olarak kullanılan popülasyon tabanlı Metasezgisel'lerden biri olan Diferansiyel Evrim'in hesaplama açısından pahalı problemlerde genel performansının iyileştirilmesi için En Yakın Komşuluk teknikleri ve Doğrusal Regresyon tekniğini içeren bir vekil model yaklaşımı önermişlerdir. Kullanılan Metasezgisel'in farklı olması ve vekil model için farklı bir Makine Öğrenmesi yaklaşımı sunulmasına rağmen tez konusu olarak sunulan vekil modelin amacına benzer bir probleme işaret etmektedir. Aynı şekilde kafes yapıların ağırlık minimizasyonunu önerilen prosedürün performansını değerlendirmek için kullanmıştır. Burada sunulan vekil destekli Diferansiyel Evrim teknikleri, farklı varyantlar kullanan Diferansiyel Evrim'in standart sürümleriyle karşılaştırılmıştır. Deneyler, benzer şekilde, sürekli ve ayrık tasarım değişkenlerine sahip beş test problemini içeren, altı farklı optimizasyon probleminden oluşmaktadır. Çalışmadaki optimizasyon süreçlerinde vekil modelin ürettiği adayların en iyisinin Sonlu Elemanlar analizine gönderilmesi ve değerlendirilmesiyle birlikte gereksiz hesaplama maliyetlerinin düşürülmesinin önemli olduğu düşünülmektedir.

Kaveh vd. (2008) 'Genetik algoritma ve Sinir Ağları Kullanılarak İletim Kulelerinin Optimum Tasarımı' başlıklı çalışmalarında iletim kulelerinin eş zamanlı şekil, boyut ve topoloji optimizasyonu süreçlerinde tasarım bilinmeyenlerinin sayısını azaltmak ve analizlerde zaman-yoğun matrislerinin ters çevrilmesi ihtiyacını ortadan kaldırmak için Enerji Yönetimi ve Kuvvet Yöntemi'ni birleştiren bir hesaplama yaklaşımı geliştirmişlerdir. Optimizasyon sürecini gerçekleştirmek üzere Göç Genetik Algoritması'nın kullanıldığı çalışmada, önerilen hesaplama yaklaşımıyla birlikte kullanılan Genetik Algoritma, sınırlı sayıda elemana sahip kulelerin optimum tasarımlarında başarılıdır. Ancak uygulamada karşılaşılan birçok unsuru bünyesinde barındıran, çok elemanlı iletim kulelerinin analizlerinde hesaplama hızı yetersiz kalmış ve zaman zaman yerel optimumda sıkışmıştır. Global optimuma yakın çözümlere ulaşmak ve analiz sayılarının azaltılması için Yapay Sinir Ağları'nın özelleşmiş bir modeli olan Radyal Bazlı İleri Beslemeli Sinir Ağı kullanılması önerilmiştir. Seçilen sinir ağı bulanık bir yorumlayıcı olarak çalışmakta ve sözü edilen problemde girdi verilerine uygun olarak spesifik çıktı değerleri ürettiği için tercih edilmiştir. Eğitim verileri başlangıçta nesildeki aday çözümlerin kesin analizlerinin gerçekleştirildiği Genetik Algoritma ile optimizasyon süreci

boyunca toplanmaktadır. Yeterli eğitim verisinin toplanmasıyla Radyal Bazlı İleri Beslemeli Sinir Ağı eğitilmekte ve sonrasında vekil olarak devreye girerek analiz sonuçlarına yaklaşık tahminler üretmektedir. Süreç boyunca Radyal Bazlı İleri Beslemeli Sinir Ağı tarafından jenerasyonun küçük bir yüzdesi kesin analize gönderilir ve amaç fonksiyonunun nesildeki adayların ortalamasındaki sapmadan yola çıkarak modelin verimliliği takip edilmektedir. Model verimliliği kabul edilebilir bir yüzdenin aşağısına düştüğünde ağ eğitimi tekrarlanır. Bu yaklaşımla sonraki nesillerin tutarlı tahminleri için modelin arama uzayını kapsaması ve enterpolasyon yeteneğini kazandırılması ayrıca dikkat çekicidir. Benzer bir ağ eğitim yaklaşımı ilgili tez konusu olarak sunulan vekil modelin optimizasyon sürecinde de uygulanmıştır.

Yu vd. (2018) 'Vekil Destekli Hiyerarşik Parçacık Sürü Optimizasyonu' başlıklı çalışmalarında tez konusu olarak sunulan çözüm yaklaşımına benzer olarak, problemlerin global optimumlarının bulunmasında Metasezgisel optimizasyon algoritmalarının uygunluk değerlendirmelerinin uzunluğundan kaynaklı olarak hesaplama maliyetlerinin yüksek olması ve gerçek dünya problemlerinde pratikte uygulanamaması problemine çözüm olarak Yapay Sinir Ağı tabanlı bir vekil model önermişlerdir. Standart Parçacık Sürü Optimizasyonu algoritmasından ve bir Sosyal Öğrenme Parçacık Sürü optimizasyon algoritmasından oluşan yüksek boyutlu problemler için vekil destekli hiyerarşik Parçacık Sürü iyileştiricisi geliştirmişlerdir. Burada Parçacık Sürü Optimizasyonu ve vekil destekli hiyerarşik Parçacık Sürü Optimizasyonu, arama alanını keşfetmek ve kullanmak için birlikte çalışır ve eş zamanlı olarak vekil modelin küresel ve yerel performansını iyileştirmektedir. Vekillerin yönetimindeki uygunluğunun yanında karmaşık optimizasyon problemlerinin küresel optimum değerlerini yakalamada çok uygun olduklarını düşündükleri Gaussian Süreci'nin (GP) hiperparametrelerinin optimize edilmesinin çok zaman alıcı olabileceğini düşündüklerinden dolayı vekil olarak mevcut sürünün aradığı bölgedeki uygunluk fonksiyonunun global profiline yaklaşırken, diğer yandan uygunluk ortamının yerel ayrıntılarını öğrenebildiği için Radyal Bazlı İleri Beslemeli Sinir Ağı'nı tercih etmişlerdir. Radyal Bazlı İleri Beslemeli Sinir Ağı modelinin oluşturulması için Latin Hiperküp Örneklemesini kullanarak arama uzayından popülasyon büyüklüğünden daha fazla adayları içeren bir başlangıç çözümü oluşturulmuş ve tam uygunluk fonksiyonu ile adaylar değerlendirilerek bu değerler arasından en iyi 'N' tanesi seçilerek bu adaylar arasından da yinelenmemiş en iyi 'P' kadarı alınarak Radyal Bazlı İleri Beslemeli Sinir Ağı modeli eğitilmiştir. Çalışmada genelleştirilmiş bir Radyal Bazlı İleri Beslemeli Sinir Ağı modelinin detayları paylaşılmamıştır. Önerilen vekil destekli hiyerarşik Parçacık Sürü ile Radyal Bazlı İleri Beslemeli Sinir Ağı'nin en iyi adayları bulunmuş ve bu adaylar tam uygunluk fonksiyonu ile uygunluk değerleri bulunarak bir listede toplanmıştır. Listedeki değerlerin değişimlerine göre Radyal Bazlı İleri Beslemeli Sinir Ağı modelinin güncellenmesi ile ağın tahmin yeteneğinin performansı ve modelin kapsayıcılığı

arttırılmıştır. Test problemleri üzerinde sunulan model ile yapılan optimizasyon süreçleri boyunca sürüdeki parçacıkların uygunluğunun vekil model ile tahmin edilip yalnızca en iyilerin uygunluk fonksiyonu ile değerlendirilerek Parçacık Sürü Optimizasyonu'nun üzerindeki hesaplama yükünün azaltıldığı paylaşılmıştır.

Tashakori ve Adeli (2002)'nin 'Sinir Dinamiği Modeli Kullanılarak Soğuk Şekillendirilmiş Çelik Uzay Yapılarının Optimum Tasarımı' başlıklı çalışmalarında tasarımlarında doğrusal olmayan, karmaşık kurallar ve süreksiz kısıtlar bulunduran resmi bir optimizasyon yaklaşımı olmadan en ekonomik kesitlerin seçiminin neredeyse imkansız olarak değerlendirildiği, ticari olarak temin edilebilen soğuk şekillendirilmiş kesitlerden üretilmiş uzay kafeslerin optimum tasarımı için Adeli ve Park'ın patentli hibrit CPN-nöral dinamik modeli (ABD, Patent 5.815.394, 29 Eylül 1998) (Adeli ve Park, 1995; Park ve Adeli, 1997), Amerikan Demir ve Çelik Enstitüsü (AISI) spesifikasyonuna uygun olarak uyarlanmıştır. Çalışmada modelin, mükemmel yakınsama sonuçlarına sahip 1548 üyeli büyük bir yapı da dahil olmak üzere, uzun açıklıklı ticari binalarda çatı yapıları olarak yaygın olarak kullanılan birkaç uzay kafes kirişi için minimum ağırlık tasarımını bulmak için kullanıldığı belirtilmiştir. Bu çalışma Yapay Sinir Ağları tabanlı bir optimizasyon modelin direk olarak gerçek dünya kafes sistem problemlerine uyarlanabilmesi adına önemli bir çalışma olarak görülmektedir.

Du vd. (2021) 'Yapı Tasarım Problemleri için Hiperparametre Optimizasyonunun Makine Öğrenimi Modelleri Üzerindeki Etkisini Anlamak' başlıklı çalışmalarında Yapı Mühendisliği'nde Sonlu Elemanlar simülasyonlarını kullanarak tasarım değerlendirmelerinin hesaplama maliyetini azaltmak için, bilgisayar destekli mühendislik tasarımında tasarım değişkenleri arasındaki karmaşık ilişkileri ve büyük veri kümelerinden gelen yanıtları öğrenme yetenekleri nedeniyle vekil modeller olarak kullanılan Makine Öğrenmesi Regresyon modellerinin hiperparametrelerin optimal değerlerini aramak için bir yaklaşım önerilmektedir. Gauss Süreç Regresyonu, Destek Vektör Makineleri (SVM), Rastgele Orman Regresyonu (RFR) ve Yapay Sinir Ağı (YSA) olmak üzere vekil olarak sıklıkla kullanılan dört Makine Öğrenmesi algoritmalarının optimum hiperparametreleri, yapı tasarımı optimizasyonunun dört karşılaştırmalı olarak test edilmiş, vekil modellerin doğruluğu, sağlamlığı ve hesaplama maliyetleri üzerindeki etkisi araştırılmıştır. Hiperparametre optimizasyonu tarafından yapılan iyileştirmelerin, çeşitli karmaşıklıklara sahip spesifik problemlere bağlı olduğu belirlenmiştir. Hiperparametre optimizasyonu performansı, karışık değişkenler ve görece daha yüksek boyutlu tasarım alanı ile karmaşık problemler ele alınırken sabit olmamaktadır. Hiperparametrelerin ayarlanmasından önceki ve sonraki eğitim maliyetleri de tek bir eğitim süreci ile değerlendirilmektedir. Rastgele Orman Regresyonu (RFR)'nun ardından Yapay Sinir Ağları'nın eğitim süresinin, hiperparametre optimizasyonuna karşı en yüksek hassasiyeti gösterdiği tespit edilmiştir.

Genel olarak hiperparametre optimizasyonu vasıtasıyla düşük hata ile Makine Öğrenmesi algoritmaları tabanlı vekil modellerin geliştirilebileceği karşılaştırmalı test sonuçları ile paylaşılmıştır. Hiperparametrelerin doğrudan hazır Makine Öğrenmesi algoritma paketleri tarafından sağlanan varsayılan değerlerle veya deneme yanılmaya dayanan değerlerle belirlenmesi yerine bu hiperparametrelerin optimizasyonu için genel bir yaklaşım önerilmesi açısından önemli bir çalışma olarak görülmektedir.

Çeşitli kısıtlar içeren optimizasyon problemlerinde yapılan eğitimin verimine bağlı olarak hata fonksiyonu ”Mutlak Fark” veya ”Ortalama Karesel Hata” seçildiğinde, hedef çıktılar ve vekilin tahmin ettiği çıktılar arasında bir fark olacaktır. Bir optimizasyon sürecinde sadece vekilin çıktı olarak ürettiği tahminsel çıktıların kullanılması hem kısıt ihlallerine sebep olacak hem de kullanılan optimizasyon algoritmasını optimalden uzak sonuçlara götürecektir. Bu soruna cevap arayan Lee vd. (2007) ‘Kısıtlı Yaklaşık Optimizasyon için Geri Yayılım Sinir Ağlarının Metamodellemesinde Türev ve Genetik Algoritma Tabanlı Yöntemler’ başlıklı çalışmalarında yaklaşık optimal çözümün kısıtlama fizibilitesini sağlayan Geri Yayılım Sinir Ağı (GSA) tabanlı bir vekil model önermişlerdir. Model bir dış ceza fonksiyonu yaklaşımı ile formüle edilmiş, hedef çıktılar ve yaklaşık çıktılar arasındaki eşitsizlik ilişkileri eklenerek kısıt fonksiyonuna eklenecek şekilde değiştirilmiştir. Böylece vekilin ürettiği çıktı değerlerinden kaynaklanan kısıt ihlallerinin önüne geçmeye çalışmışlardır. Önerilen yaklaşım standart bir 10 çubuk kafes problemi üzerinde incelenmiştir. Ayrıca önerilen geri yayılım sinir ağı (GSA) mimarisinin ağdaki katmanlar arasındaki ara bağlantı ağırlıklarını belirlemek için hem türev tabanlı yöntem hem de Genetik Algoritma (GA) olmak üzere iki yaklaşımla optimize edilmiş ve parametrelerin optimizasyonu için karşılaştırmalı sonuçları paylaşılmıştır.

Lagaros ve Papadrakakis (2004) ‘Yapısal Optimizasyonda Kullanılan Sinir Ağlarının Öğrenme İyiliştirmesi’ başlıklı çalışmalarında yapısal optimizasyon süreci için kullandıkları Evrim Stratejileri (ES)’nin Sonlu Elemanlar ile tekrarlı zaman alıcı yapısal analizlerden kaynaklı hesaplama maliyetlerini azaltmak için Yapay Sinir Ağlarının vekil olarak kullanıldığı hibrit bir model önermişlerdir. Ayrıca geleneksel olarak kullanılan İleri Beslemeli Yapay Sinir Ağı mimarilerinin, kullanılan aktivasyon fonksiyonunun kötü özelliklerinden dolayı kötü koşullu Jacobian Matrisleri üretmesinin muhtemelliğine değinerek, bu tip kötü koşullandırmaya birçok uygulamada rastlandığından ağ eğitiminin performansının, öğrenme süreci sırasında uygun şekilde güncellenmiş bir kazanç parametresi ile uyarlanabilir bir Sigmoid Aktivasyon Fonksiyonu ile ağ performansını iyileştirmeyi amaçlayan bir metodoloji de sunmuşlardır. Yapay Sinir Ağı tahminlerinin kalitesi, Yapay Sinir Ağı’nın koşullandırmasını iyileştirmek için önerilen yaklaşımın hesaplama avantajlarının gösterildiği, ES ile optimize edilmiş üç yapısal tasarım probleminde araştırılmıştır. Yapay Sinir Ağı tabanlı yapısal analiz sadece yaklaşık sonuçlar

verebildiği için özellikle kısıtların tahmin değerlerinin, arzu edilen ve arzu edilmeyen bölgeleri bölen sınıra yakın olduğu durumlarda, ortaya çıkabilecek tutarsız tahminleri önlemek için çıktı değerlerinde bir düzeltme yapılması önerilmektedir. Bu nedenle Yapay Sinir Ağları konfigürasyonunun maksimum test hatasına benzer şekilde izin verilen kısıtlama değerlerinin bir ”düzeltmesi” gerçekleştirilmiştir. Maksimum test hatası, test verileri arasında çıktı değerlerinin en büyük ortalama hatası olarak alınmış, tahmin edilen değerler, normal yapısal analizden elde edilen değerlerden daha küçük olduğunda, kısıtlamaların izin verilen değerleri, Yapay Sinir Ağları konfigürasyonunun maksimum test hatasına göre azaltılmış ve bu yaklaşımın kısıt ihlallerine sebebiyet verdiği ve tez konusu olarak sunulan çalışmada Yapay Sinir Ağları değerlerinin Sonlu Elemanlar tahminlerinde böyle bir yaklaşımla, herhangi bir şekilde yer verilmediği belirtilmelidir.

Messner (2019) ‘Periyodik Yapıların Mekanik Özellikleri için Evrişimli Sinir Ağı Tabanlı Vekil Modeller’ başlıklı çalışmasında periyodik kompozitlerin etkili mekanik özelliklerini hesaplamak için Evrişimsel Yapay Sinir Ağı modelini kullanmaktadır. Vekil model olarak görüntü tanıma süreçleri için özelleşmiş Evrişimsel Yapay Sinir Ağı modelinin kullanılması açısından özgün bir çalışma olarak nitelendirilebilir. İlgili çalışmada Evrişimli Sinir Ağı’nın, periyodik kompozitlerin mekanik özelliklerini öngören daha yavaş tam çözümlenmiş simülasyonlar için hızlı bir vekil model olarak hizmet edebileceği vurgulanmaktadır. Biçimsel olarak incelendiğinde Evrişimli Sinir Ağı bir görüntü Regresyon problemi olarak periyodik yapıların etkin mekanik özelliklerinin belirlenmesini sağlamaktadır. Yapının ikili Bitmap görüntüsü olarak tanımlaması verildiğinde, model homojenleştirilmiş, periyodik sistemin mekanik özelliklerini etkili bir biçimde tahmin edebilmektedir. Ancak bu vekil modeli oluşturmanın hesaplama maliyetine, eğitim veritabanını oluşturma maliyeti hakimdir. Amaç, periyodik yapıların etkin özelliklerini hesaplamak için hızlı vekil bir model sunmaktır ancak ağın eğitimi için 2 milyon veri üretilmiş ve bu veri setinin üretilmesi için Sonlu Elemanlar simülasyonunun defalarca çağırılması ile hesaplama maliyetlerinin bir hayli arttığı dikkat çekmektedir. Yine de çalışmada da belirttiği üzere aynı vekil modelin birden fazla uygulamada kullanılabilmesi, eğitim verilerinin üretilmesinin maliyetinin amortismana tabi tutulmasını sağlayabilir. Ortaya çıkan vekil modelin, ters problemleri çözmek için yeterince doğru olduğu görülmüş ancak model doğruluğunu iyileştirebilecek ve gerekli eğitim veritabanının boyutunu azaltabilecek mekanik problemler için uzmanlaşmış geliştirilmiş ağ topolojilerinin belirlenmesinin gerekli olduğu görülmüştür.

Kallioras vd. (2020) ‘Derin Öğrenme Yoluyla Hızlandırılmış Topoloji Optimizasyonu’ başlıklı çalışmalarında çözümleri son derece zaman alıcı olan topoloji optimizasyon problemlerini çözmek amacıyla Cezalandırmalı Katı İzotropik Malzeme hesaplama verimliliğini arttırmak için Derin İnanç Ağları entegre edilmiş bir yaklaşım

geliştirmişlerdir. Topoloji optimizasyon prosedürünü hızlandırmayı amaçlayan bu çalışmada, derin öğrenmeye dayanan yeni bir iki aşamalı metodoloji önerilmiştir. Cezalandırmalı Katı İzotropik Malzeme , topoloji optimizasyon probleminin çözümü için iyi bilinen bir yaklaşım olarak değerlendirildiğinden, özellikle, topoloji optimizasyon problemlerini çözmek için Derin İnanç Ağları, Cezalandırmalı Katı İzotropik Malzeme yaklaşımına entegre edilmiştir. Ayrıntılı olarak, Derin İnanç Ağları, nihai sonucu ile Cezalandırmalı Katı İzotropik Malzeme yaklaşımının ilk yinelemeleri boyunca alanın her Sonlu Elemanının yoğunluk değerleri arasındaki daha yüksek dereceli bağlantılarını keşfetmek için kullanılmıştır. Derin İnanç Ağları, birkaç ayrıklaştırma uygulayan iki tipik topoloji optimizasyonu test örneği kullanılarak oluşturulan iki eğitim veri kümesi üzerinde eğitilmiştir. Eğitilen ağ daha sonra, önerilen metodolojinin performansının ve sağlamlığının eğitim parametreleri, dikkate alınan veri kümeleri ve sınıflandırma türü açısından test edildiği çeşitli 2B ve 3B topoloji optimizasyonu test örneklerine uygulanmıştır. Çalışmanın ihtiyaçları doğrultusunda yapılan testlerden hareketle, önerilen metodolojinin eğitim ve sınıflandırma için kullanılan parametreler ve veri setlerinden bağımsız olarak her durumda Cezalandırmalı Katı İzotropik Malzeme yöntemini hızlandırmada başarılı olduğu gözlemlenmiştir.

Sharifi vd. (2019) ‘Hücresel Çelik I-Kirişlerin Yanal Burulma Burkulma Dayanımı Değerlendirmesi için Sinir Ağları’ başlıklı çalışmalarında Yanal Burulma Burkulması’na maruz kalan Hücresel Çelik Kiriş’lerin hasar yüklerini tahmin etmek için modellenmiş Yapay Sinir Ağı’na dayalı yeni bir formül önerilmiştir. Önerilen formül, Yanal Burulma Burkulması altında çelik hücresel kirişlerin dayanım kapasitesinin tahminlerini içermektedir. Hücresel Çelik Kiriş’in enine kesit özellikleri, dairesel hücrenin çapı, iki bitişik hücrenin merkez çizgileri arasındaki boşluk, yük ve mesnet konumları arasındaki boşluk, çeliğin tipi ve narinliği gibi girdi parametreleri ile sinir ağı modeli kullanılarak eğitilmiştir. Yanal burulma burkulma dayanımını etkileyebilecek tüm parametreler dikkate alınarak pratik bir formül değerlendirilmeye çalışılmıştır. Herbir giriş parametrelerinin tutarlılığını belirlemek için Garson Algoritması (Garson, 1991) kullanılarak bir dizi duyarlılık analizi de gerçekleştirilmiştir. Önerilen tasarım formülün, Hücresel Çelik Kiriş’lerin dayanımının altında yatan fiziksel davranışı yakalama yeteneğine sahip olduğu gösterilmiştir. Daha sonra önerilen formül ile Avustralya Standartı’ndan (AS4100) elde edilen tahminler arasında bir karşılaştırma yapılmıştır. Sonuçlar, önerilen formülün AS4100 tasarım kılavuzlarından daha doğru tahminler elde ettiğine dair bazı kanıtlar sunmaktadır.

Niutta vd. (2018) Çarpışmaya dayanıklı tasarımda kötü tasarlanmış problemler için yeni bir yaklaşım olarak ’Süreksiz Tepkilere Sahip Yapıların Tasarım Optimizasyonunda Vekil Modelleme’ başlıklı çalışmalarında artan hesaplama gücüyle, yapısal tasarım optimizasyonunda geçici doğrusal olmayan davranışlara maruz kalan çarpışmaya dayanaklı

yapıların etkili ve verimli bir şekilde tasarlanabilmesi için yeni bir vekil modelleme tekniği önermişlerdir. Tipik küresel vekil modellerin uygulanabilirliğinin, kritik süreksizlikler mevcut olduğunda sınırlı olması probleminden yola çıkarak sistem fonksiyonlarında büyük değişikliklere neden olan tasarım değişkenlerine ilişkin yanıtlardaki süreksizlikler ele alınmıştır. Burada süreksizliklerin sayısını ve tasarım alanındaki konumlarını belirlemek için bir yöntem geliştirilmiştir. Bu yöntem, süreksizliklerin varlığını, sayısını ve konumunu ve bunların yapısal tasarım optimizasyon sürecine entegrasyonunu tanımlamaktadır. Sayı ve konumdaki süreksizliklerin belirlenmesi, her sürekli alt alanın yerel bir vekil modelle tahmin edilmesini sağlar. Vekil tabanlı tasarım optimizasyonu kullanılırken, hesaplama eforunu azaltmak için yanıtlar yaklaşık değerlerle değiştirilmektedir. Sistem işlevleri, yani hedefler ve kısıtlamalar artık vekiller tarafından verilmektedir. Süreksizlikler ortaya çıktığında, atlamalar arasındaki alt alanları yaklaşık olarak belirlemek için çoklu bir yaklaşım kullanılmıştır. Her yinelemede, optimizasyon algoritması, sistemin amaçlarının ve kısıtlamalarının değerlendirildiği tasarım alanında yeni bir tasarım tanımlar. Aynı sürekli alt etki alanındaki örnekler kümelenir ve süreksizlikleri bulmak için Sınıflandırma bir Destek Vektör Makinesi algoritması ile yapılmaktadır. Destek Vektör Makinesi ile ait olunan sınıflar tanımlanarak, yeni noktalar üzerinde tahminler yapılabilmektedir. Buradaki amaç, aynı sürekli alt alanın örneklerini aynı sınıfa kümelemektir. Destek Vektör Makinesi'ni Sınıflandırma için kullanılmasındaki amaç, sınıflar arasındaki sınırları belirlemek ve sonuç olarak süreksizliklerin yerini bulmaktır. Son olarak, vekil tabanlı bir tasarım optimizasyonu gerçekleştirilmektedir. Bir geçmeli süreksizliği gösteren iki çubuklu bir kafes kirişle başlayarak, bu yöntemin bu tür süreksizlikleri hesaba kattığı ve önerilen yöntemin, sürekli alt uzayların süreksizliklerini ve yerel yaklaşımlarını bulabildiği gösterilmiştir. Sonuçlar incelendiğinde yerel yaklaşımların çok daha doğru bir yanıt yüzeyi sağladığı ve buna karşılık yanlış optimum probleminden kaçınılmasının yanında yerel yaklaşımlı bir vekil model yaklaşımıyla birlikte hesaplama maliyetlerinin önemli oranlarda azaltılabileceği sunulmuştur.

Su vd. (2017) 'Karmaşık Mühendislik Yapısal Güvenilirliği Analizi için Gauss Süreç Tabanlı Dinamik Vekil Model' adlı çalışmalarında genellikle doğrusal olmayan karmaşık bir mühendislik yapısının performans güvenilirliğinin, Sonlu Elemanlar analizicgibi zaman alıcı bir bilgisayarlı hesaplama süreci değerlendirmesinden kaynaklı olarak büyük ölçekli karmaşık mühendislik yapılarında geleneksel güvenilirlik yöntemlerini kullanırken hesaplama verimliliği ve kesinliğini birleştirmenin zorluğundan bahsedilmiştir. Yapısal sistemlerin çökme olasılığının tahmini için üç klasik yöntemden biri olarak Birinci Dereceden Güvenilirlik Yöntemi (BDGY) veya İkinci Dereceden Güvenilirlik Yöntemi (İDGY) yerine performans fonksiyonları, boyutları veya rastgele değişkenlerin dağılımları şeklinde özel gereksinimlerinin olmamasından dolayı Monte Carlo Simülasyonu (MCS) tercih edilmiştir. Pratikte, mühendislik yapılarının çökme olasılığı genellikle oldukça küçük

olduğundan, doğruluğu sağlamak için MCS tarafından çok sayıda numune oluşturulması gerektiğinden, bu gibi durumlarda tek bir analiz için yüksek hesaplama maliyeti oldukça engelleyici hale gelebildiğinden yapısal güvenilirlik analizi için bir vekil model yaklaşımıyla bu yüksek hesaplama maliyetlerinin düşürülmesi amaçlanmıştır. Performans fonksiyonunun değerini doğru bir şekilde değerlendirmek için vekil model olarak Polinom Tabanlı Yanıt Yüzeyi modelleri, Yapay Sinir Ağları, Radyal Temel Fonksiyonu, Kriging ve Destek Vektör Makinesi gibi tekniklerin yerine iyi performansı ve arzu edilen analitik özellikleri nedeniyle, Gaussian Süreci (GS) tercih edilmiştir. Karmaşık performans fonksiyonları için çökme olasılığını değerlendirmek ve dinamik bir strateji kullanarak Gaussian Süreç Regresyonu (GSR) ile Monte Carlo Simülasyonu'nu (MCS) birleştiren dinamik bir vekil model geliştirilmiştir. Bu yöntemde, birkaç başlangıç eğitim örneği alınmış, ardından, ilk eğitim örnek veri tabanına eklenerek yeni bir nokta olarak her yinelemeli adımda En Olası Nokta'yı (EON) değerlendirmek için yinelemeli bir süreç ve dinamik bir strateji izlenmiştir. Gaussian Süreç Regresyonu vekil modeli, strateji tarafından güncellenmiş ve dinamik olarak düzeltilmiştir. Ek olarak, bir Monte Carlo popülasyonunun en yüksek ortak olasılık yoğunluk noktasını bulmak için En Olası Nokta'yı olasılıksal bir şekilde değerlendirmek için yeni bir yaklaşım önerilmiştir. Çökme olasılığı, GSR vekil modeli sayesinde, her yinelemeli adımda herhangi bir fazla Sonlu Elemanlar Analizi olmaksızın MCS tarafından hızla tahmin edilmiştir. GSR vekil modelin öngörüsü sayesinde aşırı hesaplamaya gerek kalmadan En Olası Nokta belirlenir. Bu nedenle, bir eğitim örneği, performans işlevine yapılan çağrı sayısında ciddi bir azalmayla sonuçlanan bir Sonlu Eleman Analizi anlamına gelmektedir. Önerilen yaklaşımı doğrulamak için literatürden farklı karmaşıklıklara sahip dört test problemi incelenmiştir. DGSR tabanlı MCS, hata olasılığının doğruluğunun MCS'ninkine çok yakın olduğu ve işlev çağrılarının sayısı oldukça az olduğu için bu durumların hepsinde iyi performans göstermekte olduğu sunulmuştur. GPR, eğitim örneklerinin bilgisini tam olarak kullanan olasılıksal bir tahmin mekanizmasına dayandığından, DGSR tabanlı MCS'nin ilk eğitim örneklerinden büyük ölçüde etkilenmediği de görülmektedir. Önerilen yöntemin yüksek Sonlu Eleman Analizi hesaplama maliyeti olan karmaşık mühendislik yapılarının güvenilirlik problemlerinin hızlı çözümü için kullanılabileceği paylaşılmıştır.

Yapısal sistemlerin tasarım optimizasyonu genellikle yinelemeli, zaman alıcıdır ve tasarımcının bilgisi ile sınırlıdır. Bu nedenle, bu tür problemlerden kaçınmak için hızlı bir tasarım optimizasyon şeması arzu edilir. Demirkan vd. (2013) 'Sinir Ağları Kullanılarak Yapısal Değişikliklerin Hızlı Değerlendirilmesi için Yeni Bir Tasarım Yaklaşımı' başlıklı çalışmalarında hesaplama açısından yoğun analiz içeren yapısal sistemlerin verimli kavramsal tasarımı için iki tasarım metodolojisini sunmakta ve bütünleştirmektedir. Yapısal değişikliğe tabi tutulan yapının yeniden analizi için orijinal sistemin Frekans Cevap Fonksiyonlarını (FCF) kullanır. Orijinal yapının modal analizi, gerekli serbestlik

derecelerinde alım değerlerini elde etmek için yalnızca bir kez gerçekleştirilir. Daha sonra, orijinal sistemin kabullerini ve yapının değişen kısmının Dinamik Rijitlik Matrisini (DRM) kullanarak, değiştirilmiş sistemin kabullerini hesaplamak için Yapısal Modifikasyon Yöntemi (YMY) kullanılır. Tüm yapının yeniden analizi Yapısal Modifikasyon Yöntemi (YMY) kullanılarak ortadan kaldırılabilir. Bu çalışmada kullanılan yöntem, orijinal Frekans Cevap Fonksiyonlarını (FCF) ve eklenen yapının dinamik sertliğini kullanmaktadır. Matrisin tersine çevrilmesi, yalnızca orijinal yapıdan genellikle daha az serbestlik derecesine sahip olan değiştirilmiş bölge için gerçekleştirilir. Alıcılar şeklinde hesaplanan Frekans Cevap Fonksiyonları (FCF), orijinal yapının tam alıcıları kullanıldığı sürece kesindir. Yapısal Modifikasyon Yöntemi (YMY), her değişiklik için her seferinde yeni bir tam yapıyı analiz etmeye kıyasla sistemin yanıtını çok daha hızlı vermesine rağmen, Yapısal Modifikasyon Yöntemi'ndeki (YMY) hesaplama süresi, alımların hesaplandığı serbestlik derecelerinin sayısından olumsuz bir şekilde etkilenir. Ayrıca, bu işlem birçok kez tekrarlandığında, özellikle de değişen yapının tasarım parametreleri optimize edilirken, Yapısal Modifikasyon Yöntemi (YMY) hesaplamaları oldukça uzun zaman alabilir. Mevcut çalışma, Yapısal Modifikasyon Yöntemi (YMY) çalıştırmaları aracılığıyla yeterli sinir ağı eğitim verisi elde edildiğinde Yapay Sinir Ağları tekniğinin tasarım sürecine uygulanabileceğini göstermektedir.

Bu yaklaşımdan yola çıkarak İki farklı test çalışması sunulmaktadır. İlk test çalışmasında girdiler stifnerin konumu olarak seçilirken, ikinci test çalışmasında bu stifnerin genişliği olarak alınmıştır. Eğitim setinin çıktıları, ilgili modifikasyona sahip sistemin Frekans Cevap Fonksiyonları (FCF) eğrileri olarak seçilmiştir. Yapının yanıtı, bir Yapay Sinir Ağları modelinin eğitimi yoluyla her frekansta elde edilir ve bu modeller yapının tüm spektrum davranışını oluşturmak için birleştirilir. Her iki test çalışmasında da girdi-çıktı setlerini hazırlamak için Yapısal Modifikasyon Yöntemi (YMY) yaklaşımı kullanılmıştır. İyi bir genellemeye ulaşmak için bir sinir ağını eğitmek, yeterli sayıda girdi-çıktı seti gerektirir. Giriş ve çıkış katmanlarının boyutu, giriş ve çıkış parametre dizilerinin boyutuna göre belirlenir. Ancak, gizli katmanların sayısı ve bunların içinde kullanılacak nöronların miktarı, problemin ihtiyaçlarına bağlı olarak deneme-yanılma yoluyla belirlenmiştir. Çok sayıda ağ yapısı üzerinde çalıştıktan sonra, her biri yeterli sayıda nöron içeren giriş ve çıkış katmanı arasına üç gizli katman yerleştirmek, Yapısal Modifikasyon Yöntemi (YMY) tekniğinin tahminleriyle en iyi korelasyonu sağladığı görülmüştür.

Yapısal Modifikasyon Yöntemi (YMY), modifikasyon ana bilgisayar yapısının daha küçük bir bölümünde gerçekleştirilirse etkili bir araçtır ve sinir ağını eğitmek için veri hazırlama adımını basitleştirir ve hızlandırır. Ardından, Yapay Sinir Ağları tekniği, birçok tasarım yinelemesinin çok hızlı bir şekilde gerçekleştirilebilmesi için eğitimsiz koşullar için

sistemin tepkisini tahmin etmek için kullanılabilir. Yapısal Modifikasyon Yöntemi (YMY), sinir ağı yaklaşımı ile entegre edildiğinde, değiştirilmiş sistem yanıtını tahmin etmek için daha etkili ve daha hızlı bir araç elde edilebilir. Önerilen enterpolasyon şemasıyla birleştirilen her iki yöntemin de sistem performansını iyileştirmek için modifikasyonlar gerektiğinde yapının dinamik tepkisini tahmin etmede çok verimli çalıştığı gözlemlenmiştir. Model, üç katmanla bile, incelenen yapı yapısal değişikliklere maruz kaldığında tasarım yinelemelerinin işlem süresini azaltabildi. Gelecekteki çalışmalarda Yapay Sinir Ağları modeli geliştirilebilir ve farklı yapılar için daha ileri çalışmalar yapılabilir. Ancak bu yazıda sunulan iki vaka çalışması, sistem performansını iyileştirmek için modifikasyonlar gerektiğinde yapıların dinamik tepkisini tahmin etmek için Yapay Sinir Ağları'nın Yapısal Modifikasyon Yöntemine (YMY) entegre edilebileceğini göstermektedir.

Im ve Park (2013) 'Parçacık Sürüsü Optimizasyonu, Vekil Modeller ve Bayes İstatistikleri Kullanılarak Stokastik Yapısal Optimizasyon' başlıklı makalelerinde, Parçacık Sürüsü Optimizasyonu (PSO), vekil modeller ve Bayes İstatistikleri'ni kullanarak yapısal optimizasyon problemlerini çözmek için bir yönteme odaklanmaktadır. PSO, global optimumu bulmak için tasarlanmış rastgele/stokastik bir arama algoritmasıdır. Ancak, PSO, gradyan tabanlı optimizasyona kıyasla birçok değerlendirmeye ihtiyaç duyar. Bu, PSO'nun yapısal optimizasyonun analiz maliyetlerini arttırdığı anlamına gelir. Stokastik optimizasyonda hesaplama maliyetlerini düşürmenin yöntemlerinden biri de vekil model tekniklerini kullanmaktır. Bu çalışmada, yanıt yüzeyi yöntemi (YYY) ve Kriging dahil olmak üzere vekil modeller kullanılmıştır. Vekil modeller kullanıldığında, kesin değerler ile yaklaşık değerler arasında bazı hatalar vardır. Bu hatalar, optimum değerlerin güvenilirliğini azaltır ve vekil modeller kullanmanın gerçekçi yaklaşımını ortadan kaldırır. Bu çalışmada, daha güvenilir sonuçlar elde etmek için Bayes İstatistikleri kullanılmıştır. Stokastik Yapısal Optimizasyon için vekil modeller ve Bayes istatistikleri kullanılarak önerilen yöntemin etkinliğini doğrulamak ve doğrulamak için iki sayısal örnek üzerinde optimize edilmiştir. Vekil modeller oluşturmak için örnek verilere ihtiyaç vardır ve güvenilir bir yaklaşım fonksiyonu elde etmek için önemlidirler. Bu nedenle, bu çalışmada, tasarım değişkenlerinin simülasyon noktalarını oluşturmak için bir deney tasarımı olarak Merkezi Bileşik Tasarım kullanılmıştır.

Hajela ve Berke (1991) 'Yapısal Analiz ve Tasarımda Nörobiyolojik Hesaplama Modelleri' başlıklı çalışmalarında yapısal analiz ve tasarımda sinirsel hesaplama stratejilerinin rolünü inceler. Çalışmanın temel odak noktası, statik yapısal analizlerin tahmini için Yapay Sinir Ağları'nın kullanılmasıdır. Bu tür modeller, yeniden analiz için hesaplama açısından verimli yetenekler sağlar ve sayısal optimum tasarımda uygulama için uygun görünmektedir. Bu çalışmada benimsenen Denetimli Öğrenme tekniklerine özel vurgu yapılarak, sinirsel hesaplama yaklaşımına genel bir bakış sunulur. Sayısal doğruluk

ve verimlilikle ilgili olan bu tür öğrenme stratejilerinin özellikleri, temsili yapısal optimizasyon problemleri bağlamında incelenir. Yaklaşım, yapısal analiz modülünü bir Yapay Sinir Ağı modeliyle değiştirmeyi amaçlamaktadır. Temel yaklaşım, statik yapısal analizde istenen yer değiştirmeyi veya gerilme tepkisini belirlemede kullanılan yükleme-yer değiştirme ilişkilerini göstermek için bu tür ağları eğitmektir. Böyle bir model, yapısal bir tasarım senaryosunda eleman kesit boyutları olabilen bir dizi girdi değişkenini kabul eder ve bir çıktı olarak optimizasyon sürecinde kullanılmak üzere davranışsal tepki değerlerinden oluşan bir vektör sağlar. İki ayrı Yapay Sinir Ağı modeli basit bir yapısal problem üzerinde test edilir. Yapısal analiz problemindeki temel özelliklerin tanımlanmasının düz fonksiyonel bağlantı ağlarının eğitiminde önemli olduğu gösterilmiştir. Bu tür ağlarda öğrenme oranının, özellikle büyük problemler için gizli katmanlara sahip ağlardan çok daha iyi olduğuna dair bir genelleme yapılmıştır. Ancak mevcut çalışmanın tarihi ve yapısal problemler bazındaki kısıtlı çalışmalar dikkate alındığında daha sonra bu genellemenin yanlış olduğu görülecektir. Girdi-çıktı katmanlarından oluşan düz fonksiyonel sinir ağlarının yapısal problemlerin doğrusal olmayan özelliklerini yakalamada istenilen performansı göstermediği ve problem karmaşıklaştıkça çok katmanlı bir Yapay Sinir Ağının, gizli katman sayılarının arttırılmasıyla fonksiyonel olarak problemin karmaşık yapısını daha iyi yakaladıkları görülmektedir. Diğer bir taraftan sonuçlar incelendiğinde doğası gereği Yapay Sinir Ağları tarafından oluşturulan tahmin değerleri ile gerçek analiz değerleri arasında hatalar olacağından yapısal optimizasyon problemlerinin çözümünde direk olarak YSA kullanılmış ve problem bazında kısıt ihlallerinin olduğu görülmüştür.

Popülasyon tabanlı Metasezgiseller ile yapısal optimizasyon süreçleri uygunluk, kısıtlar ve yapısal tepkilerin tekrarlı değerlendirilmesini içermesinden dolayı yüksek hesaplama ve zaman maliyeti yarattığı açıktır. Bu yüksek hesaplama ve zaman maliyetlerinin düşürülmesi için, (Srinivas ve Ramanjaneyulu, 2007), ‘Genetik Algoritmalar ve Yapay Sinir Ağları Kullanılarak Köprü Tabliyelerinin Optimum Tasarımı için Entegre Bir Yaklaşım’ başlıklı çalışmalarında, yapısal tasarım optimizasyonunda uygunluk ve kısıtlama ihlallerinin değerlendirilmesinde yapısal tepkilerin tahmini için Yapay Sinir Ağları kullanmışlardır. Modelin özellikle büyük ve karmaşık problemlerin çözümlerinde, hesaplama kaynaklarında önemli ölçüde tasarruf sağlayacağı paylaşılmıştır. İlgili çalışmada, T-kirişli köprü tabliyelerinin optimal tasarımı için Yapay Sinir Ağı ile bağlantılı olarak Genetik Algoritma ile bir optimizasyon prosedürü sunulmuştur. Nörogenetik olarak atfedilen hibrit bir model yaklaşımı ile optimizasyon sürecinin performansını etkileyen önemli faktörlerden birinin uygun Yapay Sinir Ağları mimarisinin ve eğitim setinin seçilmesi olduğu belirtilir. Ancak optimizasyon sürecinde Yapay Sinir Ağları’nın genelleme yeteneği ve tahmin performansı paylaşılmamıştır. Üç katmanlı bir ileri beslemeli geri yayılımlı Yapay Sinir Ağı, çeşitli köprü tabliyesi konfigürasyonlarına ve açıklıklarına karşılık gelen farklı veri setleri ile eğitilmiştir. Köprü tabliyesi konfigürasyon

parametrelerinin bir vektöründen oluşan girdi verileri, boyuna kirişlerin uzunluğu, genişliği, derinliği, sayısı ve aralığı, çapraz kirişlerin aralığı, vb. iken istenen çıktı verileri maksimum moment ve kesme kuvveti olarak belirlenmiştir. Yapay Sinir Ağları doğrulaması, eğitim setlerinden farklı olan test veri setleri kullanılarak gösterilmiştir. Eğitilmiş Yapay Sinir Ağları'nın kabul edilebilir hata toleransı içinde giriş ve çıkış modellerini eşleştirebildiği ve ağa yeni bir test veri seti sağlandığında tasarım yanıtlarını tutarlı bir şekilde tahmin edebildiği gözlemlenmiştir. Yapay Sinir Ağının Maksimum Tasarım Momentleri'ni ve Kesme Kuvvetleri'ni doğru bir şekilde tahmin edebildiği, tahmindeki hatanın çoğu durumda %1'den az olduğu paylaşılmaktadır. Vekil model olarak seçilmiş Yapay Sinir Ağları tabanlı Genetik Algoritma ile optimum çözüme yakınsamanın performanslı olduğu ve hesaplama süresini önemli ölçüde azalttığı sunulmuştur.

# 3. KAFES SİSTEMLER

## 3.1 Optimum Tasarım Yaklaşımı

Kafes yapılar, sürekli yükler altında düğümlere bağlanan, basma veya çekme gerilmelerine maruz kalan doğrusal elemanlardan oluşan iki veya üç boyutlu yapılardır. Kafes Sistemler köprüler, uzun açıklıklı çatılar ve iletim kuleleri gibi daha birçok yapıda yaygın olarak kullanılan ve mekanik özellikleri, imalat kolaylıkları gibi artılarından dolayı yapı mühendisliğinde yaygın olarak tercih edilmektedirler (Pholdee ve Bureerat, 2014).

Kafes Sistem tasarımları bir karar verme problemi olarak ele alınabilir. Kafes Sistemlerin tasarım süreçlerinde öncellikle sistemin matematiksel olarak modellenmesi gerekmektedir. Karar verme problemleri, verilen sınırlamalar altında problemin kalitesini temsil eden bir amaç fonksiyonu değerinin minimum veya maksimum olacak şekilde modellenmesini gerektirir. Amaç fonksiyonu genellikle yapının toplam maliyetini veya malzeme maliyetini en aza indirgemek olarak kabul edilir. Genel olarak mühendislik tasarımlarının mümkün olduğunca yüksek performanslı ve güvenli olması, maliyetlerinin ise olabildiğince düşük olması amaçlanmaktadır. Bir Kafes Sistem yapısında kullanılan malzeme miktarının, yapının ağırlığı ve maliyeti ile doğrudan ilişkili olduğu varsayımından yola çıkarak, Kafes Sistem'in tasarım verimliliğini ölçmek için bu yaklaşım kullanılabilir (Camp ve Bichon, 2004). Karar değişkenleri, kullanılacak bir kaynağın miktarını veya bazı faaliyetlerin seviyesini temsil eder. Karar değişkenleri kafes elemanların kesit özellikleri olarak alınırlar. Yapı ağırlığının minimize edilmesi seçilen malzeme ve doğrusal elemanların kesit ve boyutlarına bağlıdır. Bir karar verme probleminin çözümünde tasarımcının sağlaması gereken kısıtlar olarak adlandırılan belirli sınırlamalar vardır. Gerçek dünyada karar problemlerinde bu kısıtlar her zaman bulunmaktadır. Gerçek dünya mühendislik tasarımları birçok kısıt içermekte ve birçok kısıt altında performans ve güvenlikten ödün vermeden en az maliyetle tasarımların yapılması amaçlanmaktadır. Kafes Sistemin uygulanan yükler altında davranışından kaynaklı olarak yapıda, gerilmeler ve deplasmanlar oluşacaktır. Kafes Sistem'in yük taşıma kapasitesine göre gerilme ve deplasman değerlerine bir sınırlama getirilir, bu sınır değerlere göre gerilme ve deplasman kısıtları alınırlar. Bir karar verme probleminin optimum çözümü, karar verme problemindeki amaç fonksiyonu en uç değerine ulaşacak ve problemin kısıtları karşılanacak şekilde karar değişkenlerinin en iyi değerlerini tanımlar. Tasarım süreçleri mühendisin tecrübe ve sezgilerine göre şekillenen ve çözülmesi gereken birçok problemi beraberinde getirmektedir. Örneğin her bir elemanın kesitinin mevcut 42 profilden seçilmesi gereken

basit bir 10 çubuklu Kafes Sistem'in toplam $42^{10}$ olası seçeneği vardır (Nguyen ve Vu, 2020). Böyle bir problem için deneme-yanılma yöntemi mümkün değildir. Mühendis tarafından problemlere getirilen çözümler birkaç deneme ve yanılma ile optimuma yakın olduğu varsayımına dayanmakta, bu tasarımlar performans, güvenlik veya maliyet açısından beklenileni karşılayamamaktadır.

Kafes Sistemler'in optimizasyonu verilen kısıt veya kısıtların sağlanması şartıyla, boyut, şekil ve topoloji optimizasyonu olmak üzere üç ana başlık altında toplanabilir. Boyut optimizasyonu yapısal elemanların kesit alanları yani boyutlandırılması ile ilgilenir. Optimal tasarım problemlerinin boyutlandırılmasında tasarım değişkenleri, kafes elemanların kesit alanları gibi Sonlu Elemanlar modeliyle ilişkili boyutlandırma parametreleridir. Kesit alanlarının ya sürekli ya da ayrık değerler olarak mevcut olduğu varsayılır (Rajan, 1995).

Şekil optimizasyonu ise doğrusal elemanların bağlandığı düğüm noktalarının uygun koordinatlarda olması ile ilgilenir. Yapının şeklinin değişmesine izin verildiğinde, genellikle geliştirilmiş bir tasarım elde edilebilir. Tasarım problemlerinde, yapının şeklinin, tasarım değişkenleri olarak alınan belirlenen düğümlerin düğüm koordinatları gibi sınırı tanımlayan parametrelerin kullanımı yoluyla veya dolaylı olarak seçim yoluyla değişmesine izin verilir .

Topoloji optimizasyonu gereksiz doğrusal elemanları ve düğüm noktalarının sistemden kaldırılması veya gerekli olanların muhafaza edilmesini ifade eder. Kafesler gibi ayrık yapılar için Topoloji, yalnızca düğümlerin birbirine nasıl bağlanacağını değil, aynı zamanda kaç düğümün yerleştirileceği ve ayrıca bunların nasıl destekleneceği anlamına gelir. Tasarım değişkenleri olarak üye bağlantı ve destek koşullarının seçimi, dışbükey olmayan ve süreksiz bir tasarım alanına yol açar. Bunun yanında gerilme, deplasman ve kinematik stabilite kısıtlamalarının da sonuç etkisi Topoloji Optimizasyonu'nu daha zor hale getirerek problemin karmaşıklığını arttırmaktadır (Tejani vd., 2017; Savsani vd., 2016).

Bu optimizasyon türlerinin tasarım değişkenleri ayrık veya sürekli olabilir. Boyut, topoloji veya şekil optimizasyonunda yapının tek bir yönü (Kaveh ve Talatahari, 2009; Wang vd., 2002; Degertekin vd., 2019), her bir yönün sırayla ele alındığı çok seviyeli bir yaklaşım (Miguel vd., 2013) veya iki veya daha fazla yönün birlikte ele alındığı eşzamanlı bir yaklaşım (Mortazavi ve Toğan, 2016; Noilublao ve Bureerat, 2013; Ahrari vd., 2015) ele alınabilir. Yine de bir tasarımcının aynı anda üç tasarım değişkeni türünü optimize edebilmesinin daha etkili olduğu bulunmuştur. Bu, Eşzamanlı Şekil, Boyut ve Topoloji tasarımı olarak adlandırılır (Techasen vd., 2018).

Yapısal Optimizasyon'un kullanılmasıyla tasarım sürecini bir karar verme problemi olarak formüle etmek ve mevcut matematiksel optimizasyon tekniklerini kullanarak optimum çözümü elde etmek mümkün hale gelmiştir (Saka vd., 2016). Yapısal optimizasyonda öncelikle Matematiksel Programlama (MP) ve Optimallik Kriteri (OK) yöntemlerine başvurulmuştur (Feury ve Geradin, 1978). Matematiksel Programlama'da problemlerin alt problemlere ayrılması yaklaşımı vardır. Matematiksel Programlama teknikleri Kafes Sistem'lerin Yapısal Optimizasyonu'nda kullanışlı değildir. Bu teknikler amaç fonksiyonu ve kısıtların gradyan hesaplamalarını gerektirmektedir. Matematiksel Programlama yaklaşımı ile yapısal optimizasyon problemlerinde geleneksel olarak gradyan tabanlı optimizasyon algoritmaları kullanılmış ancak lokal bölgelerde takılarak global optimumdan bir hayli uzak çözümler sundukları ve hesaplama maliyetlerinin de bir hayli fazla olduğu görülmüştür. Hatta bazı problemlerde kısıt fonksiyonları sürekli olmamakta ve gradyanlar mevcut olmamaktadır. Diğer bir taraftan Matematiksel Programlama teknikleri bir Kafes Sistem'deki kesitleri sürekli değişkenler olarak varsaymaktadır. Ancak bu kesitler ayrık değişkenler olabilmekte ve bir listeden seçilebilmektedir. Ayrıca bu teknikler optimizasyon sürecindeki iterasyonların başlatılabilmesi için çözüm vektörünün ilk tahminini gerekmektedir. Eğer ilk tasarım noktası optimum çözümden uzak seçilirse ve tasarım probleminin birkaç yerel optimumu varsa, Matematiksel Programlama algoritmasının optimum çözüm olarak yerel optimumlardan biri ile sonuçlanması daha olasıdır. Bazen ilk tasarım noktası iyi bir tahmin değilse optimuma yakın herhangi bir çözüm elde edilmeyebilmektedir. Bu özelliklerinden dolayı Kafes Sistemler'in optimizasyon süreçlerinde matematiksel programa teknikleri yetersiz kalmakta, yalnızca basit problemlerin optimum çözümlerine yaklaşmada başarılı oldukları görülmektedir.

Optimallik Kriteri yöntemlerinde, optimum tasarım varsayımlarına dayalı olarak, basit özyineleme formülleri üretilmekte ve optimum tasarıma ulaşmak için gerekli iterasyon sayısı, tasarım problemindeki tasarım değişkenlerinin sayısıyla ilişkili olmamaktadır. Ancak bu yaklaşım, sürekli tasarım değişkeni varsayımının yapılması koşuluyla, pratik yapıların optimum şekilde tasarlanmasını sağlayabilmektedir. İlk önce sürekli tasarım değişkeni varsayımı ayrık değişkenlere sahip Kafes Sistemler'in optimum tasarımlarında başarılı olamamaktadır. Ek olarak, uygun bir başlangıç tasarım noktasının seçilmesi de gereklidir ve bu algoritmaların performansı seçilen ilk tasarım noktasına bağlıdır. Genel olarak, Matematiksel Programlama teknikleri veya Optimallik Kriteri yaklaşımı kafes yapıların optimizasyonu için ihtiyaçları karşılayamamıştır.

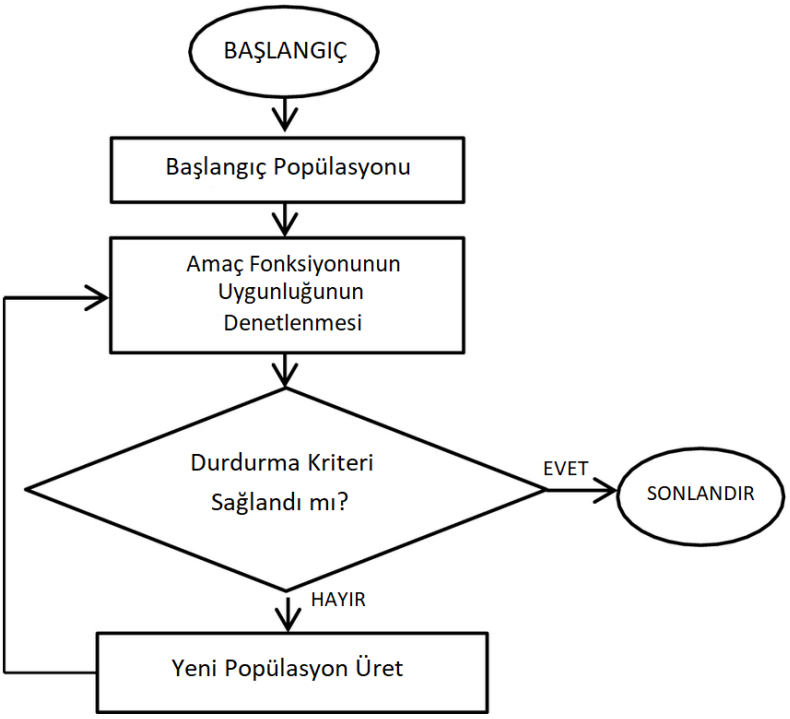
Son dönemlerde önerilen popülasyon tabanlı Metasezgisel Optimizasyon Algoritmaları, olasılıksal geçiş ilkelerini kullanarak, Optimallik Kriteri veya Matematiksel Programlama tekniklerine göre gradyan ve türev bilgisi gerektirmemelerinden dolayı lokal bölgelerde takılmadan global optimuma yakınsamada bir hayli başarılı olmuşlardır.

## 3.2 Metasezgisel Algoritmalar ile Optimizasyon Süreçleri ve Karşılaşılan Problemler

Matematiksel programlama algoritmalarının optimum çözümlere yaklaşımlarında bir rastgelelik bulunmamaktadır. Bu nedenle Deterministik Optimizasyon Teknikleri olarak da bilinirler. Bu yöntemlerde, aynı ilk tasarım noktası ile başlanırsa aynı nihai çözüme ulaşılır. Diğer tarafran metasezgiseller, optimum arayışlarında olasılıksal geçiş kurallarını kullanırlar ve belirli hareket operatörleri vasıtasıyla optimum arayışında bir rastgelelik bulunmaktadır. Bu da algoritmanın her çalıştırılışında farklı bir nihai çözüme ulaşılmasına sebep olmaktadır. Bu nedenle metasezgiseller Stokastik Optimizasyon Teknikleri olarak da bilinirler. Metasezgisel yöntemler, geleneksel olmayan stokastik arama ve optimizasyon yöntemleri olup, kombinatoryal optimizasyon problemlerinin çözümünün bulunmasında çok uygun ve etkilidirler. Amaç fonksiyonu ve kısıtlamaların gradyan bilgisine veya dışbükeyliğine ihtiyaç duymazlar. Ayrıca, amaç fonksiyonu ile kısıtlar arasında açık bir matematiksel ilişkiye ihtiyaç yoktur. Metasezgisel algoritmaların temel özellikleri, arama sürecini yönlendiren doğadan, sosyal kültürden, biyolojiden veya fizik yasalarından alınan belirli stratejileri taklit etmeleridir. Amaçları, küresel optimum olmasa da optimale yakın çözümler bulmak için bu yönetim mekanizmalarını kullanarak arama uzayını verimli bir şekilde araştırmaktır. Metasezgisel teknikler yaklaşık tekniklerdir ve elde edilen optimum çözümün global çözüm olduğuna dair matematiksel bir kanıt yoktur. Ancak, probleme özgü değildirler ve hem sürekli hem de ayrık tasarım değişkenleriyle pratik mühendislik tasarım optimizasyon problemlerinin çözümlerini elde etmede çok verimli ve sağlam oldukları kanıtlanmıştır (Saka vd., 2016).

Gradyan tabanlı bir optimizasyon yöntemi kullanıldığında, optimizasyon sürecinin en çok zaman alan kısmı, her iki durumda da, tüm matematiksel programlama optimizasyon yöntemlerinin önemli bir bileşeni olan duyarlılık analizi aşamasına ayrılmıştır. Öte yandan, olasılıksal aramaya dayalı kombinatoryal optimizasyon yöntemlerinin uygulanması, gradyan bilgisine ihtiyaç duymaz ve bu nedenle, hesaplama açısından pahalı olan duyarlılık analizi adımını uygulamaktan kaçınır.

Tüm popülasyon tabanlı metasezgisel algoritmaların iki ortak özelliği vardır. Birincisi, her yinelemede amaç fonksiyon değerlerini iyileştirmek için ilk tasarım ve süreçten başlamalarıdır. Yineleme süreci, maksimum yineleme sayısı veya işlev değerlendirmeleri olabilen sonlandırma kriterlerini karşılayana kadar gerçekleştirilir (Sonmez, 2018). Genel olarak popülasyon tabanlı Metasezgisel Optimizasyon Algoritmaları'nın akış şeması Şekil 3.1'te gösterilmiştir.




Şekil 3.1: Popülasyon Tabanlı Metasezgisellerin Genel Akış Şeması (Chong vd., 2021)

Metasezgisel algoritmaların çoğu, geleneksel olarak popülasyon olarak adlandırılan bir grup tasarım çözümünü kullanarak optimumu arar. Optimizasyon prosedürü, genellikle rastgele oluşturulan bir başlangıç popülasyonu ile başlar. Tasarım çözümlerinin popülasyonu daha sonra genellikle randomizasyonla ilgili bazı özel hesaplama işlemleri ile güncellenir veya geliştirilir. Operatörler, aramada yoğunlaştırma ve çeşitlendirme elde etmek için oluşturulur ve kullanılır. Popülasyon, bir sonlandırma kriteri karşılanana kadar iteratif olarak iyileştirilir. Metasezgiseller, fonksiyon türevlerini kullanmadıklarından, bir popülasyonda birkaç iterasyon için daha fazla iyileştirme bulunmadığında veya maksimum fonksiyon değerlendirmesi sayısına ulaşıldığında optimizasyon prosedürü durdurulur. Metasezgisellerle optimizasyon yöntemi, hemen hemen her türlü optimizasyonu çözmek için kullanılabilir (Pholdee ve Bureerat, 2014).

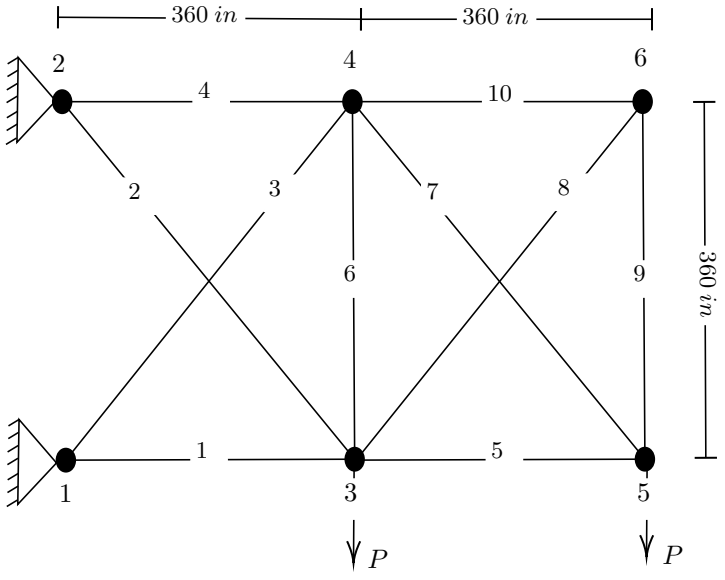
Metasezgisel Optimizasyon Algoritmaları arasında muhtemelen en yaygın kullanılan Holland (1975)'ın çalışmasında sunulan Genetik Algoritmalar (GA)'dır. GA'larda, uygun mutasyon, çaprazlama ve seçim operatörleri kullanılarak bir birey popülasyonu geliştirilir. Darwin'in doğal seçilim ve en uygun olanın hayatta kalması ilkelerine dayalı olarak, gelecek nesillere umut verici genetik bilgi yayılır ve optimizasyon sorununa çözümler sunar. GA'ların kafeslerin boyut optimizasyonu uygulanması Rajan (1995), Balling vd. (2006), Kaveh vd. (2008)'te bulunabilir.

Bir başka popüler optimizasyon yöntemi Benzetimli Tavlama Algoritması'dır. Yüksek enerjili durumda olan ve kademeli olarak minimum enerji seviyelerine soğutulan fiziksel sistemlerin tavlama sürecinden esinlenmiştir. Bu süreç ilk olarak (Kirkpatrick vd., 1983) tarafından bir optimizasyon algoritması olarak formüle edilmiştir. Kafeslerin boyut optimizasyonu için bir Benzetimli Tavlama uygulaması Lamberti (2008)'de bulunabilir.

Geem vd. (2001) tarafından önerilen Harmoni Arama kombinatoryal optimizasyon problemlerini çözmek için müzisyenlerin enstrümanlarının perdelerini doğaçlama yaparken aradıkları harmoniden ilham almıştır. Bu algoritma aynı zamanda kafes sistemlerin boyut optimizasyonu için de kullanılmıştır (Degertekin, 2012; Lee ve Geem, 2004).

Sürü Zekası algoritmaları kategorisinde Karaboga ve Basturk (2008), bal arılarının yiyecek arama davranışından esinlenen stokastik bir algoritma olan Yapay Arı Kolonisi algoritmasını önermişlerdir. Kafeslerin boyut optimizasyonu için bir Yapay Arı Kolonisi uygulaması Sonmez (2011)'de bulunabilir.

Aynı kategoride Parçacık Sürü Optimizasyonu, evrimsel bir avantaj sunan davranış kalıpları üreten üyeler arasında bilginin sosyal paylaşımına dayanır. Kennedy ve Eberhart (1995) tarafından tanıtılan yöntem, hareket eden parçacıkların hızlarını ayarlayarak tasarım uzayını araştırır. Kafeslerin boyut optimizasyonu için birçok Parçacık Sürü Optimizasyonu uygulaması literatürde bulunabilir (Talatahari vd., 2012; Dimou ve Koumousis, 2009; Perez ve Behdinan, 2007).




Şekil 3.2: On Çubuklu Kafes Sistem

Sunulan vekil model destekli optimizasyon tekniğinde kullanılan Jaya algoritması Venkata Rao (2016) tarafından ilk kez tanıtılmıştır. Popülasyon tabanlı bir metasezgisel olmasının yanı sıra diğer popülasyon bazlı metasezgisellerden farklı olarak çok sayıda spesifik kontrol parametresi içermemektedir. Yalnızca "Popülasyon Sayısı" ve "İzin Verilen Analiz Sayısı" parametre olarak tanımlanır. Sonuç olarak, Jaya algoritmasında yalnızca bu iki parametrenin kullanılması optimum çözümlerin aranmasında metasezgiselin sağlamlığını arttırmaktadır. Jaya algoritması, popülasyondaki her bir adayın en iyi çözüme yaklaşılması ve en kötü sonuçtan uzaklaştırılması prensibine dayanmaktadır. Literatürde Kafes Sistemler'in optimizasyonu için Jaya algoritması uygulaması Degertekin vd. 'nin (2018,2019) çalışmalarında bulunabilir.

Doğadan ilham alan metasezgiseller, çoklu hedefler, karışık tasarım değişkenleri türleri, amaç fonksiyonlarının düşük düzenliliği, çok sayıda doğrusal olmayan örtük kısıtlamalar ve pahalı ve/veya güvenilmez gradyanlar tarafından sunulan zorlukların üstesinden gelmeye yardımcı olabilir. Bununla birlikte, genellikle çok sayıda uygunluk ve kısıtlama değerlendirmesi gerektirirler. Örneğin on çubuklu kafes sistem probleminin (Şekil 3.2), boyut optimizasyonu için çeşitli metasezgisellerin performans karşılaştırmaları Çizelge 3.1'de sunulmuştur. Çizelge 3.1 incelendiğinde 10 çubuklu, basit bir boyut optimizasyonu problemi için bile, kısıt ihlallerinin belirlenebilmesi adına 50 bin civarında Sonlu Elemanlar çözücüsünün çağırıldığı görülmektedir. Mevcut simülasyon modellerinin boyutu ve karmaşıklığı nedeniyle, genellikle büyük hesaplama süreleri gerekmektedir.

Çizelge 3.1: On Çubuklu Kafes Metasezgisel Optimizasyon Sonuçları (Sonmez, 2018)

| | GA | PSO | YAK | JAYA |
|---|---|---|---|---|
| $A_1$ | 30.614 | 30.783 | 30.684 | 30.458 |
| $A_2$ | 0.100 | 0.100 | 0.100 | 0.100 |
| $A_3$ | 23.269 | 23.897 | 23.636 | 23.061 |
| $A_4$ | 15.072 | 14.748 | 14.136 | 15.265 |
| $A_5$ | 0.100 | 0.100 | 0.100 | 0.100 |
| $A_6$ | 0.527 | 0.100 | 0.639 | 0.539 |
| $A_7$ | 7.498 | 8.545 | 7.344 | 7.480 |
| $A_8$ | 21.298 | 20.954 | 20.284 | 21.145 |
| $A_9$ | 21.247 | 20.810 | 22.830 | 21.520 |
| $A_{10}$ | 0.100 | 0.100 | 0.100 | 0.100 |
| $W^{eniyi}$ | 5.061.428 | 5.076.730 | 5.070.360 | 5.060.967 |
| $K.ihl.$ | 0 | 2.46E−6 | 2.97E−4 | 0 |
| $W^{ortalama}$ | 5.070.853 | 5.116.834 | 5.087.724 | 5.063.210 |
| $N_A$ | 51050 | 50000 | 50137 | 50050 |

# 4. YSA DESTEKLİ VEKİL MODEL KAVRAMI

## 4.1 Vekil Model

Genellikle optimal tasarım süreci, tasarımcı veya mühendis tarafından ilk çözümlerin varsayılmasıyla başlar. İlk çözüm, tasarım kısıtlamalarını karşılayan bir dizi yineleme içeren yaklaşık çözümler üretmek için kullanılır. Bu süreç, optimal çözüme ulaşmak için farklı başlangıç çözümleri ile birden çok kez tekrarlanmaktadır. Birden fazla çözüm olabileceği için bazen tasarımcının bilgi birikimi ve geçmiş deneyimlerine dayanarak en kabul edilebilir çözüme karar vermesi gerekir. Maliyet fonksiyonunu ve malzeme mevcudiyeti, yapım kolaylığı vb. optimizasyon için pratik hususları formüle etmek, diğer taraftan insan sezgilerini ve geçmiş deneyimleri ifade etmek için bildirim kuralları oluşturmak zordur. Matematiksel Programlama ve Optimallik kriterleri yaklaşımlarının yalnızca sürekli değişkenlerden oluşan ve birkaç kısıt içeren, basit sayılabilecek optimizasyon problemlerinde başarılı olabildiği ve karmaşık gerçek dünya mühendislik problemleri için uygun yaklaşımlar olmadıkları görülmüştür (Bknz. Bölüm 3.1).

Metasezgisel algoritmalar, amaç fonksiyonlarının analitik ve türevlenebilir olmasını gerektirmediği ve daha iyi küresel arama kabiliyetine sahip olduklarından dolayı yapıların optimum tasarımlarında başarı ile uygulanmışlardır. Ancak, metasezgisel bir algoritmanın neredeyse optimal bir çözümü bulması için genellikle çok sayıda uygunluk değerlendirmesi gerekir. Bu uygunluk değerlendirmesi yani tasarım hedeflerinin performansı, Kafes Sistemler'in hesaplama maliyetleri bir hayli fazla olan Sonlu Elemanlar analizleri ile gerçekleştirilir. Popülasyon tabanlı metasezgisel algoritmalarda optimizasyon süreçleri iteratif bir iyileştirme süreci, optimuma yakın sonuçlar için iyileştirilen popülasyondaki her bir adayın uygunluk değerlendirmesini gerektirmektedir. Bu sebeple, gerçek dünyada yaygın olarak görülen geniş arama uzayları ve birçok kısıt içeren hesaplama açısından pahalı problemlere uygulanmaları büyük bir zorluk oluşturmaktadır. Bu pahalı hesaplama maliyetleri bir hayli fazla ve zaman alan süreçleri hafifletmek için simülasyon analizi veya fiziksel deneylerin yerini alacak vekil destekli optimizasyon teknolojisi geliştirilmiştir (Yu vd., 2019).

Sistemdeki yapısal elemanların davranışlarını ve tasarım performanslarını hesaplamak yerine tahmin etmek, vekil modelleme veya meta-modelleme olarak bilinirler. Vekil modeller, sınırlı bir veri kümesinden tasarım değişkenlerinin sürekli işlevini

belirleyerek tasarım hedeflerine yaklaşırlar (Nourbakhsh vd., 2018). Bu modeller, hesaplama maliyetini azaltmak için mühendislik tasarımı ve optimizasyonunda yaygın olarak kullanılmaktadır. Gerçek modele yaklaşmak için tasarım veya simülasyon veri kümelerinden regresyon yoluyla bir vekil model kurulur. Bu vekil modellerin kurulma yaklaşımları için çeşitli teknikler içeren Ong vd. (2003)'nin ilgili çalışmaları incelenebilir. Vekil modeller, ek simülasyonlar çalıştırmadan yeni tasarım alternatiflerinin yanıtlarını tahmin etmenin verimli bir yolunu sağlamaktadır (Du vd., 2021). Ayrıca vekil modeller, sınırlı analiz çalışmalarından elde edilen verilere dayalı simülasyonlardan çok daha kısa bir süre içinde sonuçları tahmin etmek için kullanılabilirler (Forrester vd., 2008). Stokastik algoritmalar kullanan pahalı optimizasyon yöntemleriyle ilgili olarak, hesaplama çabası büyük ölçüde azaltılabilir, böylece daha fazla tasarım değişkeni içeren daha karmaşık problemlere izin verilir ve hatta yüksek hesaplama çabası gerektiren problemlerin optimizasyonu sağlanır (Fang vd., 2016).

1959'da "Makine Öğrenimi" terimi ilk olarak Yapay zekanın öncülerinden Arthur Samuel tarafından tanıtılmıştır. Bu terminoloji, kelimenin tam anlamıyla, açıkça programlanmadan öğrenme yeteneği anlamına gelmektedir (Samuel, 2000). Makine öğrenimi uygulamalarının bazı örnekleri, Denetimli Öğrenme, Yarı Denetimli Öğrenme, Denetimsiz Öğrenme, Derin Öğrenme ve Pekiştirmeli Öğrenme'dir. Makine Öğrenimi, bir algılayıcı ile başlamış ve derin öğrenmeye doğru evrilmiştir. Makine Öğrenimi algoritmaları, tasarım değişkenleri arasındaki karmaşık ilişkileri ve büyük veri kümelerinden gelen yanıtları öğrenme yetenekleri nedeniyle vekil modeller olarak uygulanmaktadır (Du vd., 2021). Bu teknikler arasında Yapay Sinir Ağı (Lee vd., 2007; Santana Gomes ve Beck, 2013; Yu vd., 2018; Sharifi vd., 2019), Destek Vektör Makineleri (Hanna, 2007; Guo ve Bai, 2009), Radyal Temelli fonksiyon (Ong vd., 2003; Deng, 2006; Zhou vd., 2007) ve Gauss Süreci (Kriging) (Su vd., 2017; Echard vd., 2011; Kaymaz, 2005) en belirgin ve yaygın olarak kullanılan tekniklerdir.

## 4.2 Yapay Sinir Ağı Örneği

Yapay Sinir Ağı (YSA), insan beyninin ve sinir sisteminin işlevini taklit etmeye çalışan bir Yapay Zekâ biçimidir. Beyin gibi biyolojik sinir sistemlerinin bilgiyi işleme biçiminden ilham alan bir bilgi işleme paradigması olarak tanımlanabilir. Haykin (2009)'e göre, bir YSA, insan beyninin belirli bir ilgilenilen görevi yerine getirmesiyle aynı şekilde çalışmak üzere üretilmiş karşılaştırılabilir bir makine olarak tanımlanmıştır.

Bir problem için analitik veya deneysel bir model geliştirmek bazı durumlarda uygulanabilir olsa da, çoğu gelişmiş süreç çok yönlüdür ve bu nedenle daha az genel, daha

gerçekçi ve analitik modellerden daha ucuz modellere ihtiyaç duyulur. YSA'lar, geleneksel tekniklerin kesin kurallarla modellenmesinin çok karmaşık olduğu fiziksel bir sistemin, örüntü tanıma veya simülasyonunu gerektiren uygulamalar için çok kullanışlı araçlardır (Swingler, 1996). YSA tabanlı modellerin herhangi bir mühendislik problemini çözmedeki başarısında, eğitim için sunulan veri örnekleriyle öğrenmesi, girdi ve çıktılar arasındaki ilişkiler bilinmediğinde (Kara Kutu Problemler) veya fiziksel anlamın açıklanmasının zor olduğu durumlarda bile veriler arasındaki ince fonksiyonel ilişkileri yakalayarak, çıktı olarak kabul edilebilir yaklaşık tahminsel değerler üretmesi yatmaktadır (Prayogo vd., 2020).

Vekil model için kullanılan YSA için eğitim örnekleri sayısı ne kadar azsa, Sonlu Elemanlar analizlerinin sayısı o kadar az olacaktır. Bu nedenle, vekil modeli az ancak yeterli sayıda veri üzerinden eğitmek, Sonlu Elemanlar analizlerinin çağrı sayısını etkili bir şekilde azaltabilir (Su vd., 2017). Bir YSA mimarisinin öğrenme prosedürünün başarısını yöneten önemli bir faktör, eğitim setinin seçimidir. Eğitim algoritması olarak seçilen Geri Yayılım Algoritması'nın tatmin edici sonuçlar vermesi için, eksiksiz yapısal analizlerden elde edilen çıktı verileriyle birlikte, uzayda düzgün bir şekilde dağılmış yeterli sayıda girdi verisine ihtiyaç vardır. Uygun şekilde eğitilmiş bir Yapay Sinir Ağı'nın başlıca avantajı, kabul edilebilir bir yaklaşık çözüm üretmek için önemsiz bir hesaplama çabası gerektirmesidir (Papadrakakis vd., 1998). Model bir kez eğitildikten sonra, vekil model, Sonlu Elemanlar simülasyonundan çok daha hızlıdır. Aynı vekil model birden fazla uygulamada kullanılabilirse, eğitim verilerinin üretilmesinin maliyeti amortismana tabi tutulabilir. Bu durumda vekil yaklaşım, özellikle nispeten kolayca parametrelendirilebilen problemler olmak üzere, mühendislik pratiğinde yaygın olarak ilgi duyulan veya yaygın olan problemlere uygundur (Du vd., 2021).

Rogers (1994), yapısal analiz yapmak için YSA'yı uygulamış ve yönergeler geliştirmiştir. YSA'nın Matematiksel Optimizasyon tekniklerinden çok daha hızlı çalıştığını göstermiştir. Yapay Zeka'nın alt dallarından olan Makine Öğrenmesi tekniklerinden biri olan YSA ile oluşturulmuş Regresyon tahmin modellerinin, Yapı Mühendisliği'ndeki karmaşık problemlerin, doğrusal olmayan yapılarını yakalayabildiği ve son derece güç tahminleri bile hassas bir şekilde sağlayabildiği kanıtlanmıştır (Chou vd., 2016).

Sunulan vekil model Çok Katmanlı Yapay Sinir Ağı örneği olup, Sonlu Elemanlar analizlerinin döndürdüğü sonuçların sayısal tahminlerini gerçekleştirilebilmesi için Denetimli Öğrenme'nin alt dallarından biri olan Regresyon modeli olarak tasarlanmıştır. Vekil olarak kullanılacak YSA, matematiksel model ve davranışının anlaşılabilir olması

Çizelge 4.1: Örnek Eğitim Veri Seti

| **Deney No** | $x_1$ | $x_2$ | $f$ |
|---|---|---|---|
| **1** | 9 | 8 | 12 |
| **2** | 5 | 4 | 8 |
| **3** | 2 | 1 | 5 |

adına sonraki alt konu başlığı altındaki bölümlerde basit bir Regresyon problemi üzerinde sırasıyla açıklanacaktır.

### 4.2.1 Örnek Eğitim Veri Setinin Hazırlanması

Örnek problem için sayısal bir deney yapıldığı varsayılmıştır. İki farklı girdi değeri ($x_1$,$x_2$) alınarak, deney sonucunda bu girdilere göre bir çıktı değeri ($f$) üretildiği varsayılmıştır. Bu deney 3 kez tekrarlanmış ve elde edilen girdi ve çıktı değerlerinden bir veri seti oluşturulmuştur (Çizelge 4.1).

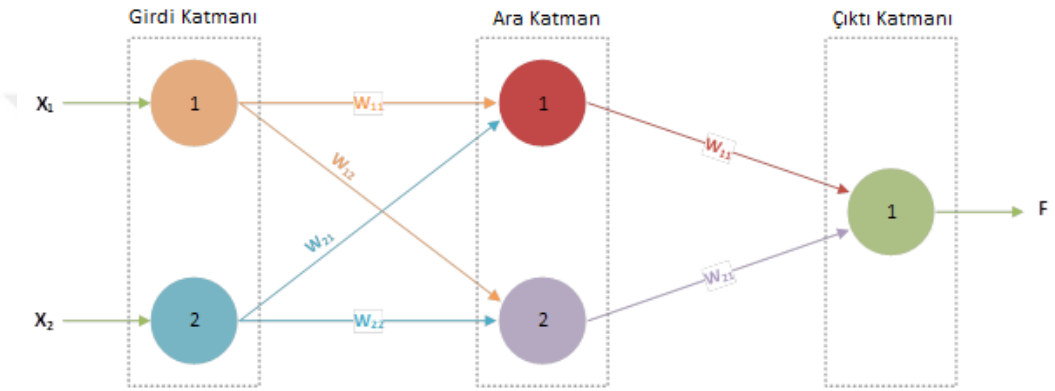
Veri setinin tamamı YSA'nın girdi ($x_1$,$x_2$) ve çıktı ($f$) değerleri arasındaki fonksiyonel ilişkiyi öğrenebilmesi için eğitim veri seti olarak alınmıştır. Örnek problem için YSA'nın matematiksel model ve arka planının açıklanabilmesi adına basit bir regresyon problemi olarak tasarlanmıştır. YSA'nın verilen girdi ve çıktı değerleri arasındaki fonksiyonel ilişkiyi yakalaması beklenmektedir. Veri setindeki girdi ve çıktı değerleri (Çizelge 4.1), aralarındaki matematiksel modelin bilinmediği varsayımıyla (4.1)'e göre hesaplanmıştır.

$$f(x_1, x_2) = x_1^2 - x_1 x_2 + 3 \tag{4.1}$$

Eğitim veri setinde bulunmayan $x_1$ ve $x_2$ girdi değerleri eğitilmiş YSA modeline girdi olarak verilecek ve YSA'nın $f$ çıktı değerini tahmin etmesi beklenecektir. $x_1$ ve $x_2$ girdi değerleri ile 4.1'e göre gerçek $f$ çıktı değeri hesaplanmıştır. Böylece (4.1) vasıtasıyla YSA'nın tahminleri ile gerçek çıktı değerleri arasındaki hatalar bulunabilmekte ve YSA tahmin performansı ölçülebilmektedir.

### 4.2.2 Örnek Yapay Sinir Ağı Tasarımı

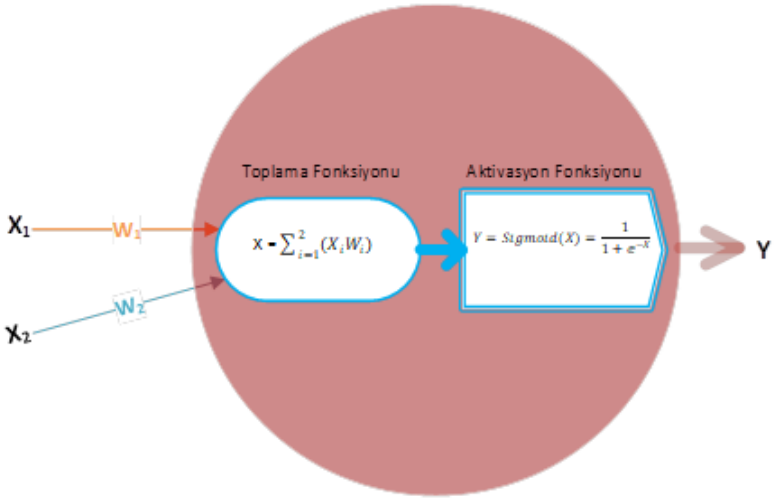
Yapay Sinir Ağı (YSA) kısaca, biyolojik olarak nöronlardan ve bu nöronları bağlayan sinapslardan esinlenilmiş bağlantılardan oluşan bir hesaplama modelidir. YSA modeli Denetimli Öğrenme Regresyon modeli olarak tasarlanmıştır. Bir veya daha çok ara katmandan oluşan YSA modellerine Çok Katmanlı Yapay Sinir Ağı veya Derin Öğrenme modeli olarak bilinmektedir. Örnek YSA modeli girdi ve çıktı katmanlarının yanında bir ara katmana sahiptir (Şekil 4.1). Ara katman sayısı arttıkça problemlerin karmaşıklığına bağlı olarak doğrusal olmayan ilişkilerin öğrenme performansı artmaktadır.




Şekil 4.1: Çok Katmanlı YSA Örneği

Şekil 4.1'de katmanlar renkli dairesel elemanlar olarak şematize edilen nöronlardan ve herbir nöronu birbirine bağlayan sinapslardan esinlenilmiş bağlantı elemanlarından oluşmaktadır. Örnek bir nöron detayı Şekil 4.2'de şematize edilmiştir. $X_1$ ve $X_2$ girdilerinin ara katman içindeki kırmızı nöron elemana ağırlıkları vasıtasıyla bağlanmaktadır $(W_1, W_2)$. Sinir hücresine gelen bilgiler, girdiler üzerinden nöron çekirdeğine ulaşmadan önce geldikleri bağlantıların ağırlığıyla çarpılarak nöron çekirdeğine iletilir $(X_i, W_i)$. Bu sayede girdilerin üretilecek çıktı üzerindeki etkisi ayarlanabilmektedir. Bu ağırlıkların değerleri pozitif, negatif veya sıfır olabilir. Ağırlığı sıfır olan girdilerin çıktı üzerinde herhangi bir etkisi olmamaktadır. Girdi değerleri nörona geldikten sonra nöron içindeki toplama fonksiyonu (4.2) vasıtası ile hesaplanır.

$$X = \sum_{i=1}^{2} (X_i W_i) \tag{4.2}$$



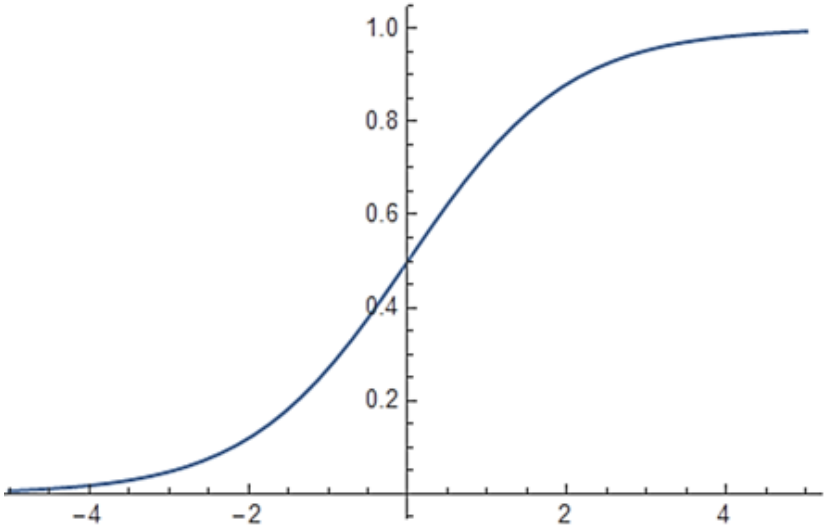

Şekil 4.2: YSA Nöron Detayı

Hesaplanan $X$ değeri daha sonra aktivasyon fonksiyonuna girdi olarak gelir. Aktivasyon fonksiyonu hücreye gelen net girdiyi işleyerek nöron hücresinde bu girdiye karşı üretilecek çıktı değerini belirler. Aktivasyon fonksiyonu olmayan bir nöron hücresi doğrusal bir yapı gösterir. Nöron hücresinin karmaşık promlemlerin doğrusal olmayan yapılarını öğrenme kabiliyeti aktivasyon fonksiyonları tarafından sağlanmaktadır. YSA modeli için problemin yapısına göre tercih edilebilecek bir çok aktivasyon fonksiyonu bulunmaktadır. Doğrusal olmayan aktivasyon seçimleri modelin çeşitli verilerle genelleştirmesini veya uyarlanmasını kolaylaştırmaktadır. Yapılan testlerde sunulan vekil model için yapısal problemlerin karmaşık doğrusal olmayan yapılarını yakalama konusunda daha iyi performans sergilediği düşünülen 'Swish' aktivasyon fonksiyonu, öte yandan karşılaştırma yapılabilmesi adına 'ReLU' ve bunların ara katmanlarda uygulanacak şekilde kombinasyonları seçilmiştir.

Örnek problem için Aktivasyon Fonksiyonu olarak Şekil 4.3'de grafiği sunulan "Sigmoid" fonksiyonu tercih edilmiştir. Grafik incelendiğinde Sigmoid fonksiyonunun toplam girdilere göre $0$ ile $1$ arasında çıktı değerleri ürettiği görülmektedir. Özellikle çıktı olarak olasılıksal tahminlerin gerektiği modeller için girdi değerlerini 0 ile 1 arasındaki olasılık değerlerine indirgemektedir.

$$\begin{aligned} Y &= Sigmoid\,(X) \\ &= \frac{1}{1+e^{-X}} \end{aligned} \tag{4.3}$$

Sigmoid Fonksiyonu, doğrusal bir fonksiyon olmaması dolayısıyla; nöronun, problemin doğrusal olmayan yapısını yakalamakta başarılıdır. Örnek problemin çözüm adımlarının




Şekil 4.3: Sigmoid Aktivasyon Fonksiyonu Grafiği

sonraki aşamalarında, gerçek çıktı ve tahminler arası hatanın minimize edilmesi için bu fonksiyonun eğiminden yararlanılacaktır. Bu sebeple türevi kolay hesaplanabilir olan bu aktivasyon fonksiyonu tercih edilmiştir. Böylece YSA'nın matematiksel modelinin daha anlaşılabilir olacağı düşünülmektedir. Sonraki aşamada nöronda elde edilen çıktı değeri ($Y$), diğer nörona iletilmek üzere nöron hücresinden çıkmaktadır.

### 4.2.3 Veri Ön İşleme Örneği

Şekil 4.3'te Sigmoid fonksiyonunun grafiği incelendiğinde, girdi değerleri pozitif yönde büyüdükçe ve negatif yönde küçüldükçe Sigmoid fonksiyonun eğimi sıfıra yaklaşmaktadır. Aktivasyon fonksiyonun türevi eğimi vermektedir. Sonraki adımlarda açıklanacağı üzere, eğimin sıfıra yakın olması YSA eğitiminde nöronun eğitim kabiliyetini düşürmektedir. Diğer taraftan Sigmoid fonksiyonu girdi değerleri için 0 ile 1 aralığında değerler üretmektedir. Bu sebeple eğitim setimizdeki tüm verilerin kendi aralarında maksimum değere göre 0 ile 1 aralığına ölçeklenmesi YSA eğitimi için uygun bir yaklaşım olacaktır (Çizelge 4.2).

Çizelge 4.2: Ölçeklenmiş Eğitim Veri Seti

| **Deney No** | $x_1/12$ | $x_2/12$ | $f/12$ |
|---|---|---|---|
| **1** | 0,75 | 0,667 | 1 |
| **2** | 0.417 | 0.333 | 0.667 |
| **4** | 0.167 | 0,083 | 0,417 |

### 4.2.4 Örnek Problem Yapay Sinir Ağı Eğitimi

Ölçeklenmiş veri setindeki $X_1$ ve $X_2$ değerleri, Şekil 4.1'daki YSA mimarisine girdi olarak verilecektir. Ancak öncesinde Girdi Katmanı ile Ara Katman arasındaki nörolanların bağlandığı ağırlık değerleri $[-1, 1]$ arasından rastgele olarak 4.4'deki gibi belirlenmiştir.

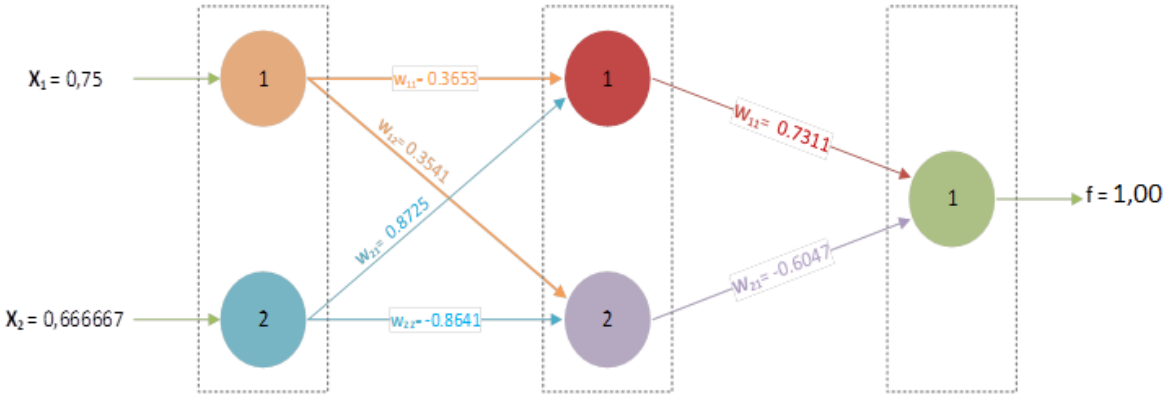
$$W_{girdi-ara} = \begin{pmatrix} W_{11} & W_{21} \\ W_{12} & W_{22} \end{pmatrix} = \begin{pmatrix} 0,3653 & 0,8725 \\ 0,3541 & -0,8641 \end{pmatrix} \tag{4.4}$$

Aynı şekilde Ara Katmanı ile Çıkış Katmanı arasındaki nörolanların bağlandığı ağırlık değerleri $[-1, 1]$ arasından rastgele olarak 4.5'deki gibi belirlenmiştir.

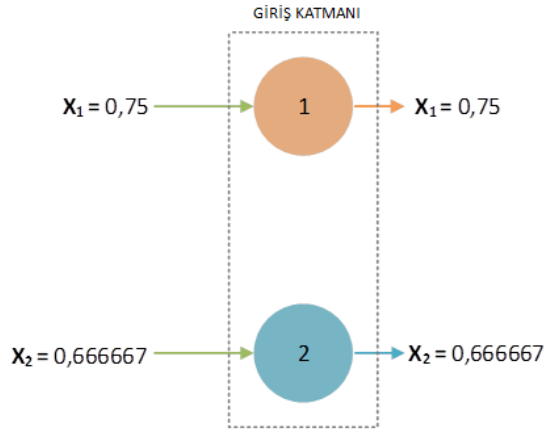
$$W_{ara-cikti} = \begin{pmatrix} W_{11} & W_{21} \end{pmatrix} = \begin{pmatrix} 0,7311 & -0,6047 \end{pmatrix} \tag{4.5}$$

Böylece örnek problemin Yapay Sinir Ağı modeli birinci aşama sonunda Şekil 4.4'de gösterildiği gibi olmuştur.

Girdi Katmanı'ndaki nöronlar aktivasyon fonksiyonu ile işleme tabi tutulmazlar. Dış dünyadan girdi katmanındaki 1 ve 2 numaralı nöronlara gelen girdi değerleri giriş katmanındaki nöronlarda çıktı olarak alınırlar. Giriş katmanında bu süreçten sonra YSA şeması Şekil 4.5'de gösterildiği gibi olmaktadır.




Şekil 4.4: Birinci Aşama YSA modeli




Şekil 4.5: Birinci Aşama YSA Giriş Katmanı

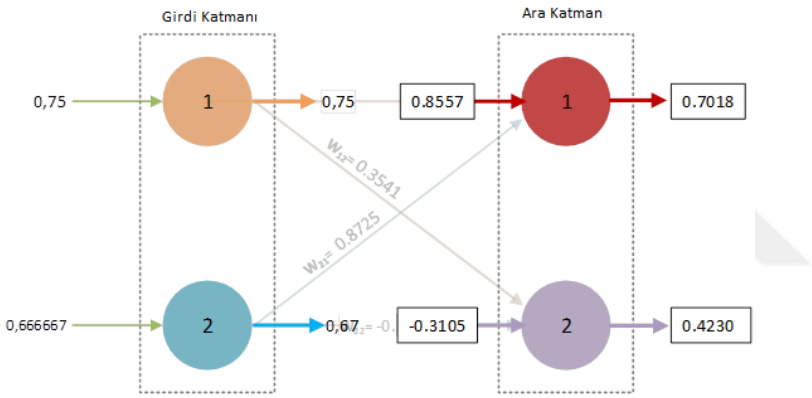
Ara katmandaki 1 numaralı ve 2 numaralı nörona gelen girdi değerleri 4.6'de matris sistemi kullanılarak hesaplanır.

$$
\begin{aligned}
X_{ara} &= \begin{pmatrix} W_{11} & W_{21} \\ W_{12} & W_{22} \end{pmatrix} \quad \begin{pmatrix} X^1_{girdi} \\ X^2_{girdi} \end{pmatrix} \\
&= \begin{pmatrix} 0,3653 & 0,8725 \\ 0,3541 & -0,8641 \end{pmatrix} \quad \begin{pmatrix} 0,75 \\ 0,667 \end{pmatrix} \qquad (4.6) \\
&= \begin{pmatrix} 0,8557 \\ -0,3105 \end{pmatrix}
\end{aligned}
$$

Sonrasında ara katmandaki 1 ve 2 numaralı nöronlara iletilen değerler Sigmoid aktivasyon fonksiyonundan geçirildikten sonra 4.7'deki matris elde edilir.

$$O_{ara} = Sigmoid(X_{ara}) = \begin{pmatrix} 0,7018 \\ 0,4230 \end{pmatrix} \tag{4.7}$$

Giriş katmanında ve ara katmanda bu süreçten sonra YSA şeması Şekil 4.5'de gösterildiği gibi olmaktadır.




Şekil 4.6: Birinci Aşama YSA Girdi-Ara Katman Detayı

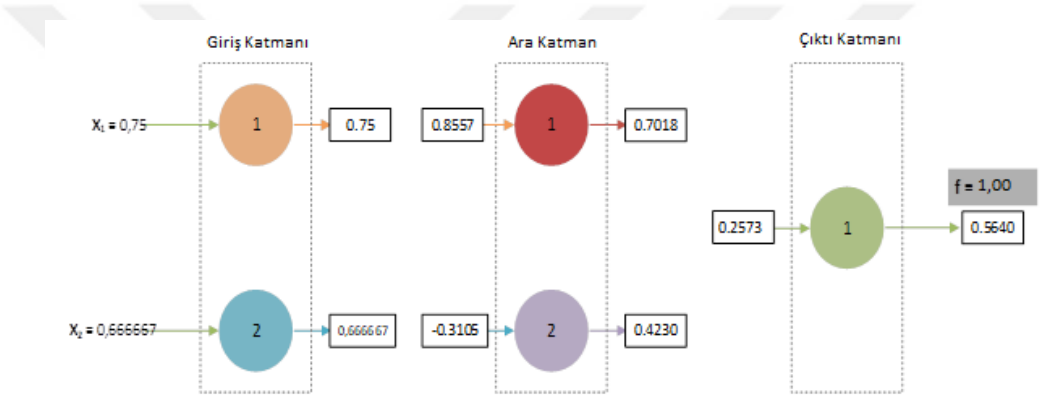
Çıktı katmanındaki 1 numaralı nörona gelen girdi değeri aşağıdaki matris sistemi kullanılarak hesaplanır.

$$\begin{aligned} X_{cikti} &= \begin{pmatrix} W_{11}^{cikti} & W_{21}^{cikti} \end{pmatrix} \quad \begin{pmatrix} X_{ara}^{1} \\ X_{ara}^{2} \end{pmatrix} \\ &= \begin{pmatrix} 0,7311 & -0,6047 \end{pmatrix} \quad \begin{pmatrix} 0,7018 \\ 0,4230 \end{pmatrix} \\ &= 0,2573 \end{aligned} \tag{4.8}$$

Yine çıktı katmanındaki 1 numaralı nörona giren değer Sigmoid aktivasyon fonksiyonundan geçirildikten sonra aşağıdaki matris elde edilir.

$$O_{cikti} = 0,564 \tag{4.9}$$

Tüm ileri besleme adımlarından sonra YSA şeması Şekil 4.7'de gösterildiği gibi olmaktadır.




Şekil 4.7: Birinci Aşama YSA Girdi-Ara Katman Detayı

### 4.2.5 Geri Yayılım Algoritması ile Hata Dağılımı

Yukarıda her bir katmandaki girdileri ve çıktıları hesaplayarak en son çıktı katmanındaki nörondan tahmin değeri olan; 0.5640 olarak hesaplanmıştır. Veri setinde 1. deney sonucunda çıktının; f = 1,0 olması beklenir. Gerçek değer ile hesaplanan tahmin değeri arasındaki fark hatayı vermektedir. Bu hatayı YSA'da Geri Yayılım Algoritması ile her bir katmandaki nöron elemanın bağlantı ağırlıkları oranında hataya nasıl etki ettikleri hesaplanacaktır. Öncelikle gerçek değer ile YSA tahminin arasındaki fark ile 4.10 ile hata değeri hesaplanır

$$\begin{aligned} Hata_{cikis} &= f - O_{cikis} = 1,0 - 0,564 \\ &= 0,436 \end{aligned} \tag{4.10}$$

Sonrasında geri yayılım yaklaşımı ile çıktı değerindeki hata ara katmandaki nöronlara 4.11'de gösterildiği gibi aktarılır.

$$\begin{aligned} Hata_{ara-cikti} &= \begin{pmatrix} W_{11} & W_{21} \end{pmatrix}^T Hata_{cikis} \\ &= \begin{pmatrix} 0,7311 & -0,6047 \end{pmatrix}^T \begin{pmatrix} 0,436 \end{pmatrix} \\ &= \begin{pmatrix} 0,319 \\ -0,264 \end{pmatrix} \end{aligned} \tag{4.11}$$

Sonrasında geri yayılım yaklaşımı ile ara katman nöronlarındaki hatalar girdi katmandaki nöronlara 4.11'de gösterildiği gibi aktarılır.

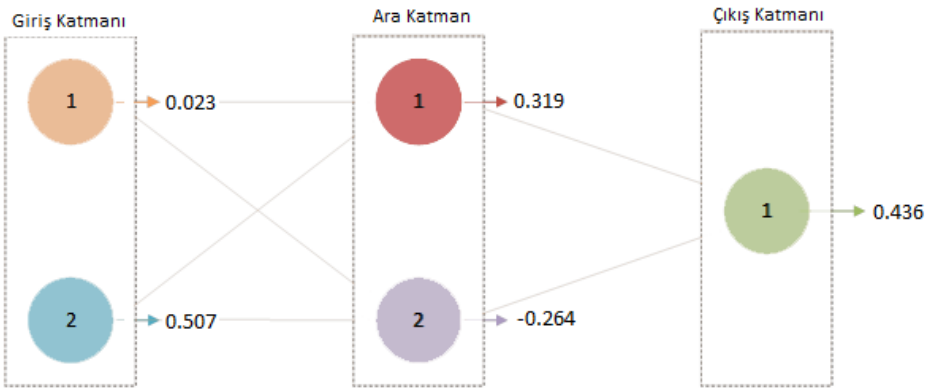
$$\begin{aligned} Hata_{giris-ara} &= \begin{pmatrix} W_{11} & W_{21} \\ W_{12} & W_{22} \end{pmatrix}^T Hata_{ara} \\ &= \begin{pmatrix} 0,365 & 0.873 \\ 0,354 & -0,864 \end{pmatrix}^T \begin{pmatrix} 0,319 \\ -0,264 \end{pmatrix} \\ &= \begin{pmatrix} 0,023 \\ 0,507 \end{pmatrix} \end{aligned} \tag{4.12}$$

Son durumda YSA şemasında her 1. deney girdi ve çıktı değerlerinin eğitime katılmasıyla rastgele olarak verilen ağırlık değerlerinden hangi ağırlık değerlerinin ne kadar hataya etki ettikleri hesaplanmıştır (Şekil 4.8).

### 4.2.6 Geri Yayılım Algoritması ile Hata Optimizasyonu

Örnek problemde veri setindeki yalnız 1. deney verisi ile işlem adımları gösterilmiştir. Diğer deney verileri için aynı işlem adımları uygulanmış ve çeşitli YSA hataları Çizelge 4.3'te paylaşılmıştır.

Çizelge 4.3'te görüldüğü gibi 3 farklı veri birleşiminin farklı farklı girdi ve çıktı değerleri hesaplanıp sonrasında hata değerleri bulunduğunda çizelgedeki ilgili sonuçlara




Şekil 4.8: YSA Geri Yayılım Hata Şeması

Çizelge 4.3: Örnek Problem YSA Hataları

| $X_1$ | $X_2$ | $f$ | $O_{cikis}$ | $f - O_{cikis}$ | $\lvert f - O_{cikis} \rvert$ | $(f - O_{cikis})^2$ |
|---|---|---|---|---|---|---|
| 0,750 | 0,667 | 1,000 | 0,564 | 0,436 | 0,436 | 0,190 |
| 0,417 | 0,333 | 0,667 | 0,541 | 0,126 | 0,126 | 0,015 |
| 0,167 | 0,083 | 0,417 | 0,522 | -0,105 | 0,105 | 0,011 |
| | | | $\frac{1}{3}\sum_{i=1}^{3} Hatalar_i$ | 0,457 | 0,667 | 0,216 |

ulaşılmıştır. Direkt olarak beklenen değer ile hesaplanan değer arasındaki fark negatif de olabileceğinden dolayı kümülatif toplamları sıfır olabilir, bu da YSA eğitiminde hata yokmuş gibi bir yanılgıya sebep olur. Direkt mutlak hata değerlerinin ortalamasının alınması, işaretlerin görmezden gelinerek işlem yapılmasına ve YSA eğitim performansı konusunda yanıltıcı bir yaklaşıma sebep olur. Gradyan İniş algoritması vasıtası ile hatalar minimize edileceğinden, En iyi yaklaşım, Ortalama Karesel Hatalar'ın alınmasıdır.

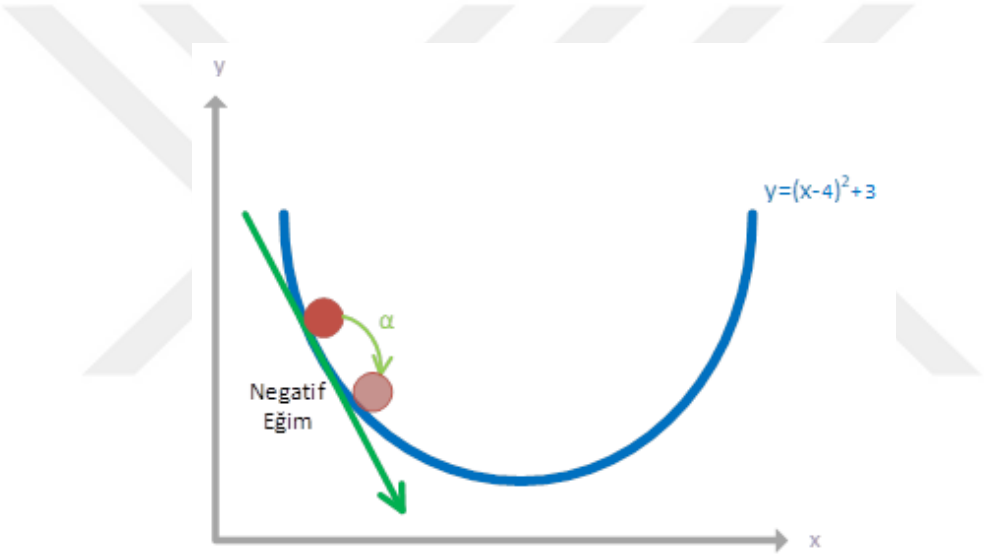
Böylece Yapay Sinir Ağı için hata değeri ($OKH$), beklenen çıktı değeri ile tahminsel çıktı değeri arasındaki farkın karesi kadar olmakta ve veri setinde bulunan veri sayısına göre ortalaması alınmaktadır (4.13).

$$OKH = \frac{1}{n}\sum_{i=1}^{n}(f_i - O_{cikti_i})^2 \tag{4.13}$$

Eşitlik incelenirse (4.13) amacın hatayı minimize etmek olduğu göz önüne alındığında $OKH$ değerinin azaltılması için eşitliğin sağ tarafındaki $f - O_k$ değerinin azaltılması gerektiği açıktır. Beklenen çıktı değeri ve hesaplanan çıktı değeri $W_{ij}$ ağırlık değerlerinin değiştirilmesi ile mümkün olmaktadır. Bunun için YSA eğitiminde sonraki bölümde açıklanacak olan, gradyan tabanlı bir optimizasyon algoritması olan Gradyan İniş Yöntemi kullanılacaktır.

#### 4.2.6.1 Gradyan İniş Kavramı

Örneğin $y = (x-4)^2 + 3$ gibi bir fonksiyon için $y$ 'nin $x$ 'e göre minimum olduğu noktayı yani yerel minimumumun bulunacağı varsayımıyla Gradyan İniş Yöntemi Şekil 4.9'da şematize edilmeye çalışılmıştır.



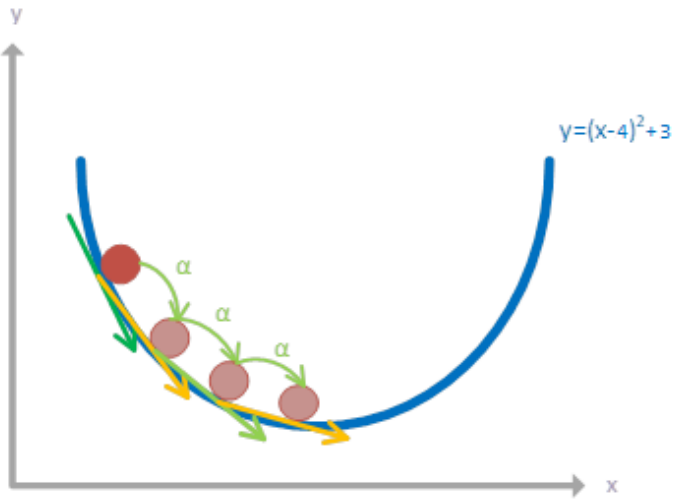

Şekil 4.9: Gradyan İniş Örneği

Bu fonksiyon basit bir fonksiyon olduğundan y 'nin minimum olduğu x noktası fonksiyonun eğiminin yani türevinin sıfır olduğu x değerinde sağlanmaktadır. Bu fonksiyonda yerel minimum ve global minimum noktalarının aynı nokta olduğu açıktır. Ancak bunun bilinmediği ve bu fonksiyonun grafiği konusunda herhangi bir fikir sahibi olunmadığı, grafiğin Şekil 4.9'daki gibi olduğu varsayılmıştır. Gradyan İniş (Dereceli Azalma), bir fonksiyonun minimumunun bulunması için birinci mertebeden yinelemeli bir optimizasyon algoritmasıdır. Eğim inişini kullanarak yerel minimum bir nokta bulmak için, geçerli noktadaki fonksiyonun gradyanının negatifiyle orantılı adımlar atılır. Bunun yerine gradyanın pozitifi ile orantılı olarak adımlar atıldığında, bu durum fonksiyonu yerel maksimuma yaklaştırmaktadır. Yani eğimin negatif olduğu yerde adımlar atıldığında minimuma yaklaşılırken, x değerlerinin arttığı görülmektedir. Şekil 4.9 sunulan grafikteki

kırmızı dairenin olduğu noktada bulunulduğu ancak grafiğin tam olarak neresinde olunduğunun bilinmediği, rastgele bir x değeri üzerinde olunduğu varsayımıyla. $\alpha$ değeri kadar adım atıldığında yeni x değeri konumu 4.14'teki gibi hesaplanmaktadır.

$$x_{yeni} = x_{eski} - \alpha \frac{\partial y}{\partial x} \tag{4.14}$$

Bu şekilde fonksiyonda, yeni x değerlerini yerine yazarak yerel minimum noktasına adım adım yaklaşılır ve fonksiyonun y değerinin artmaya başladığı son noktadan önceki yeni x değeri, bu fonksiyonun minimum olduğu x noktasını vermektedir.




Şekil 4.10: Gradyan İniş Detay

Bu yaklaşımla YSA hataları Gradyan İniş Algoritması kullanılarak kademeli olarak azaltılmaktadır (Şekil 4.10). Ağırlıklara bağlı hata fonksiyonu birçok değişken barındırmakta ve daha karmaşık bir yapıda olduğu için belki de birçok yerel minimum ve yerel maksimum noktası bulunmaktadır. Gradyan İniş Algoritması kullanılmadığı takdirde hata tek bir yerel minimumda takılır ve ondan daha küçük bir yerel minimum noktası olduğu taktirde bu nokta kaçırılarak, istenilen hata değerleri yakalanamadığı için YSA öğrenimi başarısız olmaktadır. Yapay Sinir Ağı Karesel Hatasını veren eşitlik 4.15'te sunulmuştur.

$$KH = (f - O_{cikti})^2 \tag{4.15}$$

Bu eşitliğin (4.15) türevi 4.16'da gösterilmiş olup herhangi bir $W$ ağırlık değerindeki eğimi vermektedir.

$$\frac{\partial E}{\partial W_{jk}} = \frac{\partial}{\partial W_{cikis}} \frac{1}{2} (f - O_{cikis})^2$$

$$\frac{\partial E}{\partial W_{jk}} = \frac{\partial E}{\partial O_{cikis}} \frac{\partial O_{cikis}}{\partial W_{jk}}$$

$$\frac{\partial E}{\partial W_{jk}} = -(f - O_{cikis}) \frac{\partial O_{cikis}}{\partial W_{jk}}$$

$$\frac{\partial E}{\partial W_{jk}} = -(f - O_{cikis}) \frac{\partial}{\partial W_{jk}} Sigmoid\,(\sum_j W_{jk}\ O_j) \tag{4.16}$$

$$\frac{\partial E}{\partial W_{jk}} = -\Big(f - O_{cikis}\Big)\ Sigmoid\Big(\sum_j W_{jk}\ O_j\Big) \Big(1 - \ Sigmoid\Big(\sum_j W_{jk}\ O_j\Big)\Big)\ \frac{\partial}{\partial W_{jk}}\Big(\sum_j W_{jk}\, O_j\Big)$$

$$\frac{\partial E}{\partial W_{jk}} = -\Big(f - O_{cikis}\Big)\ Sigmoid\Big(\sum_j W_{jk}\ O_j\Big) \Big(1 - \ Sigmoid\Big(\sum_j W_{jk}\ O_j\Big)\Big)\ O_j$$

Sonuç olarak hatanın ağırlığa göre türevi 4.16'daki eşitlik yardımıyla bulunur. Eşitlikteki $\alpha$ değeri öğrenme oranı olmak üzere Gradyan İniş 'e uyarlandığında 4.17'deki gibi olmaktadır.

$$\begin{aligned} W_{jk_{yeni}} &= W_{jk_{eski}} - \alpha \frac{\partial E}{\partial W_{jk}} \\ &= W_{jk_{eski}} + \ \alpha\, e_j\ Sigmoid\Big(\sum_j W_{jk}\ O_j\Big) \Big(1 - \ Sigmoid\Big(\sum_j W_{jk}\ O_j\Big)\Big)\ O_j \end{aligned} \tag{4.17}$$

4.17'deki eşitlik genel olarak formülize edildiğinde ilk adımda hatanın optimize edilmesi için ağırlığın ne kadar değiştirilmesi gerektiğini veren eşitlik 4.18 olduğu gibi formülize edilebilir.

$$\Delta W_{jk} = \Big(\alpha\, e_j\, Sigmoid(O_k)\, \Big(1 - Sigmoid(O_k)\Big)\Big).\, (O_j)^T \tag{4.18}$$

#### 4.2.6.2 Ara Katman ile Çıktı Katman Ağırlık Güncellemesi

YSA eğitimi için işlemler matris sistemiyle yapılmıştır. Gradyan İniş Algoritması ile ağırlıklara bağlı hata fonksiyonunun minimize edilebilmesi için öğrenme oranı değeri $\alpha = 0.5$ alınarak, ilk ağırlık güncellemesi yapılmış ve yeni ağırlık değerleri 4.19'a göre bulunmuştur.

$$\begin{aligned}
\Delta W_{jk} &= \Big(\alpha\, e_j\, Sigmoid(O_k)\, \Big(1 - Sigmoid(O_k)\Big)\Big).\, (O_j)^T \\
&= \Big(0,5\; 0,436\; Sigmoid(0,564)\, \Big(1 - Sigmoid(0,564)\Big)\Big).\, \begin{pmatrix} 0,035 \\ 0,024 \end{pmatrix}^T \\
&= \begin{pmatrix} 0,0087 \\ 0,0052 \end{pmatrix}
\end{aligned} \tag{4.19}$$

$$\begin{aligned}
W_{jk_{yeni}} &= W_{jk_{eski}} + \Delta W \\
&= \begin{pmatrix} 0,731 \\ -0,605 \end{pmatrix} + \begin{pmatrix} 0,035 \\ 0,024 \end{pmatrix} \\
&= \begin{pmatrix} 0,766 \\ -0,581 \end{pmatrix}
\end{aligned}$$

#### 4.2.6.3 Girdi Katmanı ile Ara Katman Ağırlık Güncellemesi

$$\Delta W_{jk} = \alpha\, e_j\, Sigmoid(O_k)\left(1 - Sigmoid(O_k)\right)\ (O_j)^T$$

$$= \left(0,5 \begin{pmatrix} 0,319 \\ -0,264 \end{pmatrix}\ Sigmoid \begin{pmatrix} 0,702 \\ 0,423 \end{pmatrix} \left(1 - Sigmoid \begin{pmatrix} 0,702 \\ 0,423 \end{pmatrix}\right)\right). \begin{pmatrix} 0,75 \\ 0,667 \end{pmatrix}^T$$

$$= \begin{pmatrix} 0.027 & 0.024 \\ -0.024 & -0.021 \end{pmatrix} \tag{4.20}$$

$$W_{jk_{yeni}} = W_{jk_{eski}} + \Delta W$$

$$= \begin{pmatrix} 0.365 & 0.872 \\ 0.354 & -0.864 \end{pmatrix} + \begin{pmatrix} 0.027 & 0.024 \\ -0.024 & -0.021 \end{pmatrix}$$

$$= \begin{pmatrix} 0.392 & 0.896 \\ 0.330 & -0.886 \end{pmatrix}$$

Örnek regresyon problemi üzerinde YSA eğitimi ilk deney sonuçlarına göre yapılmış ve ağırlıklar Gradyan İniş yöntemine göre güncellenmiştir. Rastgele seçilen ilk ağırlıklar eğitim algoritması yardımıyla optimize edilmiştir. Güncellenen ağırlık değerleriyle, yapıldığı varsayılan her bir deneyin girdi değerlerine göre YSA tahminleri yapılmış ve gerçek çıktı değerleri ile oluşturduğu Karesel Hata'lar güncellenmiş yeni hatalar elde edilmiştir. Sonuca bakıldığında hatanın yaklaşık olarak 0.002 azaldığı görülmektedir. Daha fazla verinin bulunduğu karmaşık problemlerde YSA eğitimi yapıldığında hatanın minimize edilmesi ve veri setinin tekrar tekrar iteratif bir şekilde eğitime sokulması belki yüz belki binlerce kez aynı şekilde bu işlemin tekrarını gerektirmektedir.

Yeni Ağırlık değerleri 4.21'e göre hesaplanmıştır.

$$
\begin{aligned}
X_{ara} &= \begin{pmatrix} W_{11_{yeni}} & W_{21_{yeni}} \\ W_{12_{yeni}} & W_{22_{yeni}} \end{pmatrix} \cdot \begin{pmatrix} X_{girdi^1} \\ X_{girdi^2} \end{pmatrix} \\
&= \begin{pmatrix} 0.372 & 0.878 \\ 0.348 & -0.869 \end{pmatrix} \cdot \begin{pmatrix} 0.75 \\ 0.667 \end{pmatrix} = \begin{pmatrix} 0.891 \\ -0.343 \end{pmatrix} \\
O_{ara} &= Sigmoid\big(X_{ara}\big) = \begin{pmatrix} 0,709 \\ 0,415 \end{pmatrix} \\
X_{cikti} &= \begin{pmatrix} W_{11_{yeni}} & W_{21_{yeni}} \end{pmatrix} \cdot \begin{pmatrix} O_{ara^1} \\ O_{ara^2} \end{pmatrix} \\
&= \begin{pmatrix} 0,74 & -0,599 \end{pmatrix} \cdot \begin{pmatrix} 0,704 \\ 0,421 \end{pmatrix} = 0,276 \\
O_{cikti} &= Sigmoid\big(X_{cikti}\big) = 0,566
\end{aligned}
\tag{4.21}
$$

Çıktı katmanındaki yeni hata değeri 4.22'ye göre hesaplanmıştır.

$$
\begin{aligned}
Hata_{cikti}^{yeni} &= f - O_{k_{yeni}} \\
&= 1.0 - 0.566 \\
&= 0,434 \\
\Delta Hata &= Hata_{cikti}^{yeni} - Hata_{cikti}^{eski} \\
\Delta Hata &= 0.434 - 0.436 \\
\Delta Hata &= -0.002
\end{aligned}
\tag{4.22}
$$

# 5. ÖNERİLEN VEKİL MODEL İLE OPTİMİZASYON YÖNTEMİ

Python, nesne yönelimli, yorumlamalı, modüler ve etkileşimli yüksek seviyeli bir programlama dilidir. Python'da belirli amaçlar için yazılmış birçok hazır kütüphane bulunmakta, böylece araştırmacı tasarımına odaklanarak hızlıca prototipler üretebilmektedir. Araştırma ve deneylerin verimli bir şekilde yapılabilmesi, bilimsel araştırmalarda kullanımını yaygınlaştırmıştır (Oliphant, 2007). Bu özelliklerinden dolayı sunulan vekil model ve tüm algoritma programlamaları için Python programlama dili tercih edilmiştir.

OpenSees, açık kaynak kodlu yapısal analizlerin yapıldığı bir bilgisayar yazılımıdır. Yapıların sonlu elemanlar analizi ve geoteknik deprem mühendisliği problemlerinin çözümü için kullanılmaktadır. OpenSeesPy ise bu Sonlu Elemanlar çözücüsünün Python vasıtası ile yorumlanabilmesine olanak sağlayan bir kütüphanedir ve diğer tüm kütüphaneler gibi Python'a entegre edilerek iş akışına dahil edilebilir (Zhu vd., 2018). Önerilen vekil model için sonlu elemanlar çözücüsü olarak OpenSeesPy kütüphanesi kullanılmıştır.

Tensorflow, Python'da yorumlanabilen açık kaynak kodlu bir makine öğrenmesi kütüphanesidir. İçinde çeşitli regresyon, sınıflandırma ve kümeleme algoritmalarını barındırır. Sunulan vekil model için ilgili makine öğrenmesi kütüphanesinde, Yapay Sinir Ağlarından oluşan, doğrusal olmayan bir Regresyon modeli programlanmıştır.

Detayları Bölüm 5.1'de verilen, Eşzamanlı Boyut, Şekil ve Topoloji Optimizasyonu için popülasyon tabanlı bir metasezgisel olan Jaya algoritması kullanılmıştır. Optimizasyon sürecine dahil edilecek Yapay Sinir Ağı modeli, uygulanacak vekil model stratejisi ve detayları Bölüm 5.2'de ayrıntılı olarak sunulmuştur.

## 5.1 JAYA Algoritması ile Eş Zamanlı Boyut, Şekil ve Topoloji Optimizasyonu

Bir kafes yapının optimum tasarımı sürecinde boyut, şekil ve topoloji optimizasyonu eş zamanlı olarak ele alınarak optimizasyon prosedürü Jaya algoritması ile gerçekleştirilecektir. Jaya algoritması popülasyon tabanlı bir metasezgisel olmasının yanı sıra diğer popülasyon bazlı metasezgisellerden farklı olarak çok sayıda spesifik kontrol

parametresi içermemektedir. Yalnızca "Popülasyon Boyutu" ve "İzin Verilen Analiz Sayısı" parametre olarak tanımlanır (Venkata Rao, 2016). Sonuç olarak, Jaya algoritmasında yalnızca bu iki parametrenin kullanılması optimum çözümlerin aranmasında, metasezgiselin sağlamlığını arttırmaktadır (Degertekin vd., 2019). Jaya algoritması, popülasyondaki her bir adayın en iyi çözüme yaklaşılması ve en kötü sonuçtan uzaklaştırılması prensibine dayanmaktadır.

Öncelikle Topoloji Optimizasyonu için yapısal elemanın varlığı araştırılacaktır. Boyut optimizasyonu için problemin türüne ve değişkenleri sayısına göre, var olan yapısal elemanların kesit alanlarının ($A$), şekil optimizasyonu içinse yapısal elemanların bağlandığı düğümlerin koordinat değerleri alınarak ($\xi$) tasarım değişkeni vektörü 5.1'de gösterildiği gibi seçilmiştir;

$$X = \{A_1, A_2, ...A_m, \xi_1, \xi_2, ..\xi_n\} \tag{5.1}$$

Gerçekleştirilecek optimizasyon prosedüründe basitçe genel amaç mevcut tüm kısıtların sağlanarak, kafes sistemin ağırlığının mümkün olduğunca minimize edilmesidir. Eğer yapıda bir eleman varsa, onun kesit alanından yola çıkarak her bir kesit alanının belirli kısıtları sağlayacak şekilde optimum değerlerine ulaşılır böylece sistem toplam ağırlığı mümkün olduğunca minimize edilir. Bunun için, optimum ($m$) adet kesit alanına ($A$) göre, ağırlık değeri $W(x)$ 5.2 ile hesaplanmaktadır;

$$W(x) = \sum_{i=1}^{m} k_i A_i \rho_i L_i \tag{5.2}$$

$A_i$ : Kesit Alanı

$L_i$ : Koordinat değerlerine göre kesit uzunluğu

$\rho_i$ : Özgül Ağırlık

Sunulan yaklaşımda uygun topolojinin belirlenebilmesi için yapı elemanının varlığı veya yokluğunun ağırlık değerinin hesaplanmasındaki etkisi, tasarım vektöründe kritik alan $A_{kritik} = 0.5$ seçilerek 5.3'e göre belirlenmektedir.

$$k_i = \begin{cases} 0, A_i < 0.5 \\ 1, A_i \geq 0.5 \end{cases} \tag{5.3}$$

Jaya ile Eşzamanlı Boyut şekil ve Topoloji Optimizasyonu için kafes sistemde referans alınan kısıtlar 5.4'te gösterilmiştir;

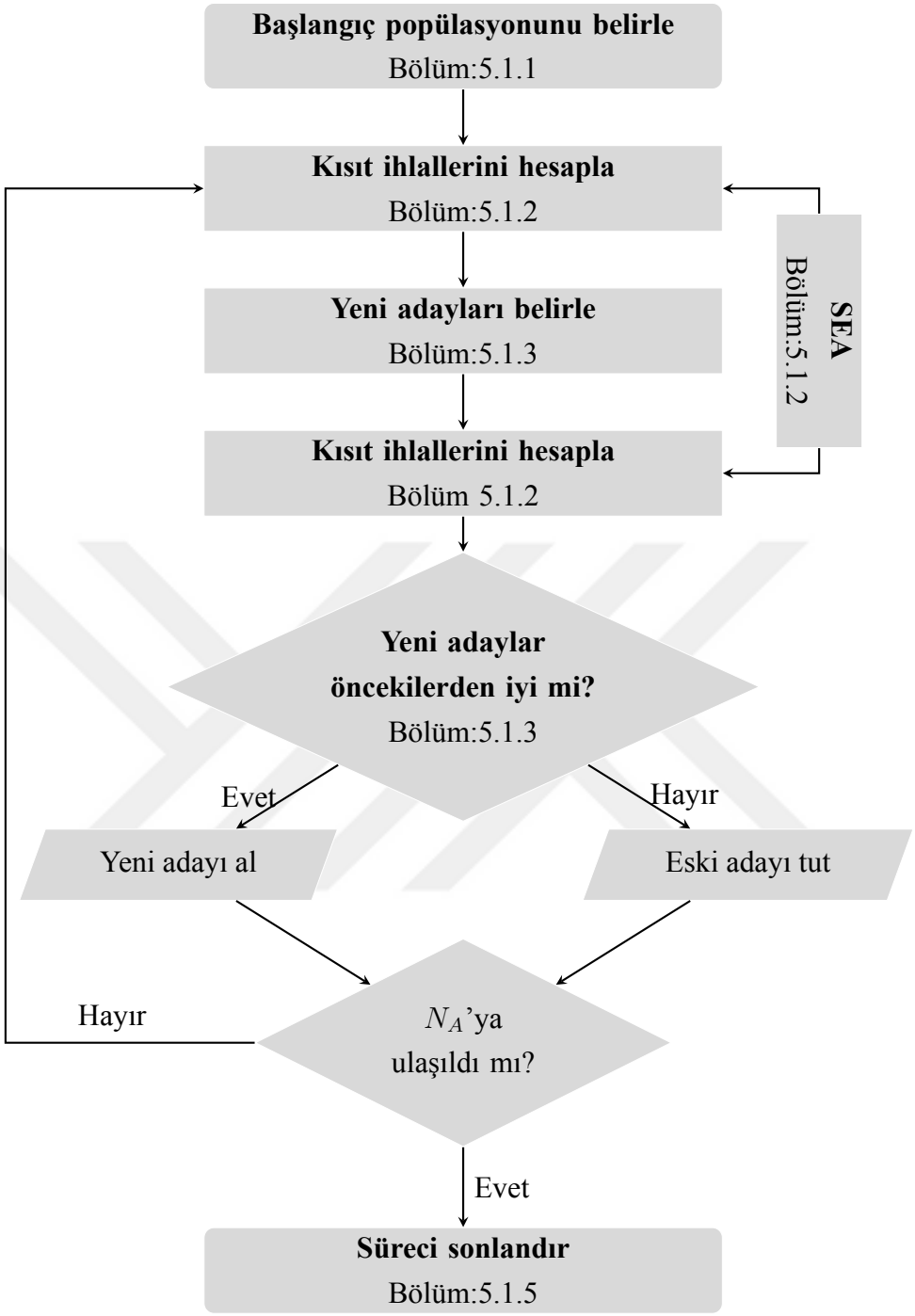
$$\begin{aligned}
&G_1 : \text{Kinematik Stabilite Kontrolü} \\
&G_2 : \text{Deplasman Kısıtları, } |\delta_j| \leq |\delta_j^{maks}| \\
&G_3 : \text{Basma-Çekme Gerilme Kısıtları, } |k_i\sigma_i| \leq |\sigma_i^{maks}| \\
&G_4 : \text{Boyut Kısıtları} \begin{cases} \text{Devamlı Değişkenler,} & A_i^{kritik} \leq A_i \leq A_i^{ust} \\ \text{Ayrık Değişkenler,} & A_i = \{a_1, a_2, ..., a_{np}\} \end{cases} \\
&G_5 : \text{Şekil Kısıtları, } \xi_j^{alt} \leq \xi_j \leq \xi_j^{ust}
\end{aligned} \tag{5.4}$$

### 5.1.1 Başlangıç Popülasyonun Belirlenmesi

JAYA algoritmasının değerlendireceği başlangıç popülasyonunun boyutu ($N_{pop}$) ve durma kriteri olarak izin verilen analiz sayısı ($N_A$) belirlenir . Oluşturulan her bir tasarım vektörü JAYA algoritmasında her bir adayı temsil etmektedir. Örneğin populasyon boyutu $N_{pop} = 5$ seçilmesi ile elde edilen populasyon 5.5'daki gibidir.

$$Pop = \begin{vmatrix} A_{11} & A_{12} & ...A_{1m} & \xi_{11} & \xi_{12} & ...\xi_{1n} \\ A_{21} & A_{22} & ...A_{2m} & \xi_{21} & \xi_{22} & ...\xi_{2n} \\ A_{31} & A_{32} & ...A_{3m} & \xi_{31} & \xi_{32} & ...\xi_{3n} \\ A_{41} & A_{42} & ...A_{4m} & \xi_{41} & \xi_{42} & ...\xi_{4n} \\ A_{51} & A_{52} & ...A_{5m} & \xi_{51} & \xi_{52} & ...\xi_{5n} \end{vmatrix} \tag{5.5}$$

Her bir aday, tasarımcı tarafından belirlenen 5.4'e göre boyut kısıtları ($G_5$) olarak kesit alanlarının ve şekil kısıtları ($G_6$) olarak düğümlerin koordinat değerlerinin alt ve üst sınırlarına göre rastgele olarak atanır. Topoloji optimizasyonu için 5.3'e göre $A^{alt} = A^{kritik} - (A^{ust} - A^{kritik})$ kesit alanı alt limit değeri hesaplanarak $\{A^{alt}, A^{ust}\}$ değerleri arasından rastgele olarak seçilir. Değişken türü sürekli ise $A^{kritik}$ değerinin üstündeki elemanlar var kabul edilerek kesit alanları olarak bu değerler alınır ve başlangıç popülasyonu belirlenir. Eğer kesit alanları ayrık değişkenlerden oluşuyorsa bu rastgele seçilim listedeki eleman sayısına göre aynı yaklaşımla alt ve üst limitlerin oluşturulmasını ve bu aralıkta rastgele seçilen sayıların en yakın tam sayıya yuvarlanmasını içerir. Pozitif tam sayılar alınarak, listede aynı sayıya karşılık gelen kesit alanı ilgili elemana atanır ve başlangıç popülasyonu belirlenir (Şekil 5.1 / 5.2).




Şekil 5.1: JAYA Optimizasyon Akış Şeması

### 5.1.2 Sonlu Elemanlar Analizi ile Kısıt İhlallerinin Hesaplanması

Genel olarak popülasyon tabanlı metasezgisellerle optimizasyon süreçlerinde kısıt ihlallerinin belirlenebilmesi süreci popülasyonun sürekli iyileştirilmesini kapsamaktadır. Bu iteratif süreçte her bir popülasyondaki her bir adayın kısıt ihlallerinin belirlenebilmesi için Sonlu Elemanlar çözücüsünde analizlerinin gerçekleştirilmesi gerekmektedir.

Optimum çözümlere yakınsamak için bu kadar çok adayın değerlendirildiği bu süreç günümüzün bilgisayarlı hesaplama teknolojilerinde bile zaman alan ve çok maliyetli bir süreçtir. Bölüm 5.2'de ayrıntılı olarak açıklanacak olan ilgili yöntem, bu bölümde yapısal analizlerin yapıldığı sonlu elemanlar çözücüsünün döndürdüğü sonuçların çoğu yerine tahminsel kestirimlerde bulunarak, Sonlu Elemanlar analizlerinin ağır hesaplama yükünü hafifletecek bir vekil olarak görev yapacaktır.

5.4'deki $G_1, G_2, G_3$ kısıtlarının kontrolü için popülasyondaki her bir aday tasarım vektörüne göre işlenen yapısal sistem sonlu elemanlar analizine gönderilir. Analizin döndürdüğü deplasman ($\delta$), gerilme ($\sigma$) vektörleri kısıt ihlalinden kaynaklı daha sonra açıklanacak olan bağlı diğer parametre değerleri, tasarım vektörünün sonuna 5.6'da eklenmiş ve yeni bir $X^{'}$ çok boyutlu listesi elde edilmiştir.

$$X^{'} = \{A_1, A_2, ...A_m, \xi_1, \xi_2, ..\xi_n, W_{pen}, W, \Delta P_\delta, \Delta P_\sigma, \delta_{ij}, \sigma_{ij}\} \tag{5.6}$$

Sonlu elemanlar analizinde öncelikle kinematik stabilite kontrolü ($G_1$) yapılan adaylar eğer stabil değiller ise Sonlu Elemanlar çözücüsü bir hata mesajı döndürür. Stabil olmadığı tespit edilen sistemler için, $X^{'}$'de tüm parametrelere 5.7'de gösterildiği üzere yüksek değerler atanır. Böylece JAYA algoritması doğası gereği bu çözümleri çok kötü çözümler olarak nitelendirdiğinden, kinematik olarak stabil olmayan bu adaylar optimizasyon sürecinde uzaklaştırılmaktadır.

$$X^{unstab} = \{A_1, A_2, ...A_m, \xi_1, \xi_2, ..\xi_n, 1e100, 1e100, 1e100, 1e100, 1e100, 1e100\} \tag{5.7}$$

Kinematik olarak stabil olan adaylar için sonlu elemanlar çözücüsü deplasman $\delta_{ij}$ ve gerilme $\sigma_{ij}$ vektörlerini döndürmektedir. Deplasman değerleri 5.4'deki $G_2$ kısıtını aştıklarında 5.8'e göre cezalandırılır.

$$\Delta P_\delta = \frac{\sum_{i=1}^{m} \frac{|\delta_i|}{\delta^{maks}} - 1}{m} \tag{5.8}$$

Gerilme değerleri 5.4'deki $G_3$ kısıtını aştıklarında 5.9'a göre cezalandırılır.

$$\Delta P_\sigma = \frac{\sum_{k=1}^{n} \frac{|\sigma_k|}{\sigma^{maks}} - 1}{n} \tag{5.9}$$

Yapısal sistemde aşılan kısıtlar daha önce 5.2'de hesaplanan sistem ağırlığı üzerinden cezalandırılmaktadır. Cezalı ağırlık 5.10 'te gösterildiği gibi hesaplanmaktadır.

$$W_{pen} = W(X)(\Delta P_{\delta} + \Delta P_{\sigma})^2 \tag{5.10}$$

Böylece başlangıçtan iyileştirilmiş her popülasyona kadar her bir aday için $X'$ çok boyutlu listesindeki tüm değerler bu bölümde sunulduğu gibi hesaplanır ve popülasyondaki tasarım vektörlerinin her biri yerine bu çok boyutlu listeler kaydedilir (Şekil 5.1 / 5.2).

### 5.1.3 JAYA ile Yeni Adayların Belirlenmesi

Bölüm 5.1.2'de 5.10'da hesaplanan cezalı ağırlık değerlerine göre JAYA algoritmasının yeni adayları belirleyebilmesi için popülasyondaki en iyi ($best$) ve en kötü ($worst$) adaylar belirlenir. Yeni adaylar önceki adaylara göre JAYA algoritmasının hareket operatörü vasıtasıyla 5.11'de gösterildiği gibi hesaplanmaktadır.

$$j_i^{t+1} = j_i^t + r_1(best_i^t - |j_i^t|) - r_2(worst_i^t - |j_i^t|) \tag{5.11}$$

$j_i$, $t$.iterasyonda, ilgili popülasyondaki aday çözümler ($j$) arasındaki, $i$.tasarım değişkenini, $best$ ve $wost$ sırası ile popülasyon içerisindeki en iyi ve en kötü adayları, son olarak $r_1$ ve $r_2$ ise $i.$ tasarım değişkeni için 0 ile 1 arasında atanmış rastgele kesirli sayısal değerleri temsil etmektedir. Hareket operatöründeki $r_1(best_i^t - |j_i^t|)$ ifadesi algoritmayı en iyi aday çözümlere yaklaştırırken, $r_2(worst_i^t - |j_i^t|)$ ifadesi ise adayları en kötü çözümlerden uzaklaştırmaktadır.

Yeni üretilen adaylar için 5.4'deki $G_4, G_5$ kısıtlarının kontrolü yapılmalıdır. Eğer bu kısıtlar aşılıyorsa deplasman ve gerilme değerlerinde alt ve üst sınır düzeltmeleri uygulanması gerekmektedir. Alt ve üst sınır kısıtlarını aşan tüm değerler alt ve üst sınırlara eşitlenmelidir. Böylece yeni adaylardan oluşan yeni bir popülasyon elde edilir. Yeni popülasyon için sonraki adım olan Bölüm 5.1.2'de belirtilen sonlu elemanlar analizi ile kısıt ihlallerinin hesaplanması adımı tekrar uygulanmalıdır (Şekil 5.1 / 5.2).

### 5.1.4 Popülasyondaki Adayların Karşılaştırılması

JAYA algoritması hareket operatörü vasıtası ile üretilen popülasyondaki her bir yeni adayın kısıt ihlalleri Bölüm 5.1.2'de 5.10'da hesaplanan cezalı ağırlık değerlerine göre önceki popülasyonun her bir adayıyla karşılaştırılır. Eğer yeni adayın cezalı ağırlık değeri önceki adaydan daha az ise bu aday daha iyi bir aday olarak önceki adayın yerini alır. Eğer yeni aday önceki adaydan daha fazla ağırlık değerine sahipse, önceki adayın ağırlık değeri

daha az olduğu için iyi bir aday olarak popülasyonda tutulmaya devam edilir. Unutulmamalıdır ki kısıt ihlalleri yapan her bir aday, kısıtı ne kadar ihmal ettiğine göre cezalandırılmış ve normal sistem ağırlığından daha yüksek bir ağırlıkmış gibi akışa dahil edilmiştir. Böylece bu adaylar yüksek ağırlık değerlerinden dolayı optimumdan uzak sonuçlar olarak JAYA algoritması tarafından optimizasyon sürecinde uzaklaştırılacaktır (Şekil 5.1 / 5.2).

### 5.1.5 Optimizasyon Sürecinin Sonlandırılması

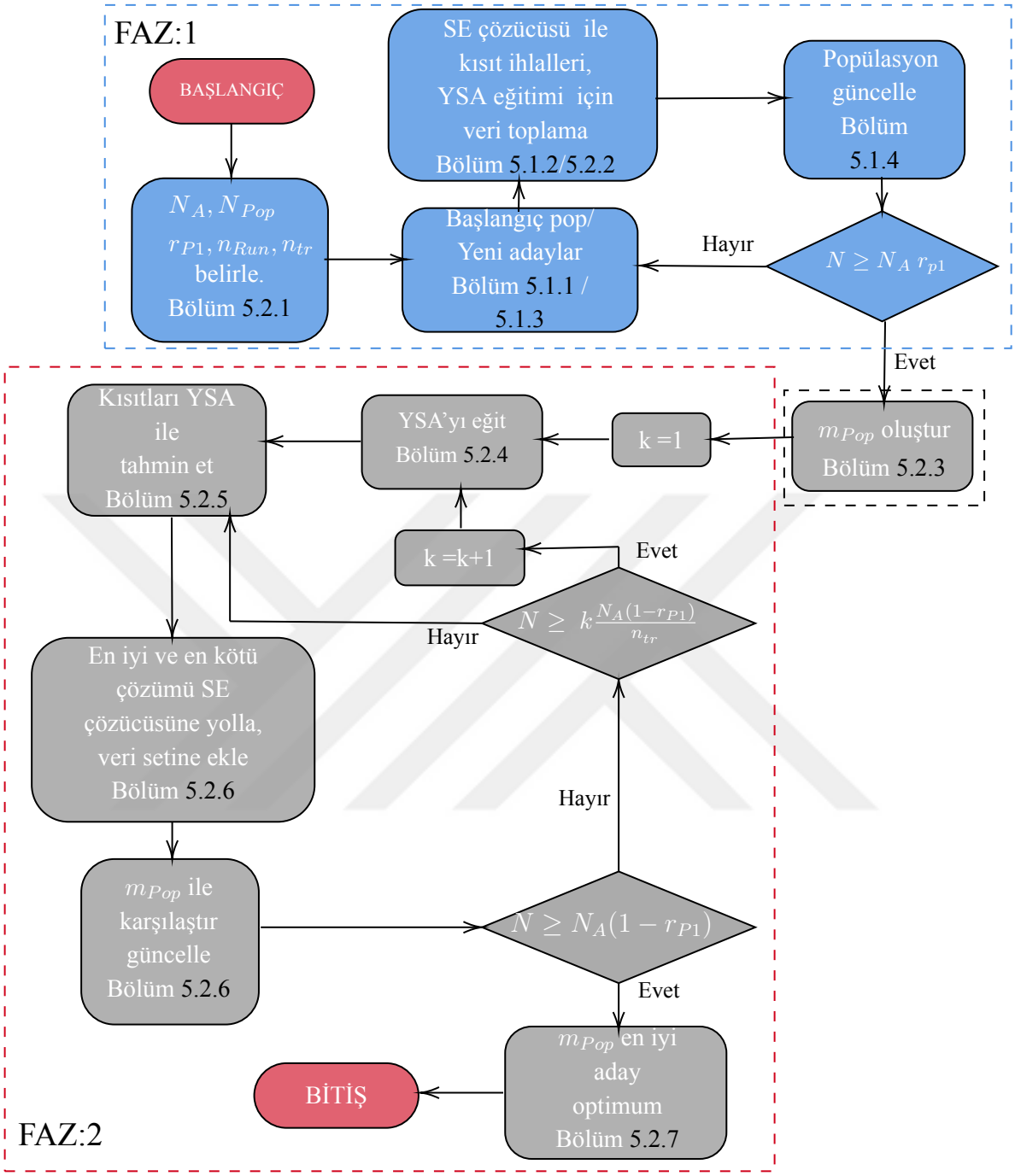
Sonlandırma kriteri İzin Verilen Analiz sayısı ($N_A$) olarak belirlenmiştir. İteratif olarak ilerleyen optimizasyon sürecinde kısıt ihlallerinin belirlendiği Bölüm 5.1.2 Sonlu Elemanlar çözücüsünün, optimizasyon süreci boyunca toplamda ne kadar çağırılacağını belirleyen bir parametredir. Her bir iterasyonda çağırılan toplam Sonlu Elemanlar çözücüsü sayısı, İzin Verilen Analiz sayısı ($N_A$) değerinden az ise son popülasyonun kısıt ihlallerinin belirlendiği adıma dönülür. Tüm süreç sonucunda çağırılan toplam sonlu elemanlar çözücüsü sayısı $N_A$ değerine ulaştığında optimizasyon süreci durdurulur ve son popülasyondaki en iyi aday optimum sonuç olarak kaydedilir (Şekil 5.1).

## 5.2 Vekil Model Olarak Derin Öğrenme Stratejisi

Bu bölümde JAYA ile eş zamanlı boyut, şekil ve topoloji optimizasyonu süreçleri içerisinde gerçekleşen her bir döngüde optimum sonuçlara yakınsamak için Bölüm 5.1.2'de bahsedilen Sonlu Elemanlar analizlerinin pahalı ve zaman alıcı hesaplama maliyetlerinin düşürülmesini amaçlayan Çok Katmanlı Yapay Sinir Ağı tabanlı bir vekil model yaklaşımı sunulmuştur. Sunulan vekil desteği ile optimizasyon yaklaşımı, iki fazlı bir süreç olarak belirlenmiştir. İlk fazda, seçilen metasezgisel algoritma ile optimizasyon süreci Bölüm 5.1.2'deki gibi gerçekleşirken, bu süreçte belirli bir analiz değerine kadar Yapay Sinir Ağının eğitimi için, girdi değerleri ve bu girdi değerlerine göre Sonlu Elemanlar çözücüsünün döndürdüğü çıktı değerleri veri olarak toplanmaktadır. İkinci fazda yine seçilen metasezgisel algoritma süreci yürütür; ancak tasarım adaylarının değerlendirilmesi yükünün çoğu Sonlu Elemanlar çözücüsü yerine Yapay Sinir Ağı tarafından üretilen tahminler doğrultusunda azaltılır. Sunulan vekil model destekli optimizasyon süreci için Şekil 5.2'de akış şeması paylaşılmış olup, metodun her bir adımı sırası ve ayrıntıları ile bu akış şemasına göre sunulmuştur.

### 5.2.1 FAZ:1 Parametrelerin Belirlenmesi

FAZ:1'in başında kullanıcı tarafından tanımlanacak parametreler Çizelge 5.1'de kısaca açıklanmıştır. Metasezgisel için popülasyonda değerlendirilecek aday sayısının




Şekil 5.2: Sunulan Vekil Model Akış Şeması

kullanıcı tarafından belirlendiği popülasyon büyüklüğü ($N_{pop}$), FAZ:1'de üretilecek verinin dağılımını etkileyeceği için önemlidir. Sunulan vekil model ile vekil olmadan yürütülen optimizasyon süreçlerinin kaç kez deneneceğini ve sonuçların karşılaştırmalı olarak tutarlılığının kontrol edilebilmesini tam sayı olarak tanımlanan $n_{Run}$ parametresi sağlamaktadır. Faz:1'de JAYA algoritması tarafından yürütülen optimizasyon süreci için toplam izin verilen analiz sayısının ($N_A$), kullanıcı tarafından belirlenen $r_{P1}$ oranına göre,

Çizelge 5.1: Vekil Model için Kullanıcı Tarafından Tanımlanacak Parametreler

| **Açıklama** | **Parametre** |
|---|---|
| Yapılacak optimizasyon denemesi sayısı | $n_{Run}$ |
| Faz:1 için ayrılacak sonlu elemanlar analizi (0 ile 1 arasında değişen oran) | $r_{P1}$ |
| Faz:2'de yapılacak Yapay Sinir Ağı eğitimi sayısı | $n_{tr}$ |
| Popülasyon boyutu | $N_{pop}$ |
| İzin verilen analiz sayısı | $N_A$ |

$r_{P1} * N_A$ kadarlık analiz kısmı vekilsiz olarak metasezgisel tarafından gerçekleştirilir. Bu süreçte YSA tabanlı vekil modelin başlangıç eğitimi için Sonlu Elemanlar analizlerinden elde edilen veriler toplanmaktadır. Faz:2'de kaç analizde bir YSA modelinin eğitiminin yapılacağını belirleyen kullanıcı tanımlı $n_{tr}$ parametresi,vekil modelin tahminlerinin genelleştirilebilmesi ve farklı girdi değerlerinin tutarlı tahminleri yani ağ performansının optimizasyon süreci boyunca arttırılması için tanımlanmıştır (Şekil 5.2).

### 5.2.2 FAZ:1 SE Çözücüsü ile Kısıt İhlallerinin Hesaplanması ve YSA Eğitimi için Veri Toplanması

Faz:1'de sunulan YSA tabanlı vekil model henüz devreye girmemiştir. JAYA algoritması ile optimizasyon süreci kullanıcı tarafından belirlenen $r_{P1}$ oranına göre, $r_{P1} * N_A$ analize kadar Şekil 5.1'deki akış şemasına göre devam etmektedir. Ancak Bölüm 5.1.2 detayları paylaşılan, Sonlu Elemanlar analizleri ile kısıtların belirlenmesi adımında optimizasyon boyunca tasarım değişkenlerinin girdi olarak verildiği ve Sonlu Elemanlar çözücüsü tarafından kısıt değerlerinin çıktı olarak döndürüldüğü değerler toplanarak bir veri seti oluşturulur.

Tasarım değişkenleri 5.1 daha önce sunulmuş olup kesit alanlarından ve probleme göre spesifik olarak belirlenmiş düğüm koordinat değerlerinden oluşmaktadır. Bu değerler daha sonra parametre ve model detayları Bölüm 5.2.4'te paylaşılacak olan Yapay Sinir Ağı'nın 5.12'de girdi değerleri olarak belirlenmiştir.

$$X^{egitim} = \{A_1, A_2, ..., \xi_1, \xi_2, ..\xi_n\} \tag{5.12}$$

YSA'nın performanslı tahminler üretmesi eğitim verilerinin yeterli fazlalıkta olması ve problemin arama uzayından seçilecek verilerin tutarlılığı ile yakından ilgilidir. Eğitim verilerinin tutarlı seçimleri için bir $limPen$ parametresi belirlenmiştir. Yalnızca bu parametrede belirlenen değerin altında ceza değerine sahip veriler eğitim veri setine eklenecektir. Sunulan mevcut metot SE analizlerinin azaltılmasına yönelik bir amaca hizmet etmesi sebebiyle veri toplanması aşamasında fazlaca SE analizi yapılması gerekmektedir. Bu analiz sayılarının artmasının önüne geçebilmek adına tasarım vektöründe kesit alanlarının ($A_i$) ayrık ve koordinat değerlerinin ($\xi_j$) sürekli değişkenler olduğu kafes sistem optimizasyon problemlerinde veri türetilmesi yolluna gidilmiştir. Örneğin 5.13'te 4 ayrık ve 2 sürekli değişkenden oluşan bir girdi katmanı kullanarak 2 adet çıktı değerlerinin tahmini için bir Yapay Sinir Ağı oluşturulmuştur.

$$\overbrace{\underbrace{0.000, 4.000, 17.000, 9.000}_{\text{Kesit Alanları}}}^{\text{Ayrık Değişkenler}} \quad \overbrace{\underbrace{728.11, 455.62}_{\text{Düğüm Koordinatları}}}^{\text{Sürekli Değişkenler}} \quad \overbrace{\underbrace{\delta_{ij} \qquad \sigma_{ij}}_{\text{Deplasmanlar - Gerilmeler}}}^{\text{Çıktılar}} \tag{5.13}$$

Ayrık değişkenlerden ilkinin bir topoloji verisi sakladığı düşünülmüştür. Bu durumda 0 ve negatif tüm değerler 0'ı temsil ettiği için 0'ın alt, orta ve üst sınırları işlenmiştir. 5.13'teki ham veri yerine, yuvarlandığında aynı ayrık değişkene götüren alt, orta ve üst sınırları kullanıp sürekli değişkenleri olduğu gibi bırakarak elde edilen 5.14'te 3 veri, eğitim verisi olarak eklenmiştir.

$$\begin{array}{lll}
\overbrace{\underbrace{-0.500, 3.500, 16.500, 8.500}_{\text{Kesit Alanları}}}^{\text{Ayrık Değişkenler}} & \overbrace{\underbrace{728.11, 455.62}_{\text{Düğüm Koordinatları}}}^{\text{Sürekli Değişkenler}} & \overbrace{\underbrace{\delta_{ij} \qquad \sigma_{ij}}_{\text{Deplasmanlar - Gerilmeler}}}^{\text{Çıktılar}} \\
\overbrace{\underbrace{0.000, 4.000, 17.000, 9.000}_{\text{Kesit Alanları}}}^{\text{Ayrık Değişkenler}} & \overbrace{\underbrace{728.11, 455.62}_{\text{Düğüm Koordinatları}}}^{\text{Sürekli Değişkenler}} & \overbrace{\underbrace{\delta_{ij} \qquad \sigma_{ij}}_{\text{Deplasmanlar - Gerilmeler}}}^{\text{Çıktılar}} \\
\overbrace{\underbrace{0.499, 4.499, 17.499, 9.499}_{\text{Kesit Alanları}}}^{\text{Ayrık Değişkenler}} & \overbrace{\underbrace{728.11, 455.62}_{\text{Düğüm Koordinatları}}}^{\text{Sürekli Değişkenler}} & \overbrace{\underbrace{\delta_{ij} \qquad \sigma_{ij}}_{\text{Deplasmanlar - Gerilmeler}}}^{\text{Çıktılar}}
\end{array} \tag{5.14}$$

Üretilen verinin rastgele %70'i ile yukarıda belirtilen şekilde veri icat edilir ve eğitim verisi listesindeki her veri icat edilen 3 veri ile değiştirilir. Böylece YSA $0.70 * 3 * N$ adet veri ile eğitilmiş olacaktır. Üretilen verinin geri kalan %30'u ile de eğitilen YSA test edilecektir.

Sonraki adımlar JAYA algoritmasının vekil model olmadan işlediği olağan süreçle aynı gerçekleşmektedir. FAZ:1'in tüm adımları ve ilgili bölümler için Sunulan Vekil Model Akış Şeması (Bknz. Şekil 5.2) incelenebilir.

### 5.2.3 FAZ:2'ye Hazırlık ve Ana Popülasyon Oluşturulması

JAYA ile optimizasyon ve veri toplama sürecinin gerçekleştiği FAZ:1, Sonlu Elemanlar analizlerinin toplamının, $r_{P1} * N_A$ analize ulaşmasıyla sona ermektedir. FAZ:2 aşamasına geçmeden önce bu optimizasyon süreci sonunda elde edilen iyileştirilmiş son popülasyon, ana popülasyon ($m_{Pop}$) olarak kaydedilir. FAZ:2 aşamasında, vekil modelin tahminleri, bu referans popülasyon ile karşılaştırılarak güncellenecektir. FAZ:2 aşamasına geçilmeden önce, FAZ:1 sonucunda elde edilmiş Sonlu Elemanlar analizlerinden toplanmış veriler ve iyileştirilmiş son popülasyon, ana popülasyon ($m_{Pop}$) olarak kaydedilmiştir (Bknz. Şekil 5.2).

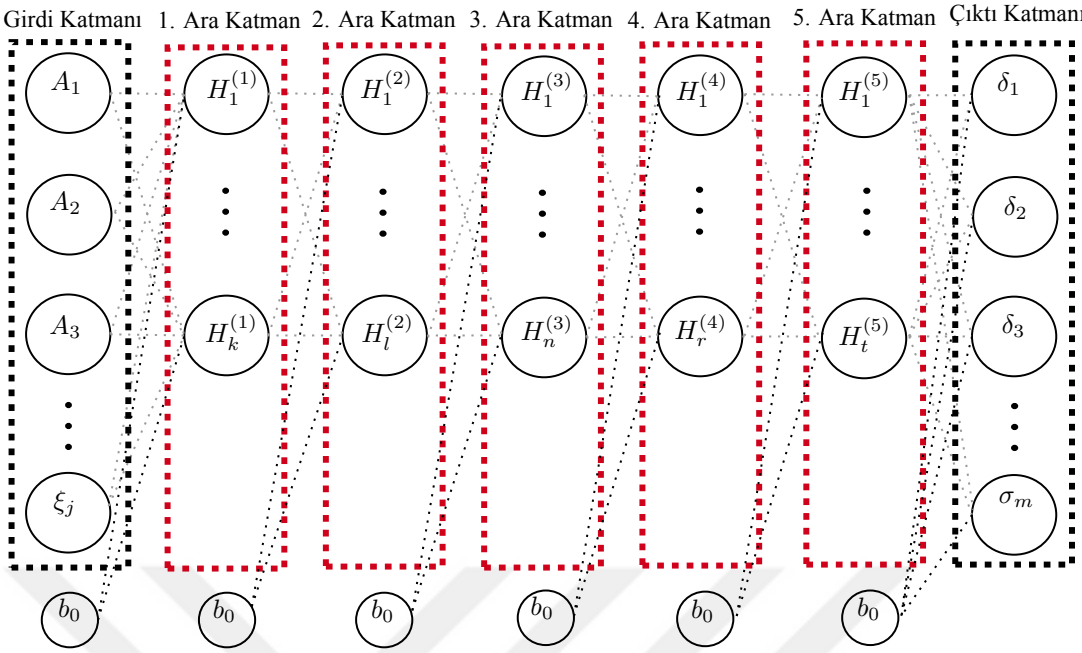
### 5.2.4 FAZ:2 Vekil Olarak Seçilen Yapay Sinir Ağı Modeli ve Eğitimi

Vekilin sonraki adımda SE analizinin döndürdüğü sonuçların tutarlı tahminlerini gerçekleştirebilmesi için sunulan YSA modeli eğitilir. Sunulan metottaki YSA model ve kullanılan parametrelerin detaylarının bu bölümde paylaşılması uygun bulunmuştur. İlgili detaylara geçmeden önce YSA ile ilgili teorik bilgi eksikliği olduğu düşünülüyorsa, öncelikle YSA'nın çalışma prensibi ve model tasarımı için genel bilgileri içeren Bölüm 4.2'nin incelenmesi önerilmektedir.

Sunulan YSA modelinde 5 adet gizli katman kullanılmıştır (Şekil 5.3). Optimizasyon problemlerinin karmaşıklığı göz önüne alındığında kullanılan 5 adet gizli katmanın bu problemlerin doğrusal olmayan yapılarını başarıyla yakaladığı tespit edilmiştir. Gizli katman numaraları giriş katmanı tarafından başlamak üzere, gizli katmanlardaki nöron sayısı 5.15'te hesaplanmıştır.

$$nH_i = 2(I + O) + \left\lfloor -\frac{i-1}{5}\,(2I + O) \right\rceil \tag{5.15}$$

$nI$ ve $nO$ giriş ve çıkış katmanlarındaki nöron sayılarıdır. $nH_i$ ise $i.$ gizli katmanın nöron sayısına işaret etmektedir. Sunulan YSA mimarisinde 5 gizli katman kullanılacağı için $i$, 1 ile 3 arasında bir sayıdır. $\lfloor\ \rceil$ operatörünün genel olarak bir sayıyı aşağı veya yukarı yuvarladığı hatırlatılmalıdır.




Şekil 5.3: Sunulan Vekil Model YSA Mimarisi

YSA mimarisinde her bir katmandaki nöronlarda aktivasyon fonksiyonları olarak 'Swish' ve 'Leaky ReLU' seçilmiştir. Leaky ReLU fonksiyonlarının negatif değer eğimi $\alpha = 0.01$ olarak alınmıştır. Aktivasyon fonksiyonu seçiminin YSA model performansı için önemli olduğu düşünülmektedir. Farklı problemler için her bir nöronda farklı kombinasyonları test edilmiş ve sonuçlar Bölüm 6.1'de paylaşılmıştır.

Üretilen veri seti beklenmeyen / aykırı (outlier) veriler içermeyeceğinden "Min-Max Scaling" veri ölçekleme yöntemi tercih edilmiş ve veriler 0 ile 1 arasında ölçeklenmiştir.

$$x_i^{norm} = \frac{x_i - x_{min}}{x_{maks} - x_{min}} \tag{5.16}$$

Burada $x_{maks}$ ve $x_{min}$ ilgili girdi sütunu için veri setinde sunulan en büyük ve en küçük değerlerdir. $x_i$ ve $x_i^{norm}$ ise ilgili girdi sütunundaki $i.$ gerçek ve normalize edilmiş (ölçeklenmiş) değerlerdir. Kullanılan aktivasyon fonksiyonları hedef (çıktı) değerlerini 0 ile 1 arasına indirgediğinden ötürü hedef (çıktı) değerleri ölçeklenmeyecektir.

Eğitimde, az bellek gereksinimi olan birinci dereceden gradyanlar gerektiren verimli stokastik hata optimizasyonu için 'Adam' algoritması kullanılır. Genel olarak, Adam'ın makine öğrenimi alanındaki çok çeşitli dışbükey olmayan optimizasyon problemlerinde

sağlam ve çok uygun olduğu görülmüştür. Adam parameterleri için tanıtıldığı makalede çoğu makine öğrenmesi uygulaması için uygun olduğu belirtilen değerler olarak $\alpha = 0.001$, $\beta_1 = 0.9$ , $\beta_2$ = 0.999 ve $\epsilon = 10^{-8}$ seçilmiştir (Kingma ve Ba, 2017).

Hata fonksiyonu olarak, Ortalama Karesel Hata (OKH) kullanılmıştır. Eşitlik 5.17'da, $n$ çıktı sayısını, $y^{target}$ hedef değeri, $y^{predicted}$ ise YSA'nın tahmin ettiği değeri gösterir.

$$OKS = \frac{1}{n}\sum_{i=1}^{n}(y_i^{target} - y_i^{predicted})^2 \tag{5.17}$$

Eğitim devam ederken $100$. kez Ortalama Karesel Hata değerinde herhangi bir iyileşme gözlenmiyorsa eğitim sonlandırılır (Bknz. Şekil 5.2).

### 5.2.5 FAZ:2 Kısıt İhlallerinin Vekil YSA ile Tahmini

Vekil model olarak sunulan YSA eğitimi sonrası eğitim veri setinin %30'u ile eğitim performansı test edilmiştir. FAZ:2 hazırlık evresinde (Bknz. Bölüm 5.2.3) kaydedilen ana popülasyon ($m_{Pop}$), FAZ:1 sonucunda kaydedilmiş son popülasyondur. Ana popülasyon ($m_{Pop}$) yeni adaylar ve önceki adayların karşılaştırılması için $Pop$ değişkenine ana popülasyonun bir kopyası olarak atanır. $Pop$'dan JAYA algoritmasının hareket operatörü vasıtası ile yeni adaylar üretilmiştir (Bknz. Bölüm 5.1.3). Yeni adayların kısıt ihlalleri hesaplanırken SE çözücüsünün döndürdüğü sonuçlar yerine, eğitilmiş vekil YSA tahminleri alınır. Böylece tüm popülasyondaki adayların hesaplama yükü, maliyetli olan SE çözücüsü yerine vekil tarafından tahminsel olarak hesaplanmaktadır. Yeni adaylardan oluşan yeni popülasyon, Bölüm 5.1.4'deki gibi $Pop$ popülasyonu ile karşılaştırılır ve $Pop$ güncellenir. Vekil YSA ile tahmin edilen, deplasman ($\delta_{ij}$) ve gerilme ($\sigma_{ij}$) değerleri, kesin değerlere yakın ancak hatalı kestirimsel sonuçlar döndürmektedir. Adayların kısıt ihlallerinin vekilin döndürdüğü hatalı tahminsel değerlerlerle belirlenmesi optimum sonuçlardan uzak ve tutarsız sonuçlara sebep olabilmektedir. Bu sebeple vekilin, SE çözücüsü yerine döndürdüğü hatalı tahminsel sonuçlar, sonraki adım olan, Bölüm 5.2.6 sunulan metotla değerlendirilmektedir.

Vekil model performasının karşılaştırmalı testlerinin yapılabilmesi ve vekil model tahminleri ile SE analizlerinin kesin sonuçları arasındaki hataların istatistiksel olarak değerlendirilebilmesi için popülasyonun ($Pop$) tamamı süreci kontrol etmek amacıyla sonlu elemanlar çözücüsü tarafından değerlendirilerek $contPop$ değişkenine atanır (Bknz. Şekil 5.2).

### 5.2.6 FAZ:2 Vekil Model Tahminlerinin SE ile Değerlendirilmesi Yaklaşımı

Popülasyondaki ($Pop$) en iyi ve en kötü adaylar ve popülasyonda kaçıncı sıradaki adaylara karşılık geldikleri (indeks numaraları) kaydedilir. En iyi ve en kötü aday SE çözücüsüne yollanarak, bu iki adayın kısıt ihlalleri hesaplanır. Artık popülasyonda ($Pop$), YSA'nın en iyi ve en kötü olduğunu iddia ettiği adayların gerçek değerlendirme verileri bulunmaktadır. Örneğin en iyi ve en kötü adayların sırasıyla popülasyondaki 3. ve 18. adaylar oldukları varsayımından yola çıkarak, popülasyondaki ($Pop$), 3. aday ana popülasyondaki ($m_{Pop}$) 3. aday ile, benzer şekilde popülasyondaki ($Pop$) 18. aday, ana popülasyondaki ($m_{Pop}$) 18. aday ile karşılaştırılarak ana popülasyon güncellenir. Popülasyondaki 3. ve 18. adaylar 5.14'de sunulan yaklaşımla veri üretilerek YSA'nın tekrar eğitilerek, genelleme yeteneğinin ve performansının arttırılması için eğitim veri setine eklenir (Şekil 5.2).

### 5.2.7 FAZ:2 Vekil Model ile Optimizasyon Sürecinin Sonlandırılması

Optimizasyon süreci boyunca her bir iterasyonda toplam SE tarafından gerçekleştirilen analiz sayısı ($N$), Çizelge 5.1'de paylaşılmış olan kullanıcı parametrelerine göre $N_A\ (1 - r_{P1})$ analiz sayısına ulaşmadığı taktirde optimizasyon süreci devam etmektedir. Kullanıcı tarafından belirlenen, $n_{tr}$ parametresine göre kaç analizde bir YSA eğitimi yapılacağının kontrolü 5.18'e göre yapılır. Eğer analiz sayısı ($N$) bu değerden büyük veya eşitse YSA eğitilmekte ve $k$ değeri her bir iterasyonda bir arttırılmaktadır, küçükse Bölüm 5.2.5'te açıklanan adıma dönülmektedir.

$$N_{kontrol} = k\ \frac{N_A\ (1 - r_{P1})}{n_{tr}} \tag{5.18}$$

SE tarafından gerçekleştirilen analiz sayısı ($N$), $N_A\ (1 - r_{P1})$ analiz sayısına ulaşır veya bu analiz sayısı aşılırsa ana popülasyondaki ($m_{Pop}$) en iyi aday optimum sonuç olarak sunulur (Bknz. Şekil 5.2).

# 6. BULGULAR VE TARTIŞMA

## 6.1 Vekil Model ile Eşzamanlı Boyut, Şekil ve Topoloji Optimizasyonu Deneyleri

### 6.1.1 Test Problemleri ve Vekil Modelsiz Sonuçlar

Deneylerin yapılacağı (sırasıyla 10, 15, 39 çubuk), popüler üç kafes sistem problemi tanıtılmış (Bknz. Bölüm 6.2.1, 6.3.1, 6.4.1) ve sunulan yaklaşımın gerçekçi bir karşılaştırması için, bu problemler üzerinde metasezgisel olarak seçilen Jaya algoritması vasıtası ile öncelikle vekil modelin devrede olmadığı eş zamanlı boyut, şekil ve topoloji optimizasyonu gerçekleştirilmiştir. Ayrıca vekil modelin devrede olmadığı optimizasyon sonuçlarının tutarlılığının gözlenebilmesi adına literatürde aynı problemler üzerinde yapılmış çalışmaların referans optimum sonuçları ile karşılaştırılmıştır (Bknz. Bölüm 6.2.1.1,6.3.1.1, 6.4.1.1).

### 6.1.2 YSA Eğitim Deneyleri

Vekil olarak düşünülen YSA eğitiminin performansına etki edebileceği düşünülen çeşitli parametrelerle her bir problem üzerinde bağımsız deneyler gerçekleştirilmiştir (Bknz. Bölüm 6.2.1.2, 6.3.1.2, 6.4.1.2). Sonrasında gerçekleştirilecek vekil model yaklaşımı deneyleri için YSA eğitim parametreleri bu deneylerde elde edilen bilgiler ışığında seçilmiştir. Deney sonuçlarından elde edilen grafik detayları aşağıda listelenmiştir.

**İzin Verilen Analiz Sayısı - Veri Toplanma Yüzdesi:** Vekil olarak belirlenen YSA'nın test problemleri için izin verilen analiz sayılarına göre SE analizlerinin döndürdüğü adaylardan belirli bir ceza değerinin altındaki adaylar eğitim veri setine dahil edilmiş, Kinematik olarak stabil olmayan ve belirli bir ceza değerinin üstünde değere sahip olan adayların, eğitim veri setinde gürültüye ve eğitim uzayının genişlemesiyle vekil YSA tahminlerinde büyük sapmalara sebep olduğu gözlenmiştir. Bu sebeple bu adaylar, vekil YSA'nın eğitim setine dahil edilmemiş ve bu sebeple bu faktörün etkilerinin testleri için Limit Ceza ($limPen$) parametresi belirlenmiştir. Seçilen her bir Limit Ceza ($limPen$) parametresiyle (1.0,0.5,0.1) İzin verilen analiz sayısına göre SE analizlerinden toplanan veri yüzdesi grafikleri sunulmuştur (Bknz. Şekil 6.5, 6.15, 6.25).

**İzin Verilen Analiz Sayısı - İterasyon:** Test problemlerinde izin verilen analiz sayılarına göre toplanan veri ile vekil YSA'nın eğitimi gerçekleştirilmiştir. Eğitim her bir ara

katmanda uygulanan aktivasyon fonksiyonlarının eğitim performansına etkisini değerlendirmek üzere, bir kombinasyonu olarak test edilmiştir. Kullanılan aktivasyon fonksiyonları Çizelge 6.1'de sunulmuştur. Eğitim belirli bir iterasyon değerine ulaşır ve Ortalama Karesel Hata, Adam algoritması ile optimize edilir. Belirli bir başarı yüzdesi yakalandığında ve artık bir gelişme gözlenmediğinde YSA eğitimi otomatik olarak sonlandırılmaktadır. Burada vekil YSA eğitiminde bahsedilen iterasyon teriminin İngilizce 'Epoch' terimine karşılık geldiği hatırlatılmalıdır. Seçilen her bir Limit Ceza ($limPen$) parametresiyle (1.0, 0.5, 0.1) izin verilen analiz sayısına göre iterasyon değerleri grafikleri sunulmuştur. (Bknz. Şekil 6.6, 6.16, 6.26).

**İzin Verilen Analiz Sayısı - Eğitim OKH:** Seçilen limPen parametrelerine bağlı olarak İzin verilen analiz sayılarına göre SE analizlerinden rastgele toplanan verilerin %70'i ile YSA eğitimi yapılmış ve eğitimin sona erdiği iterasyon sayılarında YSA eğitim performansının, ölçülebilmesi adına Ortalama Karesel Hatalar kaydedilmiştir. Seçilen her bir Limit Ceza ($limPen$) parametresiyle (1.0,0.5,0.1) İzin verilen analiz sayısına göre Eğitim OKH değerleri grafikleri sunulmuştur (Bknz. Şekil 6.7, 6.17, 6.27).

**İzin Verilen Analiz Sayısı - Test OKH:** Seçilen $limPen$ parametrelerine bağlı olarak İzin verilen analiz sayılarına göre YSA'nın daha önce karşılaşmadığı SE analizlerinden rastgele toplanan verilerin %30'u test verisi olarak alınmıştır. Bu veriler eğitilmiş YSA'ya girdi olarak verilmiş ve gerçek değerleri ile test verileri için Ortalama Karesel Hata'ları kaydedilmiştir. Böylece daha önce görmediği verilerle YSA'nın performans değerlendirmesi yapılmıştır. Seçilen her bir Limit Ceza ($limPen$) parametresiyle (1.0, 0.5, 0.1) İzin verilen analiz sayısına göre Test OKH değerleri grafikleri sunulmuştur (Bknz. Şekil 6.8, 6.18, 6.28).

**İzin Verilen Analiz Sayısı - Test OKH/Eğitim OKH:** Seçilen limPen parametrelerine bağlı olarak İzin verilen analiz sayılarına göre kaydedilen Test Ortalama Karesel Hata'ları ile Eğitim Ortalama Karesel Hata'ları oranlanmış ve bir performans tutarlılık oranı ölçülmüştür. Tek başına eğitim veya test OKH değerleri YSA'nın performansı hakkında fikir vermeyebilir. YSA modelinde fazla uyum ya da yetersiz uyum olabilmektedir. Objektif bir performans değeri için bu değerin 0,5 ile 1,0 arasında olması YSA'nın vekil olarak başarılı kabul edilebileceğini göstermektedir. Seçilen her bir Limit Ceza ($limPen$) parametresiyle (1.0, 0.5, 0.1) İzin verilen analiz sayısına göre Test OKH/Eğitim OKH oranları grafikleri sunulmuştur (Bknz. Şekil 6.9, 6.19, 6.29).

Çizelge 6.1: On Çubuklu Kafes:Vekil YSA Aktivasyon Fonksiyonları (Özbaşaran, 2021)

| | **1.ARA KATMAN** | **2.ARA KATMAN** | **3.ARA KATMAN** | **4.ARA KATMAN** | **5.ARA KATMAN** |
|---|---|---|---|---|---|
| ReLU | Leaky ReLU | Leaky ReLU | Leaky ReLU | Leaky ReLU | Leaky ReLU |
| Swish | Swish | Swish | Swish | Swish | Swish |
| Swish-ReLU | Swish | Leaky ReLU | Swish | Leaky ReLU | Swish |
| ReLU-Swish | Leaky ReLU | Swish | Leaky ReLU | Swish | Leaky ReLU |

### 6.1.3 Vekil Model Performans Sonuçları

sunulan Yapay Sinir Ağı tabanlı vekil model tekniği ile üç kafes sistem problemi üzerinde Eş zamanlı Boyut, Şekil ve Topoloji optimizasyonu deneyleri yapılmıştır. İzin verilen analiz sayısı 2500 olarak sınırlandırılmış ve her bir optimizasyon süreci bağımsız olarak 20 kez tekrarlanaarak bulunan en iyi, en kötü ve ortalama amaç değerleri için test edilmiştir. Popülasyon sayısı ($N_{pop}$), 25 olarak belirlenmiştir. Her bir ara katmanda 'Swish' aktivasyon fonksiyonu tercih edilmiştir. Her bir test probleminde vekil modelin ikamet etmediği sonuçlar ile sunulan vekil model tekniği istatistiki olarak karşılaştırılarak sonuçlar paylaşılmıştır (Bknz. Bölüm 6.2.1.3, 6.3.1.3, 6.4.1.3).

**Vekil Model Performansı:** Sunulan vekil model ile optimizasyon süreçlerinin, FAZ:1 için belirlenen $rP1$ parametrelerine (Bknz. Bölüm 5.2) ve sunulan vekil model için belirlenen toplam izin verilen analiz sayılarına ($N_A$) göre vekil modelin devrede olmadığı Jaya algoritması ile belirlenen optimum ağırlık değerleri karşılaştırılarak bağıl ağırlık değerleri ve bağıl varyasyon katsayıları paylaşılmıştır. Toplam izin verilen analiz sayısına göre sunulan vekil model ile gerçekleştirilen optimizasyon süreci sonucunda elde edilen son popülasyonda gözlenen en iyi ağırlık değerinin, aynı izin verilen analiz sayısına denk gelecek şekilde vekilsiz olarak gerçekleştirilen optmizasyon sürecinin sonucuna oranlanmasıyla en iyi bağıl amaç değeri bulunmuştur (6.1).

$$N_{W_{best}} = \frac{W_{sur}^{best}}{W^{best}} \tag{6.1}$$

Bu oranın 1'den küçük olması vekil modelin son popülasyonda bulunan en iyi adayın performansının vekilin devrede olmadığı modelden daha iyi olduğunu, büyük olması ise vekil modelin devrede olmadığı olağan optimizasyon sürecinin daha verimli olduğunu göstermektedir.

Toplam izin verilen analiz sayısına göre sunulan vekil model ile gerçekleştirilen optimizasyon süreci sonucunda elde edilen son popülasyonda gözlenen en kötü ağırlık değerinin, vekil model devrede olmadan olarak gerçekleştirilen optmizasyon sürecinin sonucuna oranlanmasıyla en kötü bağıl amaç değeri bulunmuştur (6.2).

$$N_{W_{worst}} = \frac{W_{sur}^{worst}}{W^{worst}} \tag{6.2}$$

Bu oranın 1'den küçük olması vekil modelin son popülasyondaki en kötü adayın amaç fonksiyonun vekilin devrede olmadığı metasezgiselden daha küçük olduğunu, büyük olması ise vekil modelin, vekil desteksiz metasezgiselden kullanımının daha daha verimsiz olduğunu göstermektedir.

$$\mu_W = \frac{\sum_{i=1}^{N_{pop}} \frac{W_{sur}^i}{W_0^i}}{N_{pop}} \tag{6.3}$$

Vekil model ile gerçekleştirilen optimizasyon sonucunda elde edilen son popülasyondaki tüm adayların amaç değerlerinin ortalamasının, vekilin devrede olmadığı Jaya optimizasyonu sonucunda bulunan amaç değerlerine oranı ortalama bağıl amaç değerini vermektedir. Böylece vekil modelle bulunan popülasyonun grafikteki merkezi eğilimi bulunabilir (6.3).

Bağıl Varyasyon katsayısı, karşılaştırılan iki değerin olasılık dağılımı için bir normalize edilmiş istatistiksel yayılma ölçülerinin oranı olarak alınmıştır. Burada göreli bir karşılaştırma için vekil modelin devrede olduğu amaç değerine göre varyasyon katsayısının, vekil modelin devrede olmadığı optimizasyon süreçlerindeki amaç değerlerine göre varyasyon katsayısına oranı olarak alnmıştır. Her iki optimizasyon süreçlerinde bulunan amaç fonksiyonu değerinin ortalamaya göre farklılıklarının derecesini ve ortalamaya göre ne kadar değiştiğini ölçmek amacıyla sunulmuştur (6.4).

$$\frac{c_v^{vekil}}{c_v^0} = \frac{\sigma_{W_{vekil}} \ \ \mu_{W_0}}{\sigma_{W_0} \ \ \mu_{Wvekil}} \tag{6.4}$$

Vekil modelle desteği ile yapılan 20 optimizasyon denemesi için ağırlık değerlerinin

Çizelge 6.2: On Çubuklu Kafes Sistem Tasarım Parametreleri

| **Tasarım Parametreleri** | **Değer** |
|---|---|
| Elastisite Modülü | $10^4\ ksi$ |
| Özgül Ağırlık | $0.1\ lb/in^3$ |
| İzin Verilen Basınç Gerilmesi | $25\ ksi$ |
| İzin Verilen Çekme Gerilmesi | $25\ ksi$ |
| İzin Verilen Dikey Yerdeğiştirme | $2\ in$ |

standart sapması bulunmuş ($\sigma_{W_{vekil}}$) ve bulunan ağırlık değerlerinin ortalamasına ($\mu W_{vekil}$) oranlanarak varyasyon katsayısı bulunmuştur ($c_v^{vekil}$). Aynı şekilde vekil modelin devrede olmadığı standart 20 optimizasyon denemesi sonrasında ağırlık değerlerinin standart sapması bulunmuş ($\sigma_{W_0}$) ve bulunan ağırlık değerlerinin ortalamasına ($\mu W_0$) oranlanarak bulunan varyasyon katsayısına ($c_v^0$) oranlanmıştır. Bu oranın 1'den küçük olması ortalama vekil modelsiz ağırlık değerlerine yakın sonuçların elde edildiği anlamına gelmektedir. 1'den büyük değerlerde amaç değerinin ortalamaya yakın olması istenirse vekil model desteğine gerek olmadığı anlaşılmalıdır.

## 6.2 10 Çubuklu Kafes Sistem Deneyleri

### 6.2.1 10 Çubuklu Kafes Sistem Problemi

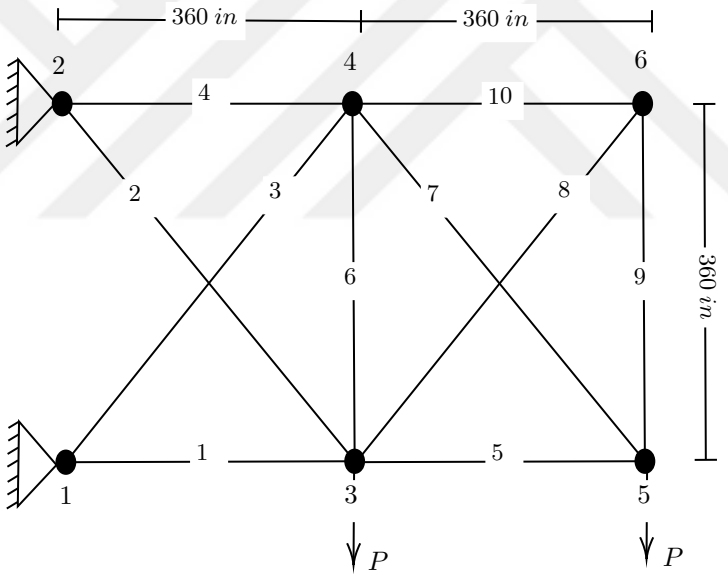
Problem, tamsayı ve sürekli tasarım değişkenlerini aynı anda ele aldığı için karma değişkenli eş zamanlı şekil, boyut ve topoloji optimizasyonu problemi olarak ele alınmıştır. Malzeme özellikleri Elastisite Modülü ($E$) ve Özgül Ağırlık ($\rho$) sırayla $10^4\ ksi$ ve $0.1\ lb/in^3$ olarak alınmıştır. On çubuklu kafes sistemin yükleme durumu $3$ ve $5$ düğümlerinin her ikisinde $P = 1000000\ lb$'dir (Şekil 6.1).

Şekil optimizasyonu için, on çubuklu kafes sistemde (Şekil 6.1) $1$ düğümü orijin olmak üzere, üç üst düğümün $(2, 4, 6)$ dikey koordinatlarının $180\ in$ ile $1000\ in$ arasında hareket etmesine izin verilmektedir (Miguel vd., 2013). Koordinat değerleri bu aralıkta

sürekli değişkenler olarak alınmaktadır. Boyut optimizasyonu için kesit alanları 6.5'te ilgili çalışmada verilen 32 ayrık değişken arasından seçilmiştir (Rajan, 1995).

$$\begin{aligned}&(1.62, 1.80, 2.38, 2.62, 2.88, 3.09, 3.13, 3.38, 3.63, 3.84, 3.87, 4.18, 4.49, 4.80,\\ &4.97, 5.12, 5.74, 7.22, 7.97, 11.5, 13.5, 13.9, 14.2, 15.5, 16.0, 13.9, 14.2, 15.5,\\ &13.9, 14.2, 15.5, 16.0, 18.8, 19.9, 22.0, 22.9, 26.5, 30.0, 33.5)\end{aligned} \tag{6.5}$$

Bu test probleminin eş zamanlı şekil, boyut ve topoloji optimizasyonu daha önce (Rajan, 1995) ve (Balling vd., 2006) tarafından Genetik Algoritma, (Martini, 2011) tarafından, Harmoni Araması Algoritması, (Miguel vd., 2013) tarafıdan Ateşböceği Algoritması kullanılarak gerçekleştirilmiştir. Her bir çalışmanın sonuçları Çizelge 6.3 'de paylaşılmıştır.




Şekil 6.1: On Çubuklu Kafes Sistem

On çubuklu kafes sistemin şekil, boyut ve topoloji optimizasyonu eş zamanlı olarak vekil model devrede olmadan, Jaya algoritması kullanılarak gerçekleştirilmiştir (Bknz. Bölüm 5.1).

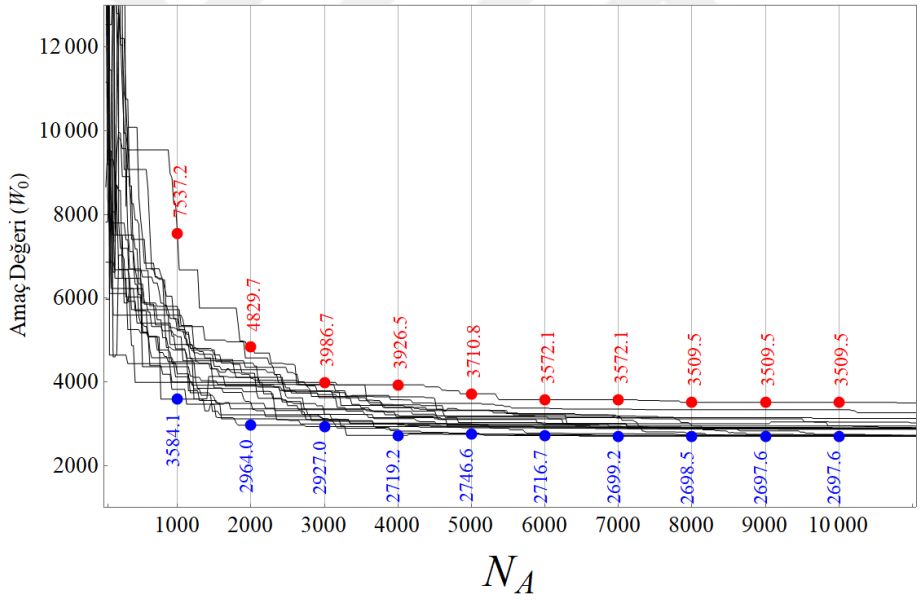
#### 6.2.1.1 10 Çubuk Test Problemi İçin Vekil Modelsiz Sonuçlar

Optimizasyon denemesi sayısı ($n_{Run}$), 20 olarak belirlenmiştir. Öncesinde optimizasyon süresince gerçekleştirilen, 10000 analize kadar olan süreçte, 20 optimizasyon

Çizelge 6.3: On Çubuklu Kafes Boyut, Şekil ve Topoloji Optimizasyonu için Referans Sonuçlar

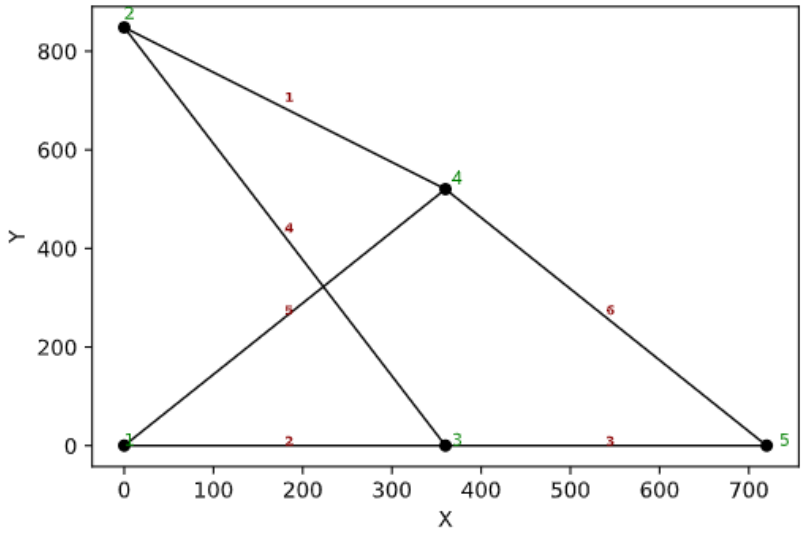
| **Sonuçlar** | **(Rajan, 1995)** | **(Balling vd., 2006)** | **(Martini, 2011)** | **(Miguel vd., 2013)** |
|---|---|---|---|---|
| Analiz Sayısı | 3840 | 500000 | 4075 | 50000 |
| Ağırlık ($lb$) | 3254 | 2736 | 2900 | 2705 |
| Ortalama Ağırlık ($lb$) | - | - | - | 2796 |

denemesi sonuçları paylaşılmış, deneme sonuçları arasında bulunan ilgili izin verilen analiz sayılarına ($N_A$), göre son popülasyondaki en optimuma yakın adaya ait en iyi ağırlık değerleri mavi renk ile, optimumdan en uzak adayların, en kötü ağırlık değerleri ise kırmızı renk ile gösterilmiştir (Şekil 6.2).



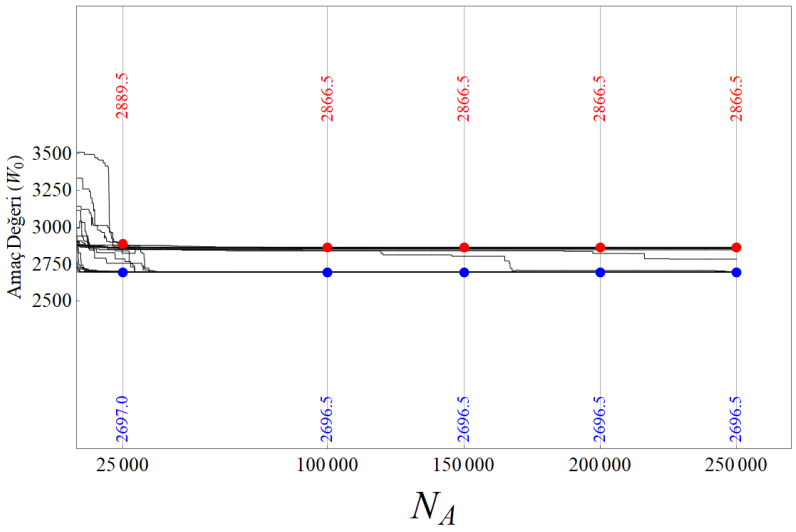

Şekil 6.2: On Çubuklu Kafes Sistem: İzin Verilen Analiz Sayısı - Vekil Modelsiz Optimum Sonuçlar (a)

On bininci analiz sonucunda bulunan en iyi boyut, şekil ve topoloji (Şekil 6.3) 6. denemede kaydedilmiştir. Bulunan en iyi ağırlık değeri herhangi bir kısıt ihlali olmadan $W_0 = 2697.6\ lb$ ve en iyi ve en kötü ortalama ağırlık değeri $W_0^{\mu} = 2712.77\ lb$ olarak bulunmuştur.




Şekil 6.3: On Çubuklu Kafes Sistem: On bin Analiz Sonucu Optimum Şekil ve Topoloji

Vekil model devrede olmadan optimizasyon süreci 250000 analize kadar devam ettirilmiş ve sonuçlar paylaşılmıştır (Şekil 6.4). 25000 analizden sonra amaç değeri biraz daha optimuma yaklaşırken, en iyi ve en kötü adayların amaç değerlerinde önemli bir değişim gözlenmemiştir (Bknz. Bölüm 6.1.1).




Şekil 6.4: On Çubuklu Kafes Sistem: İzin Verilen Analiz Sayısı - Vekil Modelsiz Optimum Sonuçlar (b)

On Çubuklu Kafes Sistem için vekil model devrede olmadan gerçekleştirilen optimizasyon süreçlerinin referans sonuçlarla (Bknz. Çizelge 6.3) karşılaştırıldığında tutarlı oldukları ve bu test problemi için literatürde optimum olarak sunulan sonuçlara çok yakın sonuçlar elde edildiği görülmektedir. Sunulan YSA tabanlı vekil model ile sonuçların karşılaştırılması ve performans ölçümü için vekil model devreye alınmadan gerçekleştirilen optimizasyon süreçlerinin tutarlı sonuçlar vermesi önemlidir. Öncelikle bundan sonraki bölümlerde sunulan metotta FAZ:1 sürecinde verinin toplanması ve vekil YSA'nın eğitim performansı irdelenecek (Bknz. Bölüm 6.1.2) ve çeşitli parametrelerin bu eğitimi nasıl etkilediği sunularak tartışalacaktır. Sunulan metot On Çubuklu Kafes Sistem probleminde test edilmiş ve sonuçlarla birlikte vekil modelin performansı irdelenmiştir (Bknz. Bölüm 6.1.3).

#### 6.2.1.2 10 Çubuk Test Problemi İçin YSA Eğitim Performansı

On çubuklu kafes sistem için vekilin eğitiminde toplanacak verilerin öncelikle Limit Ceza değeri ($limPen$) parametresi sırayla $1.0, 0.5, 0.1$ olarak alınmış ve SE analizlerine yollanan adayların bu Limit Ceza değerlerinin altında olan adaylar eğitim setine alınmak üzere toplanmıştır.

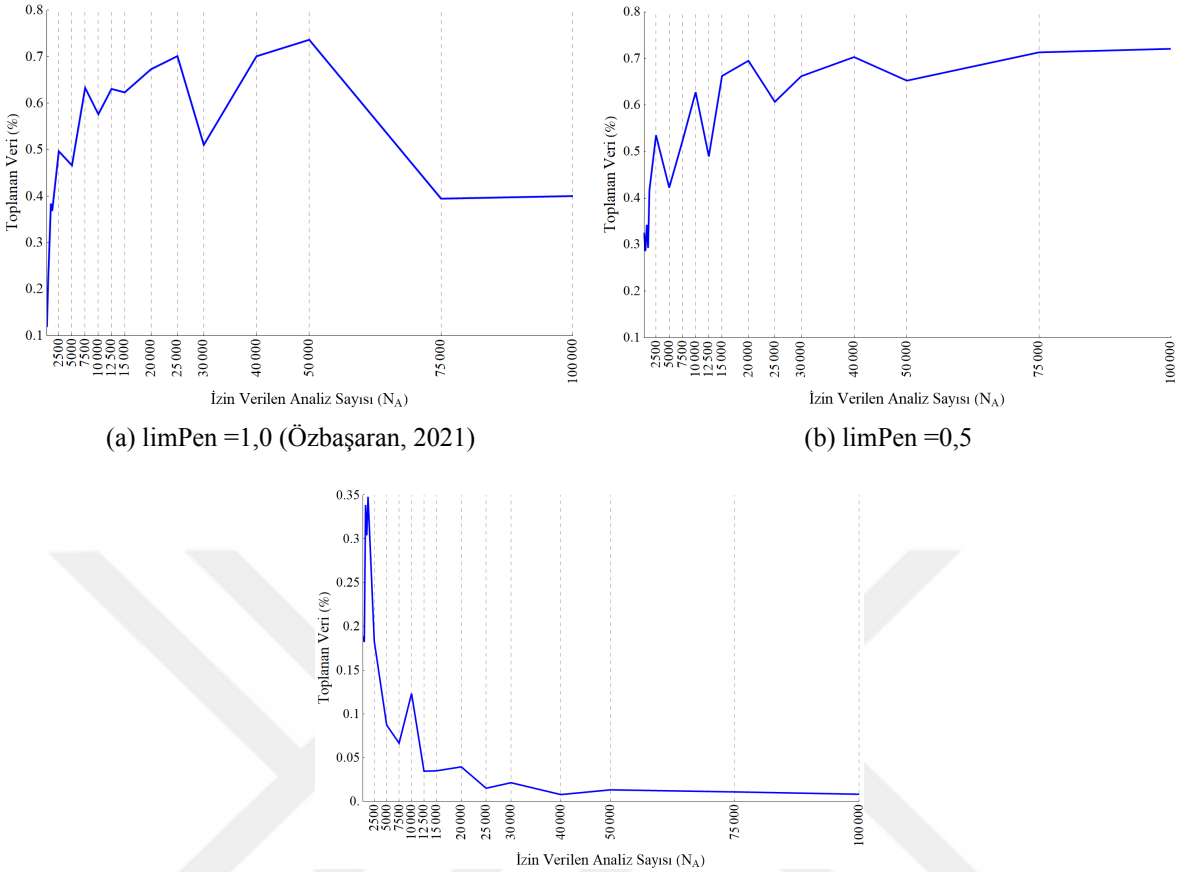
Verilerin $limPen$ parametresine bağlı, toplanma yüzdelerinin izin verilen analiz sayısına göre değişimleri grafiği Şekil 6.5'te sunulmuştur.

Limit Ceza ($limPen$) parametresine bağlı olarak her bir izin verilen analizde toplanan veriye göre eğitimin sonlandığı iterasyon değerlerinin grafiği Şekil 6.6'da sunulmuştur.

Seçilen Limit Ceza ($limPen$) parametrelerine bağlı, ara katmanlarda kullanılacak Aktivasyon Fonksiyonları'nın İzin Verilen Analiz ($N_A$) sayılarına göre Eğitim Ortalama Karesel Hata ($OKH$) 'ları grafikleri Şekil 6.7'de sunulmuştur.

Seçilen Limit Ceza ($limPen$) parametrelerine bağlı, ara katmanlarda kullanılacak Aktivasyon Fonksiyonları'nın İzin Verilen Analiz ($N_A$) sayılarına göre Test Ortalama Karesel Hata ($OKH$)'ları grafikleri Şekil 6.8'de sunulmuştur.

Seçilen Limit Ceza ($limPen$) parametrelerine bağlı, ara katmanlarda kullanılacak Aktivasyon Fonksiyonları'nın İzin Verilen Analiz ($N_A$) sayılarına göre Test/Eğitim Ortalama Karesel Hata ($OKH$) oranları grafikleri Şekil 6.9'da sunulmuştur.




(a) limPen =1,0 (Özbaşaran, 2021)

(b) limPen =0,5



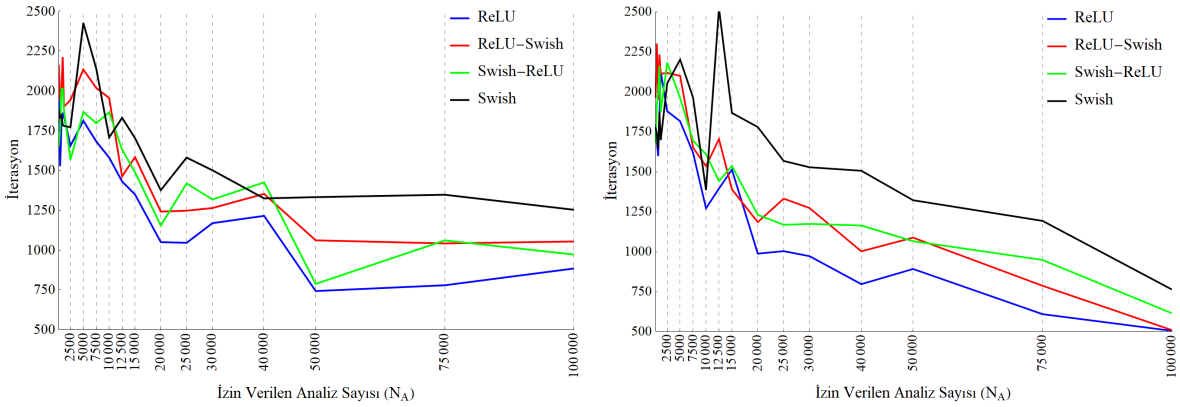

(c) limPen =0,1

Şekil 6.5: On Çubuklu Kafes Sistem: İzin Verilen Analiz Sayısı - Toplanan Veri Yüzdesi

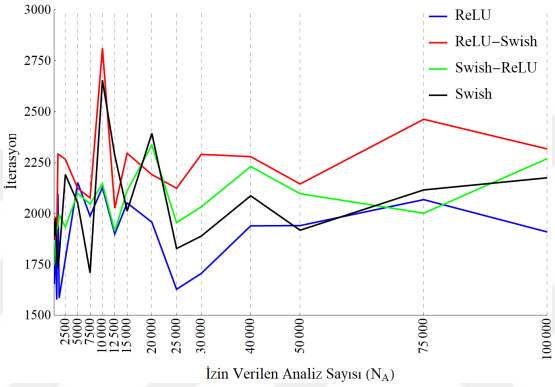
#### 6.2.1.3 10 Çubuk Test Problemi İçin Vekil Model Performansı

En iyi bağıl amaç değerinin 1'den küçük olması vekil modelin performansının vekilsiz modelden verimli olduğunu, büyük olması ise vekil model desteksiz metasezgiselin verimli olduğunu göstermektedir. On Çubuklu Kafes Sistem için $rP1$ değerinin 0.8 olarak belirlendiği 1500. analizde en iyi bağıl amaç değeri gözlenmiştir.

En kötü bağıl amaç değerinin 1'den küçük olması vekil modelin performansının vekil desteksiz metasezgiselden verimli olduğu, büyük olması ise vekil desteksiz metasezgiselin verimli olduğu anlamına gelmektedir. On Çubuklu Kafes Sistem için $rP1$ değerinin 0.8 olarak belirlendiği 250. analizde en kötü bağıl amaç değeri gözlenmiştir. Sonuçlar incelendiğinde problemin vekil modelin devrede olduğu optimizasyon sonuçlarının belirli parametre ve analiz sayılarında en kötü adaylarda kayda değer bir iyileşmeye sebep olduğu ve son popülasyonu iyileştirdiği söylenebilir.

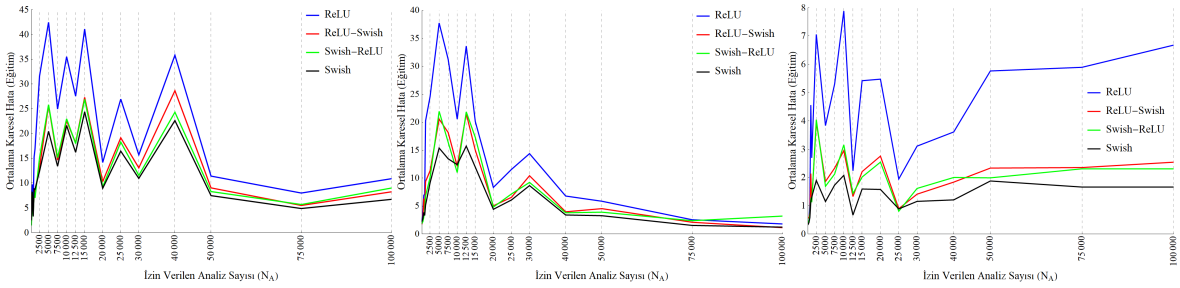



(a) limPen =1,0 (b) limPen =0,5




(c) limPen =0,1

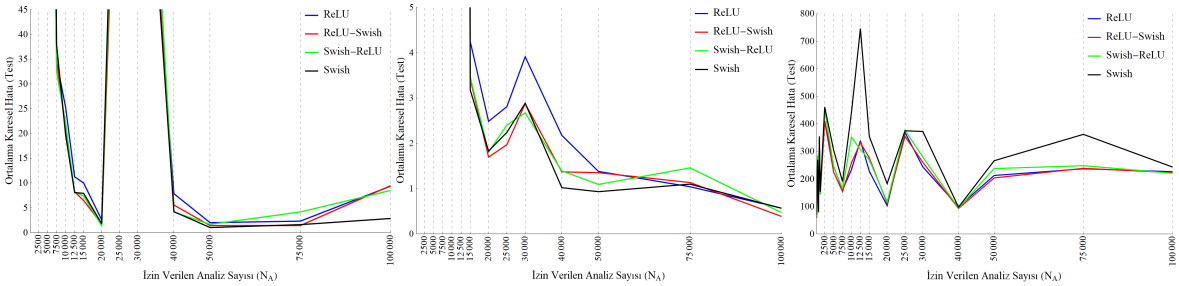
Şekil 6.6: On Çubuklu Kafes Sistem: İzin Verilen Analiz Sayısı - İterasyon Sayısı




(a) limPen =1,0 (Özbaşaran, 2021) (b) limPen =0,5 (c) limPen =0,1

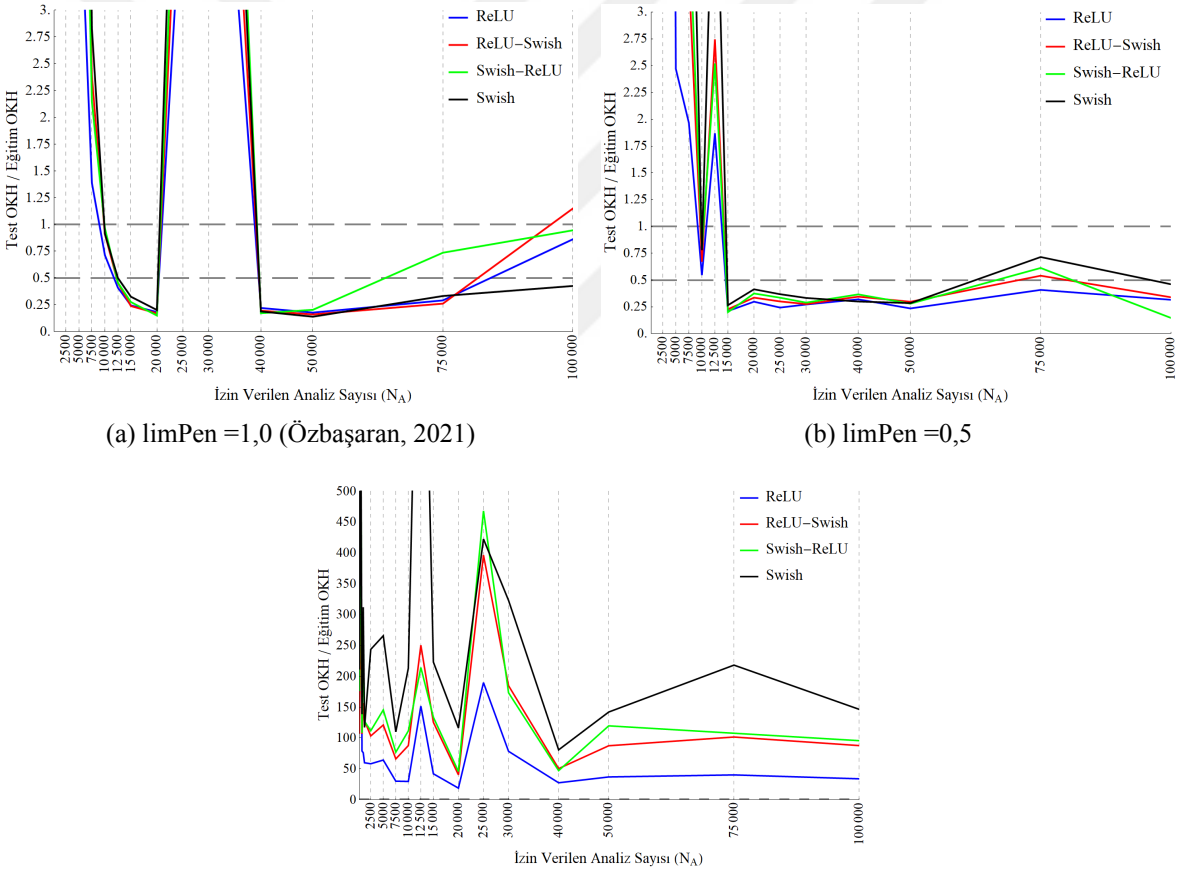
Şekil 6.7: On Çubuklu Kafes Sistem: İzin Verilen Analiz Sayısı - Eğitim OKH

Vekil model ile gerçekleştirilen optimizasyon sonucunda elde edilen son popülasyondaki tüm adayların amaç değerlerinin ortalamasının vekil desteksiz Jaya algoritması sonucunda bulunan amaç değerlerinin oranı ortalama bağıl amaç değerini




(a) limPen =1,0 (Özbaşaran, 2021) (b) limPen =0,5 (c) limPen =0,1

Şekil 6.8: On Çubuklu Kafes Sistem: İzin Verilen Analiz Sayısı - Test OKH

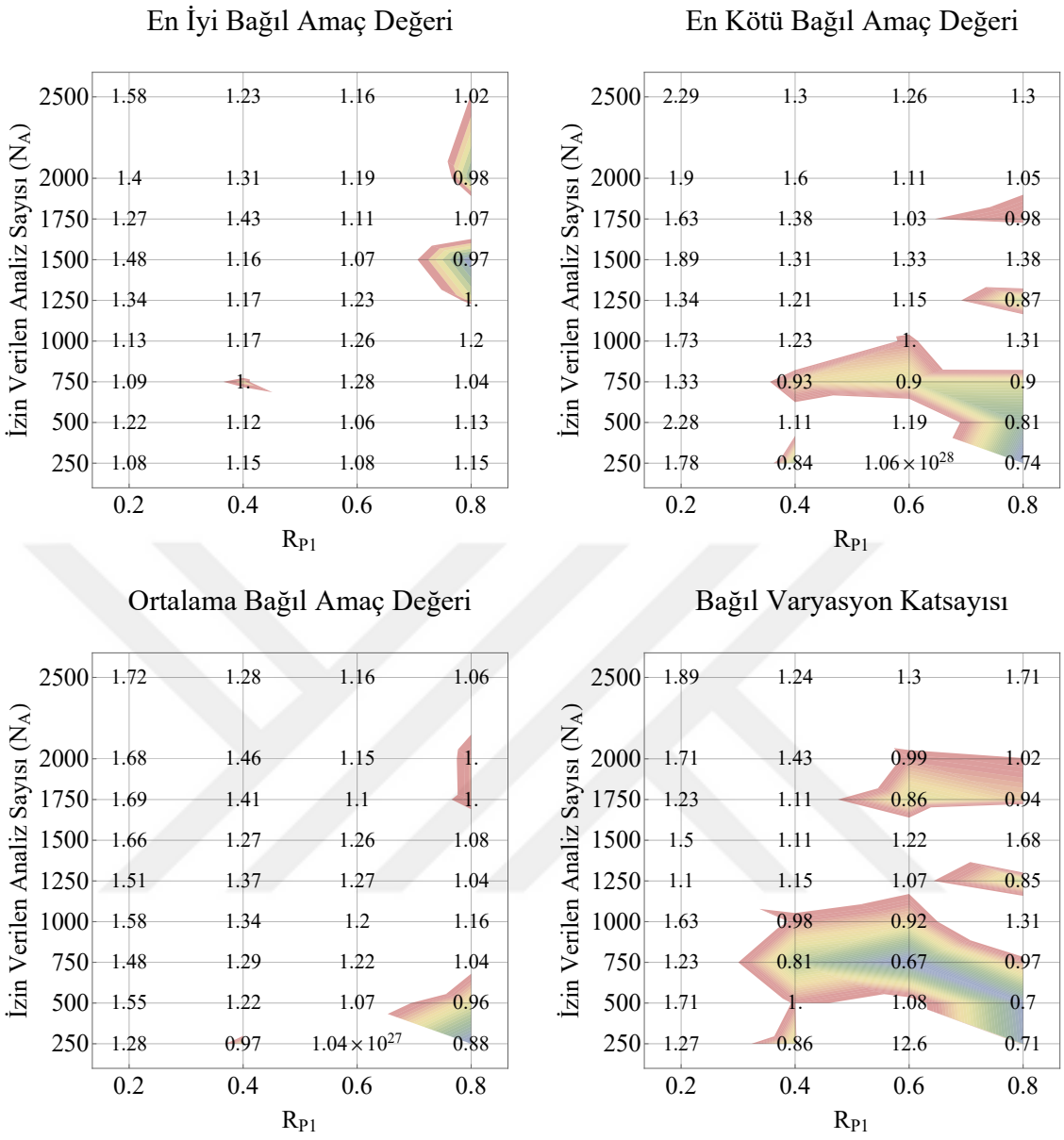



(a) limPen =1,0 (Özbaşaran, 2021) (b) limPen =0,5

(c) limPen =0,1

Şekil 6.9: On Çubuklu Kafes Sistem: İzin Verilen Analiz Sayısı - Test / Eğitim OKH

vermektedir. Böylece vekil modelle bulunan popülasyonun grafikteki merkezi eğilimi bulunabilir. Ortalama bağıl amaç değerlerine bakıldığında FAZ:1'de toplam izin verilen analiz sayısının 250 olduğu durumda ve bu analizlerin %80'i eğitim verisinin toplanması




Şekil 6.10: On Çubuklu Kafes Sistem: Vekil Model Performans Sonuçları

için kullanıldığında vekil modelin popülasyonun genelinde optimum çözüme vekilin devrede olmadığı süreçten daha performanslı yaklaştığı izin verilen 500 analizde performansın düştüğü ve daha yüksek analiz sayılarında en fazla vekil modelin olmadığı optimizasyon süreci kadarlık bir performans sergilediği görülmektedir.

Bağıl varyasyon katsayısı oranın 1'den küçük olması vekil model desteği ile ortalama vekil modelin devrede olmadığı ağırlık değerlerine yakın sonuçların elde edildiği anlamına gelmektedir. 750 analizde $rP1$ parametresinin 0,6 olarak seçildiği durumda bağıl varyasyon katsayısı en iyi değeri vermiştir. Bağıl varyasyon katsayısının 1'in altında olduğu izin verilen

analiz sayıları ve $rP1$ parametreleri değerlerinde vekil modelin yararlı olabileceği, 1'den büyük değerlerde amaç değerinin ortalamaya yakın olması istenirse vekil model desteğine gerek olmadığı görülmektedir.

## 6.3 15 Çubuklu Kafes Sistem Deneyleri

### 6.3.1 15 Çubuklu Kafes Sistem Problemi

Problem, tamsayı ve sürekli tasarım değişkenlerini aynı anda ele aldığı için karma değişkenli eş zamanlı şekil, boyut ve topoloji optimizasyonu problemi olarak ele alınmıştır. Malzeme özellikleri Elastisite Modülü ($E$) ve Özgül Ağırlık ($\rho$) sırayla $10^4\ ksi$ ve $0.1\ lb/in^3$ olarak alınmıştır. On beş çubuklu kafes sistemin dikey yükleme durumu 7. düğümde $P = 10000\ lb$'dir (Şekil 6.11). İzin verilen gerilme değeri $25\ ksi$'dir.Boyut optimizasyonu için 15 kesit alanı ve şekil optimizasyonu için $3, 4, 5, 6$ düğümlerinin $x, y$ koordinatları, $7, 8$ düğümlerinin $y$ koordinatları tasarım değişkenleri olarak alınmıştır. $4$ ve $6$ koordinatları sırasıyla $3$ ve $5$ koordinat değerlerine eşittir (6.6).

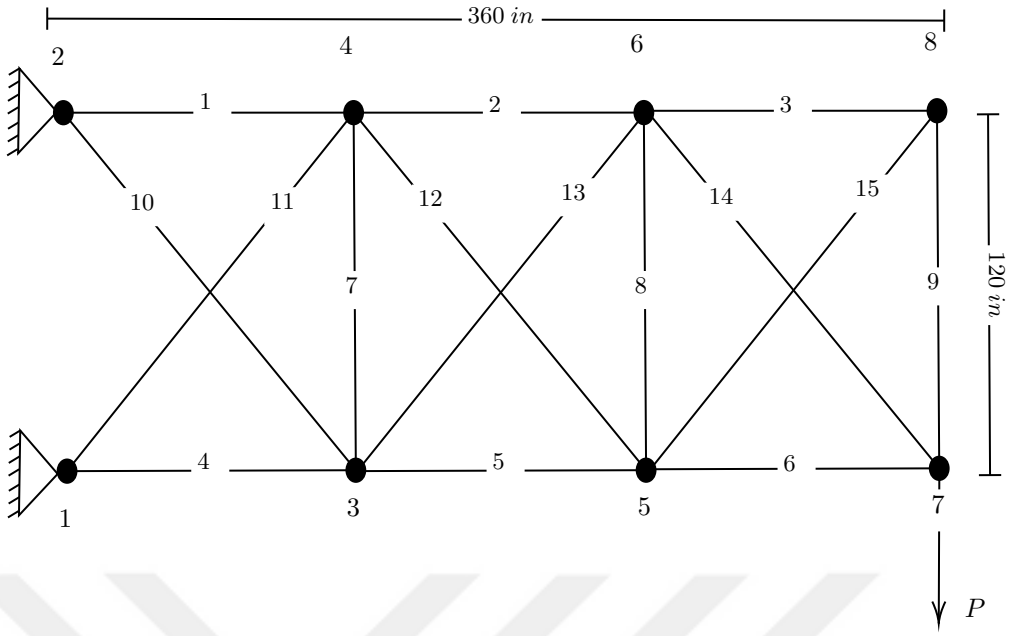
$$(x_4 = x_3, x_6 = x_5, y_3, y_4, y_5, y_6, y_7, y_8) \tag{6.6}$$

Kesit alanları, topopoloji optimizasyonu için $\{-30, 31\}$ değerleri arasından rastgele olarak alınmış ve rastgele değerlere göre 32 ayrık değişken (6.7) arasından seçilmiştir (Miguel vd., 2013).

$$\begin{aligned}(&0.111, 0.141, 0.174, 0.220, 0.270, 0.287, 0.347, 0.440, 0.539, 0.954, 1.081,\\ &1.174, 1.333, 1.488, 1.764, 2.142, 2.697, 2.800, 3.131, 3.565, 3.813, 4.805,\\ &5.952, 6.572, 7.192, 8.525, 9.300, 10.850, 13.330, 14.290, 17.170, 19.180)\end{aligned} \tag{6.7}$$

Şekil optimizasyonu için düğüm noktaları koordinat kısıt aralıkları 6.8'de verilmiştir.

$$\begin{aligned} -20 &\leq y_3 \leq 20\\ 100 &\leq x_4 = x_3 \leq 140\\ 100 &\leq y_4 \leq 140\\ -20 &\leq y_5 \leq 20\\ 220 &\leq x_6 = x_5 \leq 260\\ 100 &\leq y_6 \leq 140\\ 20 &\leq y_7 \leq 60\\ 50 &\leq y_8 \leq 90 \end{aligned} \tag{6.8}$$




Şekil 6.11: On beş Çubuklu Kafes Sistem

Çizelge 6.4: On beş Çubuklu Kafes Sistem Tasarım Parametreleri

| **Tasarım Parametreleri** | **Değer** |
|---|---|
| Elastisite Modülü | $10^4\ ksi$ |
| Özgül Ağırlık | $0.1\ lb/in^3$ |
| İzin Verilen Basınç Gerilmesi | $25\ ksi$ |
| İzin Verilen Çekme Gerilmesi | $25\ ksi$ |
| İzin Verilen Dikey Yerdeğiştirme | $2\ in$ |

Bu test probleminin eş zamanlı şekil, boyut ve topoloji optimizasyonu daha önce Tang vd., 2005'te ve Wu ve Chow, 1995'te Genetik Algoritma ile, Miguel vd., 2013'te Ateşböceği Algoritması ile gerçekleştirilmiştir (Çizelge 6.5).

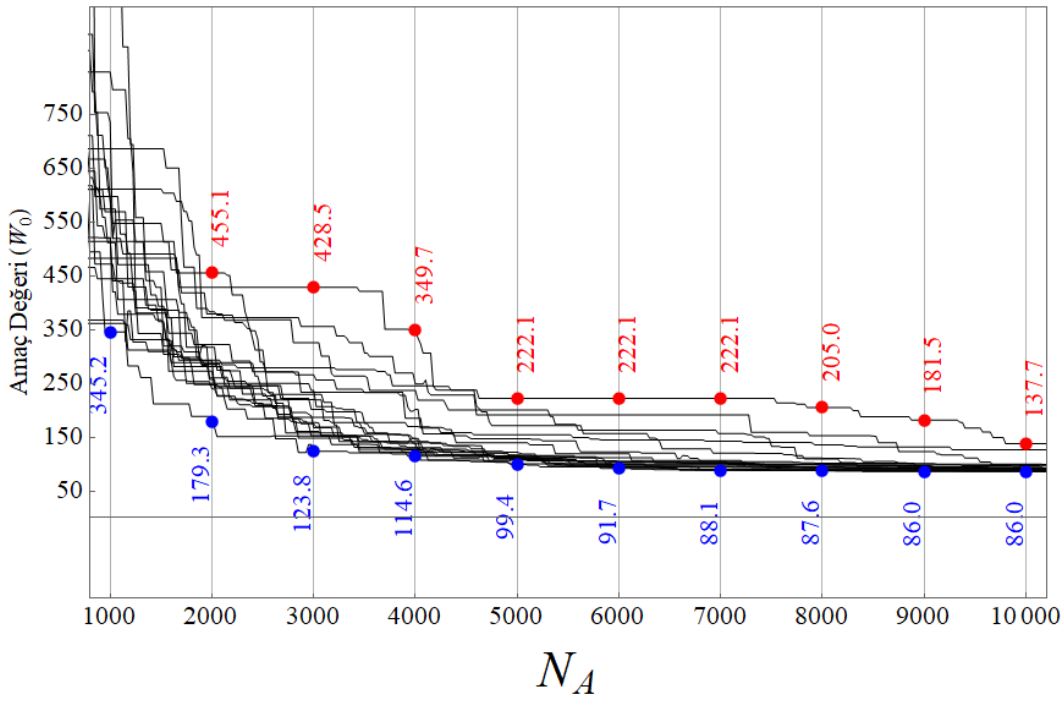
On beş çubuklu kafes sistemin şekil, boyut ve topoloji optimizasyonu eş zamanlı olarak vekil model devrede olmadan, Jaya algoritması kullanılarak gerçekleştirilmiştir (Bknz. Bölüm 5.1).

Çizelge 6.5: On beş Çubuklu Kafes Boyut, Şekil ve Topoloji Optimizasyonu için Referans Sonuçlar

| **Sonuçlar** | **Wu ve Chow, 1995** | **Tang vd., 2005** | **Miguel vd., 2013** |
|---|---|---|---|
| Analiz Sayısı | - | 8000 | 8000 |
| Ağırlık ($lb$) | 120.528 | 79.82 | 75.55 |
| Ortalama Ağırlık ($lb$) | - | 95.38 | - |

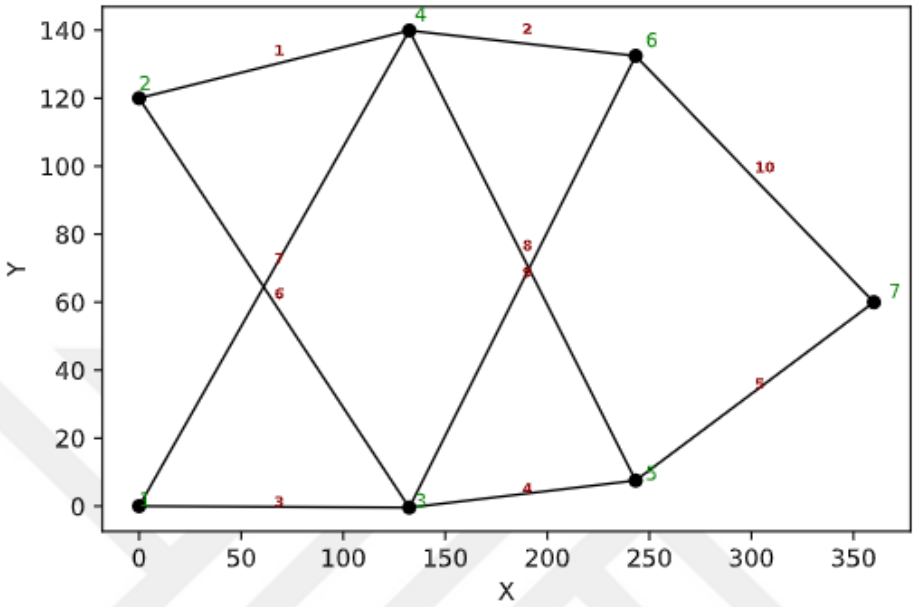
#### 6.3.1.1 15 Çubuk Test Problemi İçin Vekil Modelsiz Sonuçlar

Optimizasyon denemesi sayısı $nRun$, 20 olarak belirlenmiştir. Öncesinde optimizasyon süresince gerçekleştirilen, 10000 analize kadar olan süreçte, 20 optimizasyon denemesi sonuçları paylaşılmış, deneme sonuçları arasında bulunan ilgili izin verilen analiz sayılarına ($N_A$), göre son popülasyondaki en optimuma yakın adaya ait en iyi ağırlık değerleri mavi renk ile, optimumdan en uzak adayların, en kötü ağırlık değerleri ise kırmızı renk ile gösterilmiştir (Şekil 6.12).




Şekil 6.12: On Beş Çubuklu Kafes Sistem: İzin Verilen Analiz Sayısı - Vekil Modelsiz Optimum Sonuçlar (a)

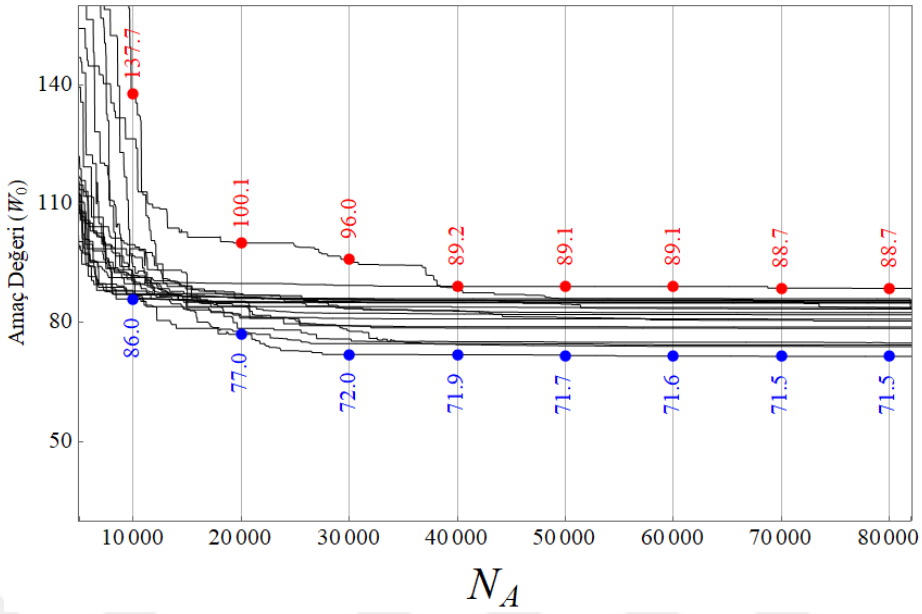
Bulunan en iyi boyut, şekil ve topoloji (Şekil 6.13), 4.denemede kaydedilmiştir. Bulunan en iyi ağırlık değeri herhangi bir kısıt ihlali olmadan $W_0$= 71.5 $lb$ ve en iyi ve en kötü ortalama ağırlık değeri $W_{\mu 0}$= 80.1 $lb$ olarak bulunmuştur (Şekil 6.14).




Şekil 6.13: On beş Çubuklu Kafes Sistem: Bulunan Optimum Şekil ve Topoloji

Vekil model devrede olmadan optimizasyon süreci 250000 analize kadar devam ettirilmiş ve sonuçlar paylaşılmıştır (Şekil 6.14). 250000 analiz sonucunda optimuma en yakın değere ulaşılmış, en iyi ve en kötü adayların amaç değerlerinde önemli bir değişim gözlenmemiştir.

On beş Çubuklu Kafes Sistem için vekil model devrede olmadan gerçekleştirilen optimizasyon süreçlerinin referans sonuçlarla (Bknz. Çizelge 6.5) karşılaştırıldığında tutarlı oldukları ve bu test problemi için 8000 analiz için paylaşılan referans sonuçlarla karşılaştırıldığında literatürde optimum olarak sunulan sonuçlara yakın sonuçlar elde edildiği görülmektedir. Sunulan YSA tabanlı vekil model ile sonuçların karşılaştırılması ve performans ölçümü için vekil model devreye alınmadan gerçekleştirilen optimizasyon süreçleri için tutarlı sonuçların elde edilmesi önemlidir. Öncelikle bundan sonraki bölümlerde sunulan metotta FAZ:1 sürecinde verinin toplanması ve vekil YSA'nın eğitim performansı irdelenecek ve çeşitli parametrelerin bu eğitimi nasıl etkilediği sunularak tartışalacaktır. Sunulan metot, On beş Çubuklu Kafes Sistem probleminde test edilmiş ve elde edilen sonuçlarla birlikte vekil modelin performansı irdelenmiştir.




Şekil 6.14: On Beş Çubuklu Kafes Sistem: İzin Verilen Analiz Sayısı - Vekil Modelsiz Optimum Sonuçlar (b)

#### 6.3.1.2 15 Çubuk Test Problemi İçin YSA Eğitim Performansı

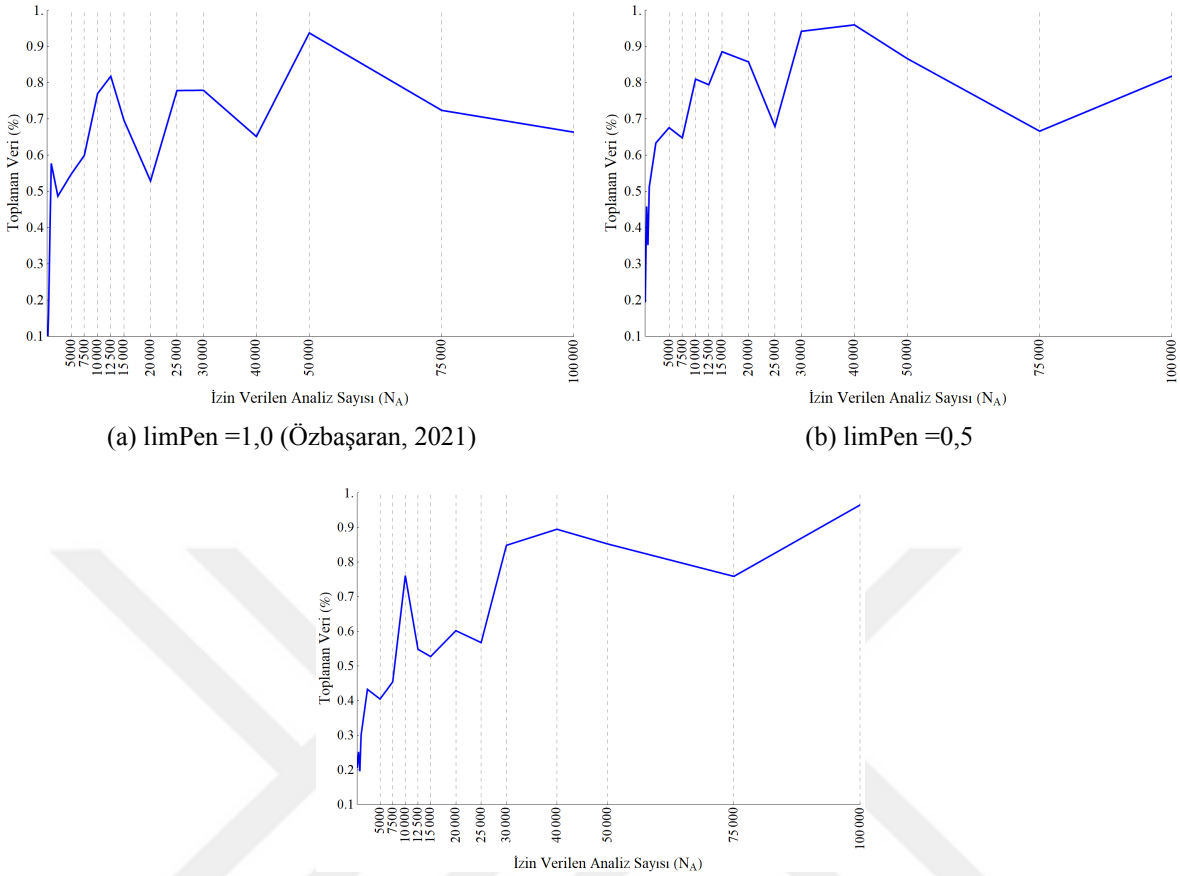
On beş çubuklu kafes sistem için vekilin eğitiminde toplanacak verilerin öncelikle Limit Ceza değeri ($limPen$) parametresi sırayla $1.0, 0.5, 0.1$ olarak belirlenmiş ve SE analizlerine yollanan adayların bu Limit Ceza değerlerinin altında olan adaylar eğitim setine alınmak üzere toplanmıştır.

Verilerin $limPen$ parametresine bağlı, toplanma yüzdelerinin izin verilen analiz sayısına göre değişimleri grafiği Şekil 6.15'te sunulmuştur.

Limit Ceza ($limPen$) parametresine bağlı olarak her bir izin verilen analizde toplanan veriye göre eğitimin sonlandığı iterasyon grafiği Şekil 6.16'da sunulmuştur.

Seçilen Limit Ceza ($limPen$) parametrelerine bağlı, ara katmanlarda kullanılacak Aktivasyon Fonksiyonları'nın İzin Verilen Analiz ($N_A$) sayılarına göre Eğitim Ortalama Karesel Hata ($OKH$) 'ları grafikleri Şekil 6.17'de sunulmuştur.

Seçilen Limit Ceza ($limPen$) parametrelerine bağlı, ara katmanlarda kullanılacak Aktivasyon Fonksiyonları'nın İzin Verilen Analiz ($N_A$) sayılarına göre Test Ortalama Karesel Hata ($OKH$)'ları grafikleri Şekil 6.18'de sunulmuştur.




(a) limPen =1,0 (Özbaşaran, 2021)

(b) limPen =0,5

(c) limPen =0,1

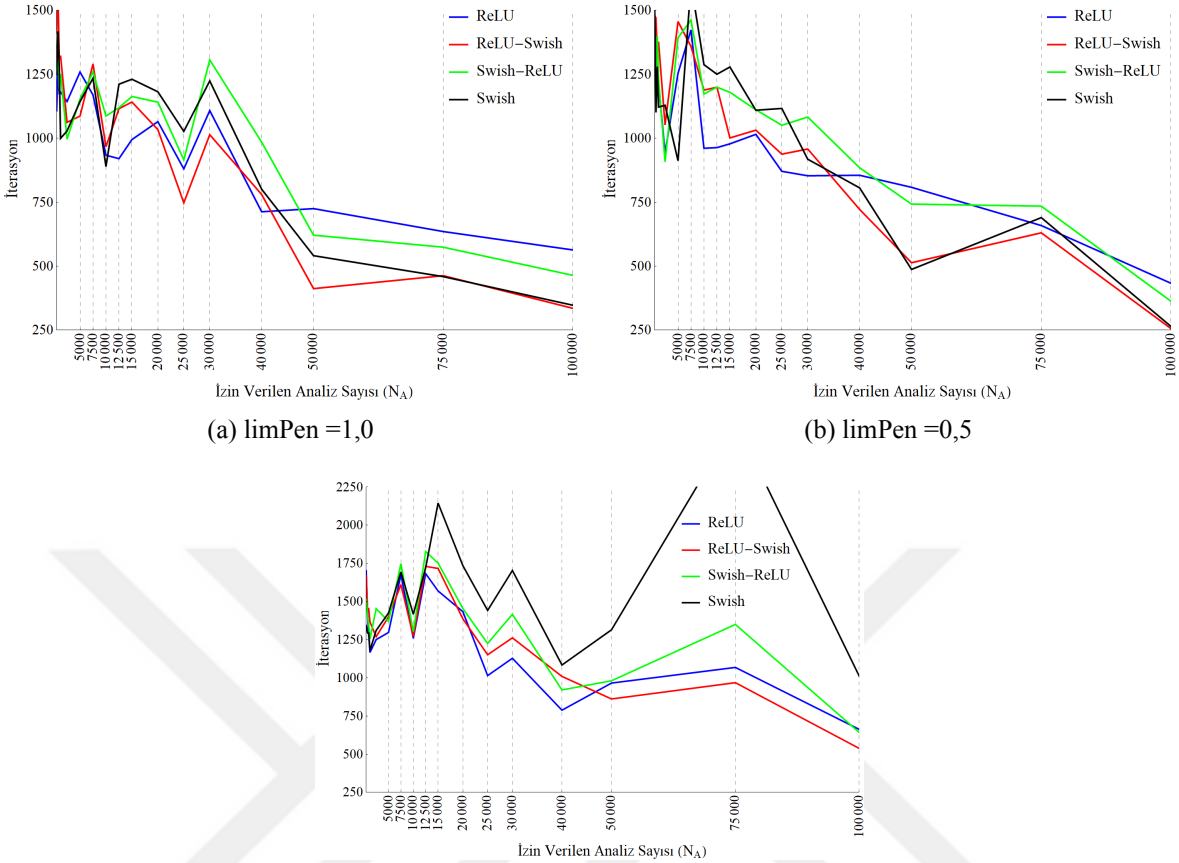
Şekil 6.15: On beş Çubuklu Kafes Sistem: İzin Verilen Analiz Sayısı - Toplanan Veri Yüzdesi

Seçilen Limit Ceza ($limPen$) parametrelerine bağlı, ara katmanlarda kullanılacak Aktivasyon Fonksiyonları'nın İzin Verilen Analiz ($N_A$) sayılarına göre Test/Eğitim Ortalama Karesel Hata ($OKH$) oranları grafikleri Şekil 6.19'de sunulmuştur.

#### 6.3.1.3 15 Çubuk Test Problemi İçin Vekil Model Performansı

On beş çubuklu Kafes Sistem problemi için sunulan vekil modelin (Bknz. Şekil 5.2) performans değerlendirmesi için istatistiki sonuçlar paylaşılmıştır (Şekil 6.20).

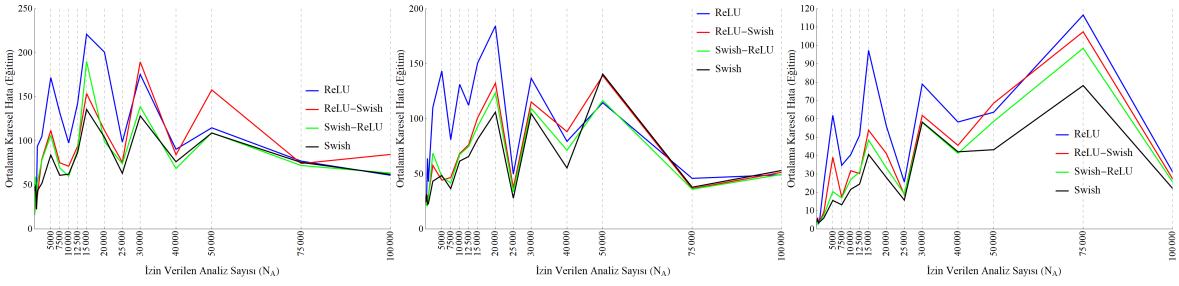
En iyi bağıl amaç değeri oranın 1'den küçük olması vekil modelin optimum değere yaklaşmada aynı analiz sayılarında vekilsiz gerçekleştirilen optimizasyon sürecinden verimli olduğunu, büyük olması ise sunulan vekil modelin kullanımının verimli olmadığını işaret etmektedir. On beş Çubuklu Kafes Sistem için $r_{P1}$ değerinin her değeri için $N_A = 1250$ analiz değerleri için verimli olduğu görülmektedir. En iyi bağıl amaç değeri




(a) limPen =1,0 (b) limPen =0,5

(c) limPen =0,1

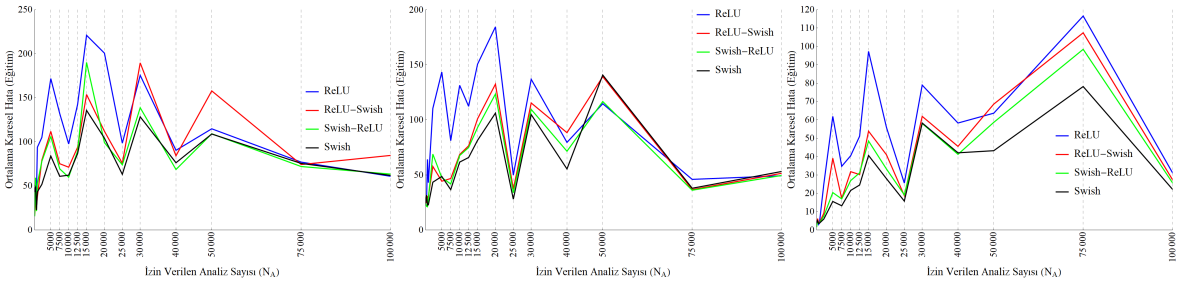
Şekil 6.16: On beş Çubuklu Kafes Sistem: İzin Verilen Analiz Sayısı - İterasyon Sayısı




(a) limPen =1,0 (Özbaşaran, 2021) (b) limPen =0,5 (c) limPen =0,1

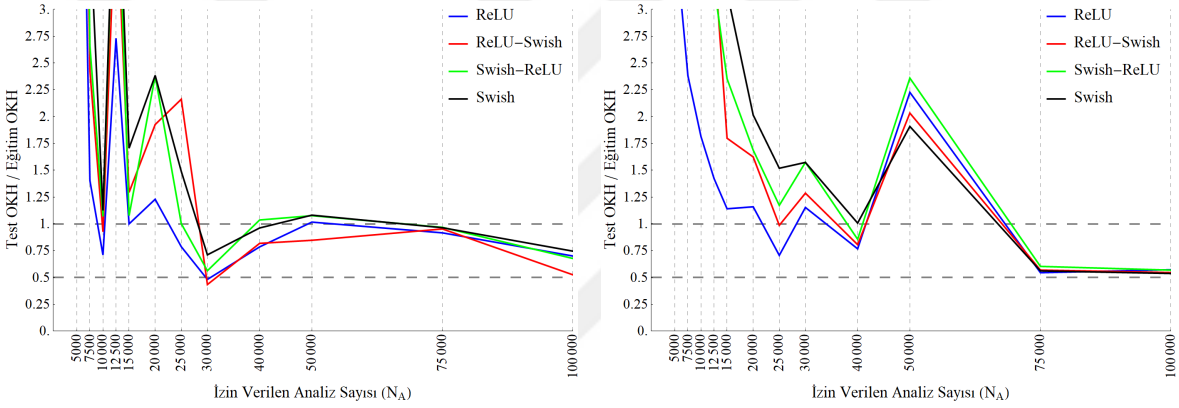
Şekil 6.17: On beş Çubuklu Kafes Sistem: İzin Verilen Analiz Sayısı - Eğitim OKH

için en iyi performans değerlerinin On beş çubuklu test probleminde olduğu görülmüştür (Şekil 6.20).

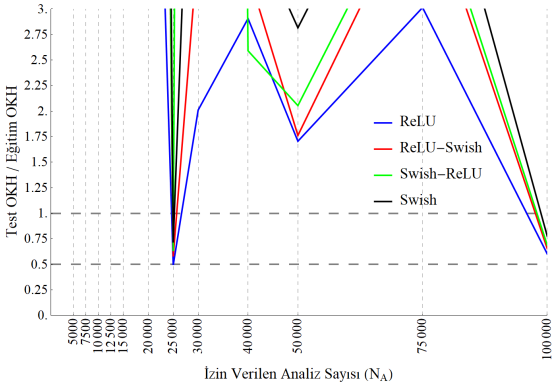



(a) limPen =1,0 (Özbaşaran, 2021) (b) limPen =0,5 (c) limPen =0,1

Şekil 6.18: On beş Çubuklu Kafes Sistem: İzin Verilen Analiz Sayısı - Test OKH



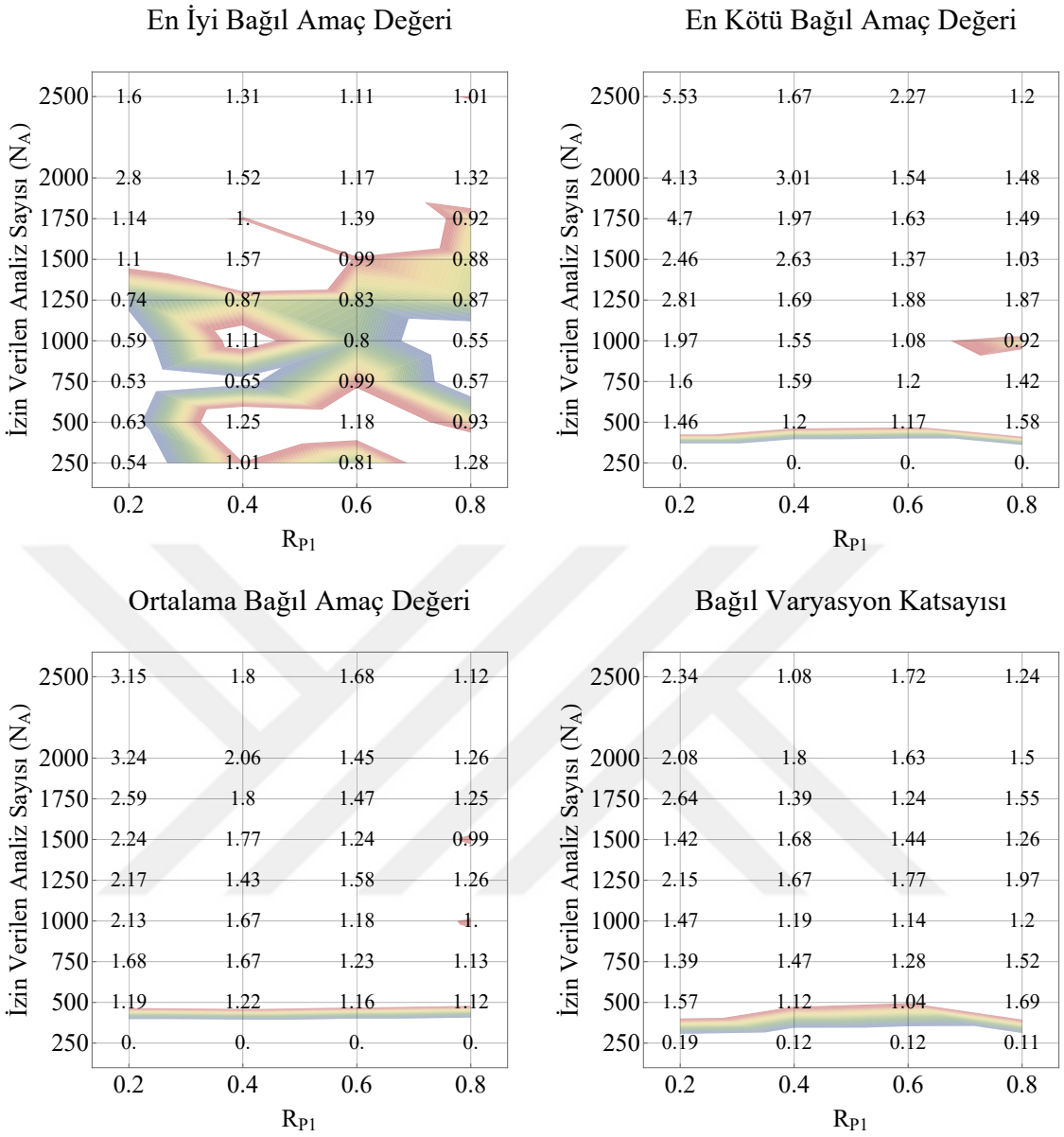

(a) limPen =1,0 (Özbaşaran, 2021) (b) limPen =0,5




(c) limPen =0,1

Şekil 6.19: On beş Çubuklu Kafes Sistem: İzin Verilen Analiz Sayısı - Test / Eğitim OKH

En kötü bağıl amaç değerinin 1'den küçük olması vekil modelin performansının popülasyondaki en kötü adayın iyileştirilmesinde vekilin devrede olmadığı modelden daha verimli olduğunu, büyük olması ise sunulan vekil modelin kullanımının verimli olmadığını




Şekil 6.20: On beş Çubuklu Kafes Sistem: Vekil Model Performans Sonuçları

işaret etmektedir. On beş Çubuklu Kafes Sistem için $r_P1$ değerinin 0.8 olarak belirlendiği izin verilen analiz değeri 1000 olduğunda en kötü bağıl amaç değeri ve vekil model ile %8'lik bir iyileşme gözlenmiştir. Ancak genel olarak en kötü bağıl amaç değerleri için vekil modelin verimli olmadığı görülmektedir (Şekil 6.20).

Ortalama bağıl amaç değerlerine bakıldığında On bes çubuklu kafes sistemde vekil modelin kullanılmasına gerek olmadığı çıkmaktadır (Şekil 6.20).

Bağıl varyasyon katsayısının 1'den küçük olması ortalama vekil modelsiz ağırlık değerlerine yakın sonuçların elde edildiği anlamına gelmektedir. Bağıl varyasyon katsayısnın 1'in altında olduğu izin verilen analiz sayıları ve $rP1$ parametrelerine göre vekil modelin kullanılabileceğini göstermektedir. 1'den büyük değerlerde amaç değerinin ortalamaya yakın olması istenirse vekil model desteğine gerek olmadığı anlamına gelmektedir. Bu problem için ortalama ağrılık değerlerine ulaşmak için vekil desteğinin verimsiz olduğu düşünülmektedir.

## 6.4 39 Çubuklu Kafes Sistem Deneyleri

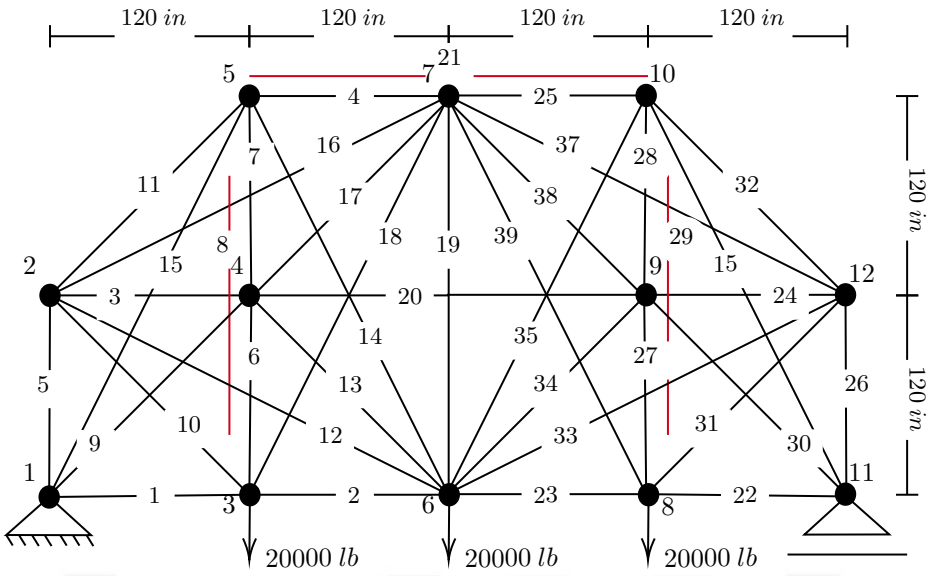
### 6.4.1 39 Çubuklu Kafes Sistem Problemi

Problem, 39 çubuk ve 12 düğümden oluşmakta, yalnızca sürekli tasarım değişkenlerini ele aldığı için sürekli değişkenli eş zamanlı şekil, boyut ve topoloji optimizasyonu problemi olarak ele alınmaktadır. Malzeme özellikleri Elastisite Modülü ($E$) ve Özgül Ağırlık ($\rho$) sırayla $10^4$ $ksi$ ve $0.1$ $lb/in^3$ olarak alınmıştır. Otuz dokuz çubuklu kafes sistemin 3, 6 ve 8. düğümlerde dikey yükleme durumu $P = -20000$ $lb$'dir (Şekil 6.21).

$$(y_7, x_2 = -x_{12}, y_2 = -y_{12}, x_4 = x_9, y_4 = y_9, x_5 = x_{10}, y_5 = y_{10}) \tag{6.9}$$

İzin verilen basma ve çekme gerilmeleri değeri $25$ $ksi$, deplasman kısıtı her bir düğüm için $2$ $in$'dir (Çizelge 6.6). Tasarım değişkenleri; şekil optimizasyonu için belirlenen düğüm koordinatları 6.9 dikkate alınarak, arama uzayından $[-120in, 120in]$ sürekli değişkenleri arasından rastgele seçilmiş, topoloji ve boyut optimizasyonu için kesit alanları kafes sistemin simetrisinden dolayı 21 adet $(A_1, ...A_{21})$ olacak şekilde arama uzayından [-0.225,2.25] ($in^2$) değerleri arasındasından 0.5 $in^2$ ve üstündeki kesit alanları var olarak kabul edilecek şekilde seçilmiştir.

Bu test probleminin eş zamanlı şekil, boyut ve topoloji optimizasyonu daha önce, Ahrari vd., 2015'de, Evrim Stratejisi ile, Miguel vd., 2013'te, Ateşböceği Algoritması ile, Wu ve Tseng, 2010'da, Çok Popülasyonlu Diferansiyel Evrim ile, Luh ve Lin, 2011'de Parçacık Sürüsü Algoritması ile, Deb ve Gulati, 2001'de Genetik Algoritma ile gerçekleştirilmiştir (Çizelge 6.7).




Şekil 6.21: Otuz dokuz Çubuklu Kafes Sistem

Çizelge 6.6: Otuz dokuz Çubuklu Kafes Sistem Tasarım Parametreleri

| **Tasarım Parametreleri** | **Değer** |
|---|---|
| Elastisite Modülü | $10^4\ ksi$ |
| Özgül Ağırlık | $0.1\ lb/in^3$ |
| İzin Verilen Basınç Gerilmesi | $25\ ksi$ |
| İzin Verilen Çekme Gerilmesi | $25\ ksi$ |
| İzin Verilen Dikey Yerdeğiştirme | $2\ in$ |

Otuz dokuz Çubuklu Kafes sistemin şekil, boyut ve topoloji optimizasyonu eş zamanlı olarak vekil model devrede olmadan, Jaya algoritması kullanılarak gerçekleştirilmiştir (Bknz. Bölüm 5.1).

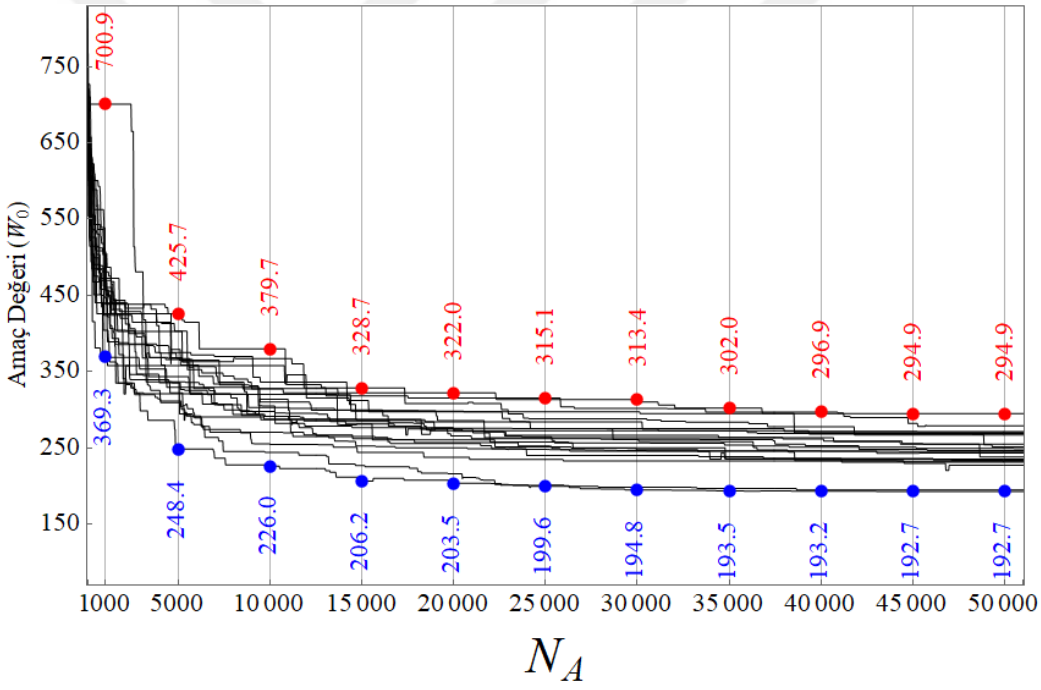
#### 6.4.1.1 39 Çubuk Test Problemi İçin Vekil Modelsiz Sonuçlar

Optimizasyon denemesi sayısı $nRun$, 20 olarak belirlenmiştir. Öncesinde optimizasyon süresince gerçekleştirilen, 50000 analize kadar olan süreçte, 20 optimizasyon denemesi sonuçları paylaşılmış, deneme sonuçları arasında bulunan ilgili izin verilen analiz sayılarına ($N_A$), göre son popülasyondaki en optimuma yakın adaya ait en iyi ağırlık

Çizelge 6.7: Otuz Çubuklu Kafes Boyut, Şekil ve Topoloji Optimizasyonu için Referans Sonuçlar

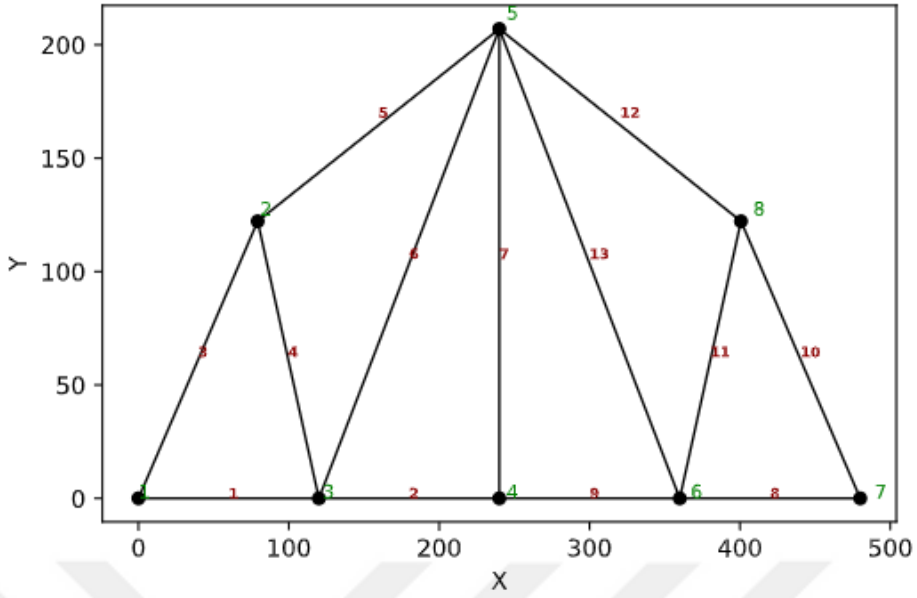
| **Sonuçlar** | **Ahrari vd., 2015** | **Miguel vd., 2013** | **Wu ve Tseng, 2010** | **Luh ve Lin, 2011** | **Deb ve Gulati, 2001** |
|---|---|---|---|---|---|
| Analiz Sayısı | 40256 | 50000 | 137200 | 262500 | 504000 |
| Ağırlık ($lb$) | 180.98 | 191.30 | 188.45 | 188.86 | 191.19 |

değerleri mavi renk ile, optimumdan en uzak adayların, en kötü ağırlık değerleri ise kırmızı renk ile gösterilmiştir (Şekil 6.22).



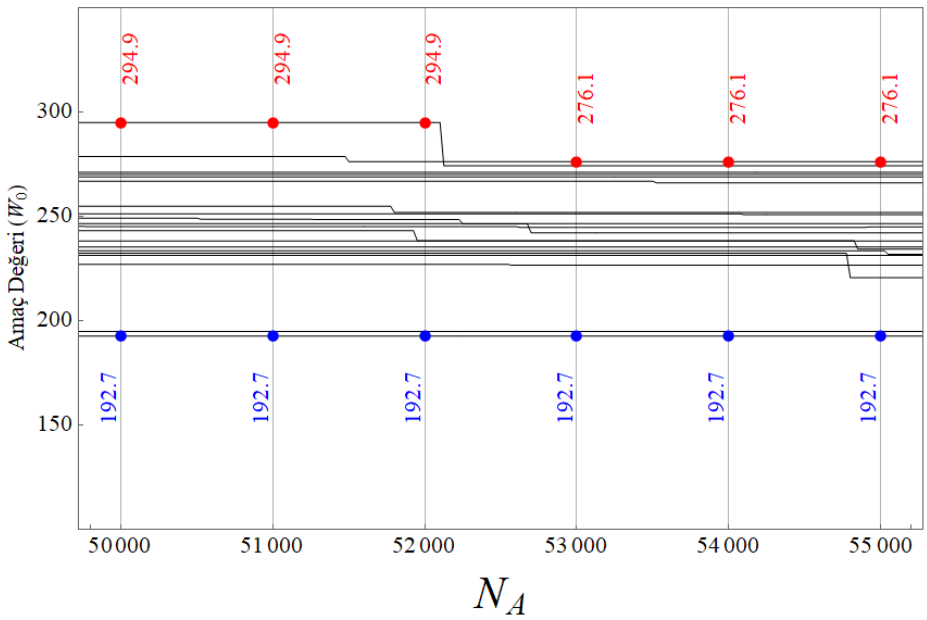

Şekil 6.22: Otuz Dokuz Çubuklu Kafes Sistem: İzin Verilen Analiz Sayısı - Vekil Modelsiz Optimum Sonuçlar (a)

Bulunan en iyi boyut, şekil ve topoloji (Şekil 6.23), 10.denemede kaydedilmiştir. Bulunan en iyi ağırlık değeri herhangi bir kısıt ihlali olmadan $W_\mu 0$= 192.7 $lb$ olarak bulunmuştur (Şekil 6.22).




Şekil 6.23: Otuz dokuz Çubuklu Kafes Sistem: Bulunan Optimum Şekil ve Topoloji

Vekil model devrede olmadan optimizasyon süreci 55000 analize kadar devam ettirilmiş ve sonuçlar paylaşılmıştır (Şekil 6.24). 55000 analiz sonucunda optimuma en yakın değere ulaşılmış, en iyi ve en kötü adayların amaç değerlerinde önemli bir değişim gözlenmemiştir.




Şekil 6.24: Otuz Dokuz Çubuklu Kafes Sistem: İzin Verilen Analiz Sayısı - Vekil Modelsiz Optimum Sonuçlar (b)

Otuz dokuz Çubuklu Kafes Sistem için vekil model devrede olmadan gerçekleştirilen optimizasyon süreçlerinin referans sonuçlarla (Bknz. Çizelge 6.7) karşılaştırıldığında tutarlı oldukları ve bu test problemi için 50000 analiz için paylaşılan referans sonuçlarla karşılaştırıldığında literatürde optimum olarak sunulan sonuçlara yakın sonuçlar elde edildiği görülmektedir. Sunulan YSA tabanlı vekil model ile sonuçların karşılaştırılması ve performans ölçümü için vekil model devreye alınmadan gerçekleştirilen optimizasyon süreçleri için tutarlı sonuçların elde edilmesi önemlidir. Öncelikle bundan sonraki bölümlerde sunulan metotta vekil YSA'nın eğitim performansı irdelenecek ve çeşitli parametrelerin bu eğitimi nasıl etkilediği sunularak tartışalacaktır. Sunulan metot ayrıca, Otuz dokuz Çubuklu Kafes Sistem probleminde test edilmiş ve elde edilen sonuçlarla birlikte vekil modelin performansı irdelenmiştir.

#### 6.4.1.2 39 Çubuk Test Problemi İçin YSA Eğitim Performansı

On çubuklu kafes sistem için vekilin eğitiminde toplanacak verilerin öncelikle Limit Ceza değeri ($limPen$) parametresi sırayla $1.0, 0.5, 0.1$ olarak belirlenmiş ve SE analizlerine yollanan adayların, bu Limit Ceza değerlerinin altında olan adaylar eğitim setine alınmak üzere toplanmıştır. Verilerin $limPen$ parametresine bağlı, toplanma yüzdelerinin izin verilen analiz sayısına göre değişimleri grafiği Şekil 6.25'te sunulmuştur.

Limit Ceza ($limPen$) parametresine bağlı olarak her bir izin verilen analizde toplanan veriye göre eğitimin sonlandığı iterasyon grafiği Şekil 6.26'da sunulmuştur.

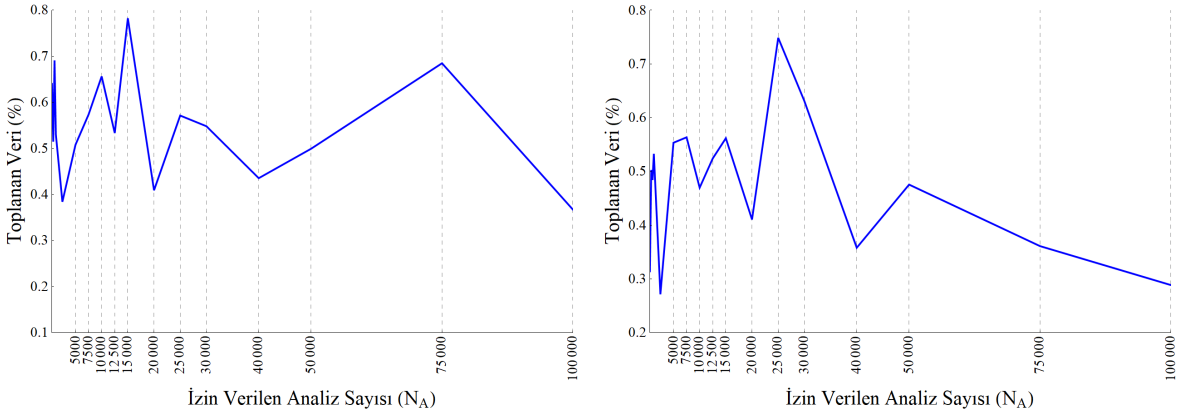
Seçilen Limit Ceza ($limPen$) parametrelerine bağlı, ara katmanlarda kullanılacak Aktivasyon Fonksiyonları'nın İzin Verilen Analiz ($N_A$) sayılarına göre Eğitim Ortalama Karesel Hata ($OKH$) 'ları grafikleri Şekil 6.27'de sunulmuştur.

Seçilen Limit Ceza ($limPen$) parametrelerine bağlı, ara katmanlarda kullanılacak Aktivasyon Fonksiyonları'nın İzin Verilen Analiz ($N_A$) sayılarına göre Test Ortalama Karesel Hata ($OKH$)'ları grafikleri Şekil 6.28'de sunulmuştur.

Seçilen Limit Ceza ($limPen$) parametrelerine bağlı, ara katmanlarda kullanılacak Aktivasyon Fonksiyonları'nın İzin Verilen Analiz ($N_A$) sayılarına göre Test/Eğitim Ortalama Karesel Hata ($OKH$) oranları grafikleri Şekil 6.29'de sunulmuştur.

#### 6.4.1.3 39 Çubuk Test Problemi İçin Vekil Model Performansı

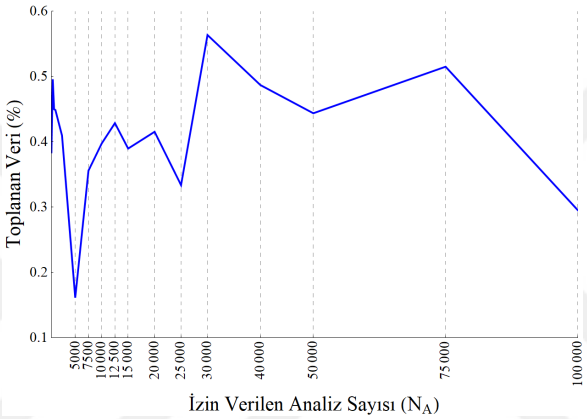
En iyi bağıl amaç değeri oranın 1'den küçük olması vekil modelin optimum değere yaklaşmada aynı analiz sayılarında vekilsiz gerçekleştirilen optimizasyon sürecinden verimli olduğunu, büyük olması ise sunulan vekil modelin kullanımının verimli olmadığını




(a) limPen =1,0




(b) limPen =0,5

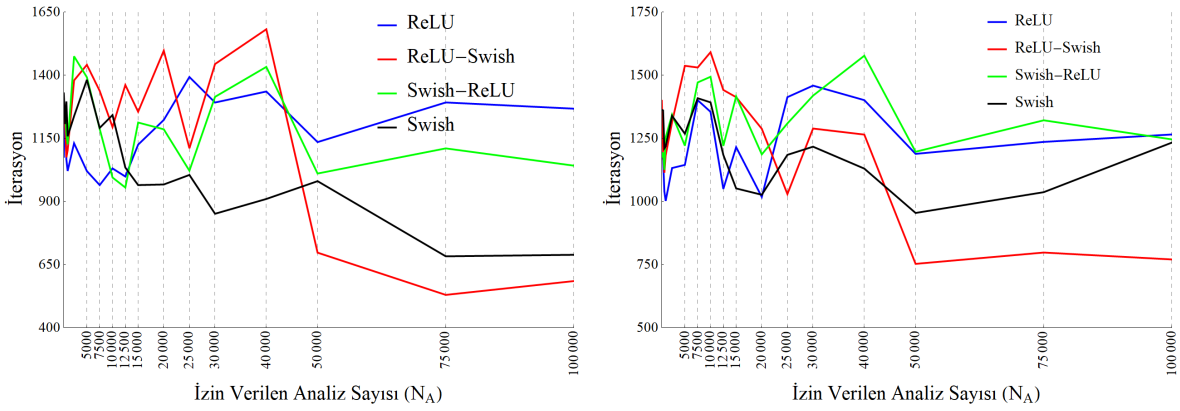



(c) limPen =0,1

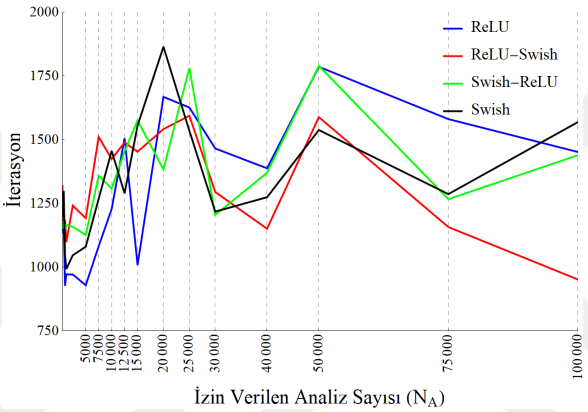
Şekil 6.25: Otuz dokuz Çubuklu Kafes Sistem: İzin Verilen Analiz Sayısı - Toplanan Veri Yüzdesi

işaret etmektedir. Otuz dokuz Çubuklu Kafes Sistem için $r_P1$ değerinin 0.4 ve 0.8 olduğu değerlerde belirli izin verilen analiz değerlerinde maksimum %9 verim sağladığı görülmektedir. İlgili performans grafiği incelendiğinde popülasyondaki en iyi adayların iyileştirilmesinde yeterli verimi sağlamadığı söylenebilir.

En kötü bağıl amaç değerinin 1'den küçük olması vekil modelin performansının popülasyondaki en kötü adayın iyileştirilmesinde vekilin devrede olmadığı modelden daha verimli olduğunu, büyük olması ise sunulan vekil modelin kullanımının verimli olmadığını işaret etmektedir. Otuz dokuz Çubuklu Kafes Sistem için sunulan vekil modelin en umut vadeden sonuçları en kötü bağıl amaç değerinde gözlenmektedir. Vekil modelin devrede olmadığı olağan optimizasyon sürecine göre daha optimuma yakın adayların olduğu bir son popülasyon ve genel olarak %29'a kadar daha optimum bir son popülasyon gözlenmiştir. Problemin test kafes sistem problemleri arasındaki en karmaşık problem olduğu

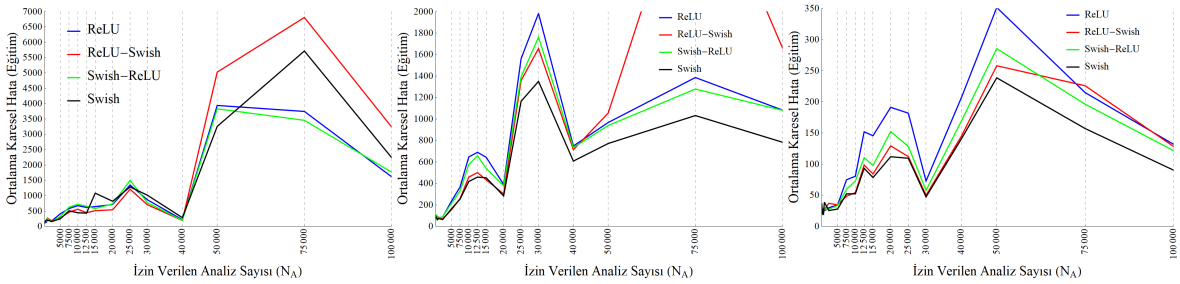



(a) limPen =1,0 (b) limPen =0,5




(c) limPen =0,1

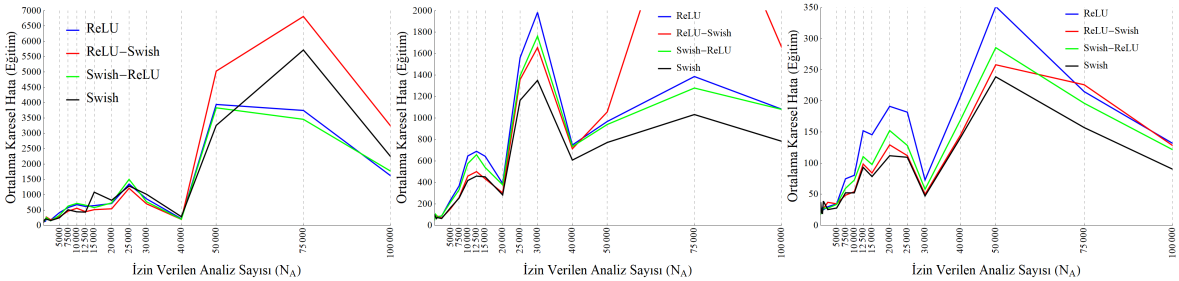
Şekil 6.26: Otuz dokuz Çubuklu Kafes Sistem: İzin Verilen Analiz Sayısı - İterasyon Sayısı




(a) limPen =1,0 (b) limPen =0,5 (c) limPen =0,1

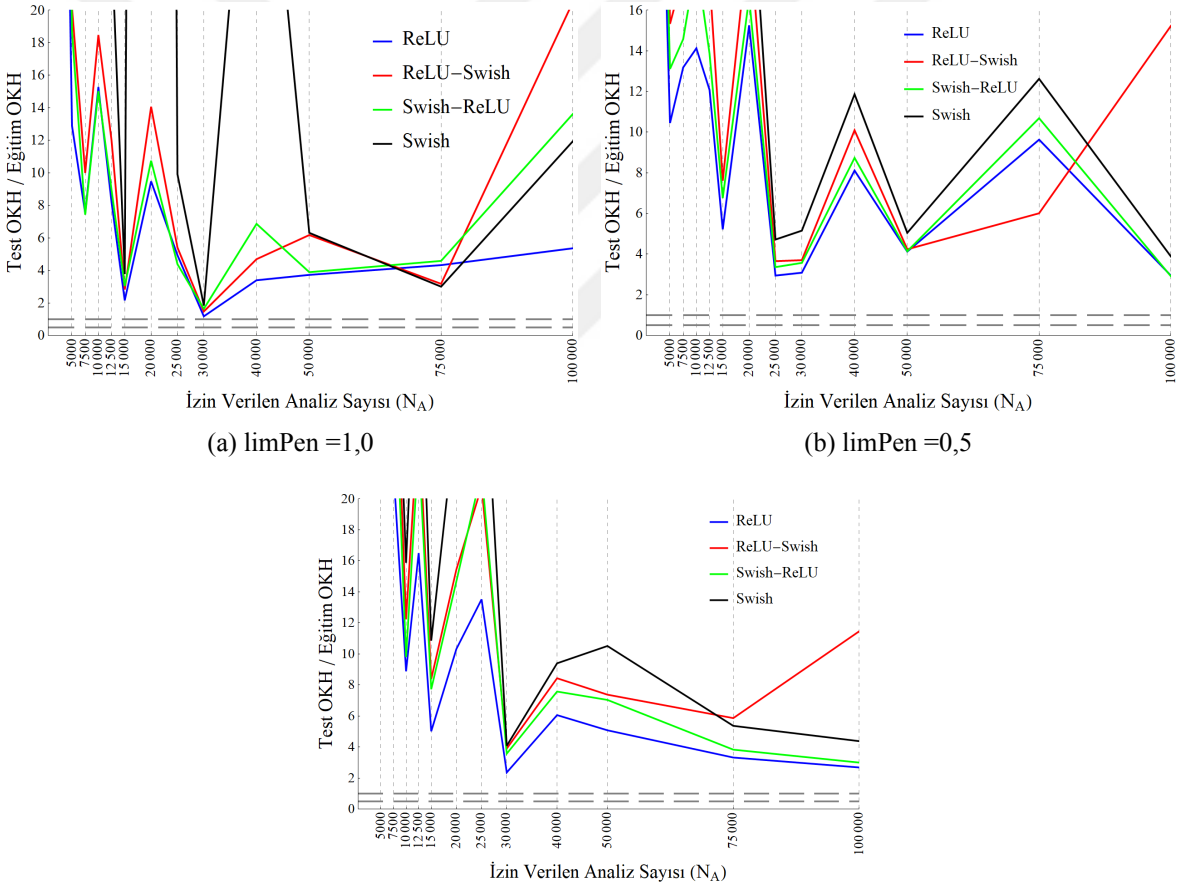
Şekil 6.27: Otuz dokuz Çubuklu Kafes Sistem: İzin Verilen Analiz Sayısı - Eğitim OKH

hatırlatılmalıdır. Vekil modelin sınırlı analizlere izin verilen ve sert kısıtların olduğu problemlerde problemin büyüklüğünün artmasıyla birlikte metasezgiselin olağan




(a) limPen =1,0 (b) limPen =0,5 (c) limPen =0,1

Şekil 6.28: Otuz dokuz Çubuklu Kafes Sistem: İzin Verilen Analiz Sayısı - Test OKH

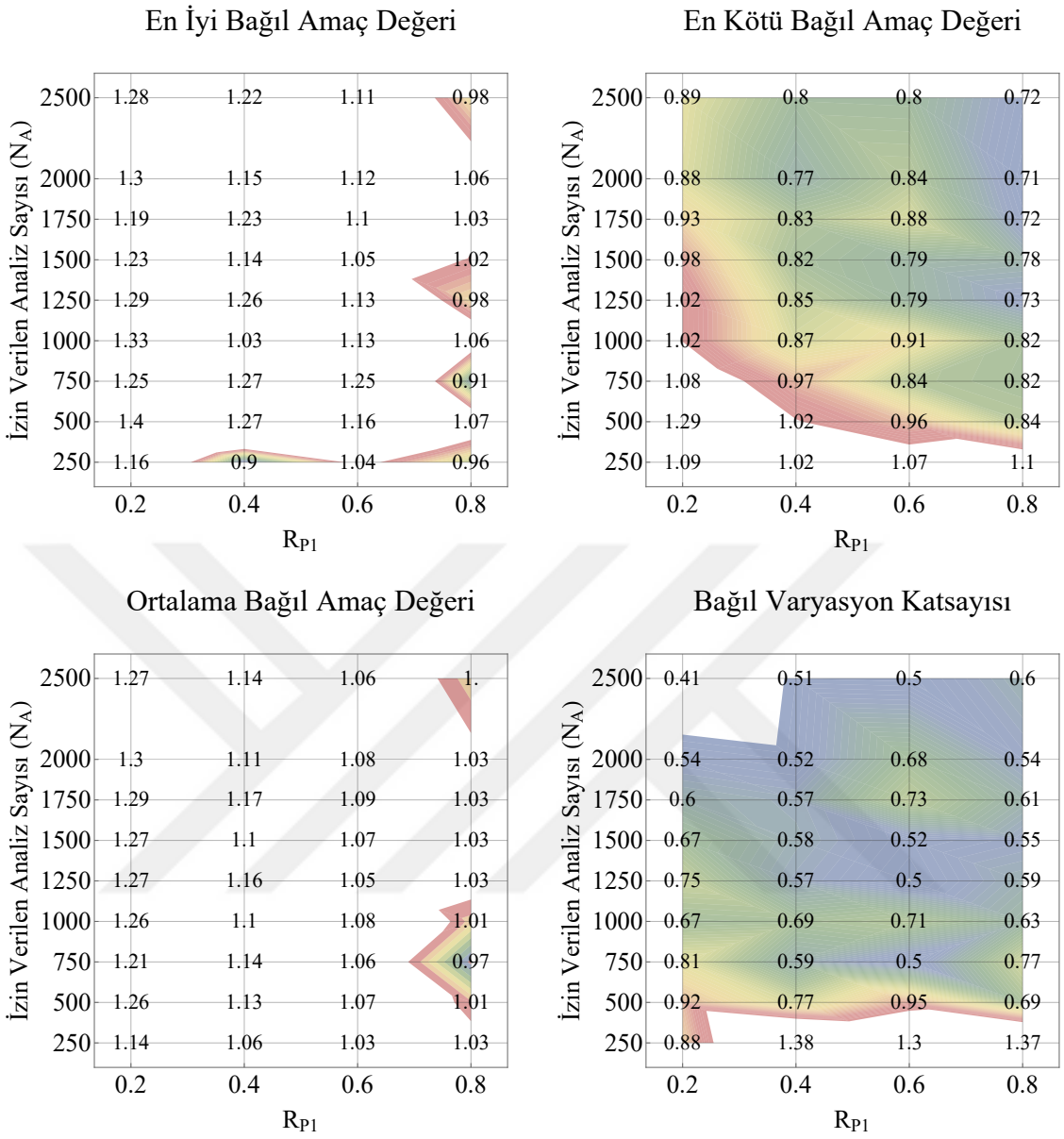



(a) limPen =1,0 (b) limPen =0,5

(c) limPen =0,1

Şekil 6.29: Otuz dokuz Çubuklu Kafes Sistem: İzin Verilen Analiz Sayısı - Test / Eğitim OKH

optimumumu arama sürecinde kullanılan metasezgiseli optimuma yakın noktalara taşıdığı çıkarımı yapılabilir.




Şekil 6.30: Otuz dokuz Çubuklu Kafes Sistem: Vekil Model Performans Sonuçları

Bağıl amaç değeri ile vekil model vasıtasıyla bulunan popülasyonun grafikteki merkezi eğilimi bulunabilir. Ortalama bağıl amaç değerlerine bakıldığında On bes çubuklu kafes sistemde vekil modelin kullanılmasına gerek olmadığı çıkarımı yapılabilir.

Bağıl Varyasyon katsayısı değerinin 1'den küçük olması ortalama vekil modelsiz ağırlık değerlerine yakın sonuçların elde edildiği anlamına gelmektedir. Bağıl varyasyon katsayısının 1'in altında olduğu izin verilen analiz sayıları ve $rP1$ parametrelerine göre vekil modelin kullanılabileceğini göstermektedir. 1'den büyük değerlerde amaç değerinin ortalamaya yakın olması istenirse vekil model desteğine gerek olmadığı görülmektedir.

Otuz dokuz çubuklu kafes sistem bağıl varyasyon katsayısı değerleri incelendiğinde, bu değerlerin genel olarak 1'den küçük oldukları görülmekte, amaç değerinin ortalamaya yakın değerler verdiği sonucu çıkmaktadır. Böylece vekil modelin kullanımının test problemleri arasındaki en karmaşık kafes sistemde, verimli olduğu söylenebilir.

# 7. SONUÇ VE ÖNERİLER

Vekil modelle yapılan deneyler sonucunda üç popüler kafes sistem probleminin sonuçları incelenmiş ve ulaşılan sonuçlar ve öneriler aşağıda maddeler halinde sıralanmıştır.

• Sunulan vekil destekli optimizasyon tekniğinin, ihlaline kesinlikle izin verilmeyen sıkı kısıtların olduğu optimizasyon problemlerinde etkili olmadığı görülmüştür. Özellikle diğer Makine Öğrenme teknikleriyle birlikte Yapay Sinir Ağı ile kestirim modellerinin vekil olarak desteklendiği optimizasyon süreçlerinde, kısıt ihlallerinin belirlenmesi Sonlu Elemanlar analizlerinin tahminlerini içermektedir. Doğası gereği YSA tahminleri, kısıt ihlallerinin hesaplamaları için kullanılan Sonlu Elemanlar analizlerinin döndürdüğü gerçek değerlerle aralarında bir hata payı bulundurmaktadır (Açıkgöz ve Özbaşaran, 2020). Bu hata, Yapay Sinir Ağı'nın eğitim performanslarına göre küçük olabilir ancak yine de SE analizlerinin yaklaşık sonuçlarını döndürdüklerinden, kısıt ihlalleri olmasa bile metasezgiselin bu ihlalleri varmış gibi algılamasına ve optimizasyon sürecini yanıltıcı bir davranışa sürükleyerek optimum noktalardan uzaklaşılmasına sebep olabilmektedir.

• Sunulan vekil model tekniğinin işlem ve hesaplama gücünün belirli bir yeterlilikte olmadığı, sıkı kısıtlar içeren problemlerin optimizasyon süreçlerinde son derece kullanışlı olduğu görülmüştür. İzin verilen analiz sayısı problemin karmaşıklığına göre belirli bir analiz sayısını geçemiyorsa, seçilen metasezgisel sonuçları arama uzayında iyi noktalara sürükleyecek olmasına rağmen optimal sonuçları keşfedecek kadar çok analize izin verilmediğinden optimalden uzak sonuçlarla karşılaşılabilmektedir. Burada vekil modelin sonuçların iyileştirilmesine, arama uzayında sürecin optimuma yakın noktalara taşınabilmesine yardımcı olduğu çıkarımı yapılabilir.

• Vekil modellerin performanslarının probleme özgü olduğu görülmüştür. Örneğin 15 Çubuklu Kafes Sistem probleminde son popülasyondaki en iyi sonuçlarda çok fazla iyileşme gözlenirken, özellikle 39 Çubuklu Kafes Sistem probleminde son popülasyonda en kötülerin iyileştiği, vekil model tarafından popülasyonun daha optimuma yakın noktalara taşındığı ve verimli bir optimizasyon mekanizmasına yol açtığı çıkarımı yapılabilir. Henüz vekil modelin iyileştirme mekanizması ile alakalı genelleme yapılabilecek yeterli bilgi toplanamamış ve çalışmaların bu kapsamda genişletilmesi gerekmektedir.

• Bu tez çalışmanın sonucu olarak ortalamaların belirli bir yaklaşımla iyileştirilip, iyileştirilemeyeceği ya da problemin karmaşıklığına bağlı olarak optimizasyon sürecinde

iyileştirmenin tam olarak neye göre yapıldığı, en iyi sonuçların hangi parametrelere bağlı olarak iyileştirilip iyileştirilemeyeceğine dair kesin bulgura rastlanmamıştır. Bunun için deneylerin ve etki parametrelerinin ışında deneylerin kapsamının genişletilmesi gerekmektedir. Sunulan vekil, optimum çözümlerin aranmasında olağan metasezgiselin arama mekanizmasında kötüleştirme yapmaz, en kötü ihtimal aynı bırakmaktadır. Bu tez çalışmasında sunulan vekil modelin olağan optimizasyon sürecinde yarattığı destek mekanizması hakkında aynı şekilde genellemeler yapılamaz ancak verimli bir yaklaşım olduğu açıktır.

• Büyük ölçekli yapıları içeren gerçek dünya problemleri üzerinde daha kapsamlı deneyler yapılmadan sunulan vekil modelin tam işlerliği ile alakalı bir şey söylemek mümkün olamamıştır. 10,15,39 Çubuklu Kafes problemlerde iyi performans gösterdiği durumlar veya genel olarak metasezgiselin performansını etkilemediği durumlar olmasına rağmen yazar sunulan vekil modelin daha büyük ölçekli problemlerde daha fazla hissedileceği görüşünde olup ileriki çalışmaların kapsamını bu doğrultuda genişletecektir.

• Vekil YSA için veri toplanması sırasında belirli bir cezadan daha büyük cezaya sahip adayların eğitim setine alınmaması fikri başarılı olmuştur. Optimizasyon sonuçlarını limPen =1'den büyük alınmaması optimizasyon sonucunda en iyi ve en kötü adayların iyileşmesini sağlamıştır.

• Metasezgiselin vekil model olmaksızın optimizasyon sürecini yürüttüğü YSA eğitimi için eğitim verilerinin toplanması aşamasında, metasezgiselin optimuma yakın adayları bulmada gösterdiği optimizasyon davranışının verilerin toplanma davranışıyla paralel olduğu görülmüştür.

• İhlaline izin verilebilen yumuşak kısıtlar ile çalışmalar yapılarak daha büyük ölçekli yapısal optimizasyon problemleri üzerinde sunulan vekil modelin bu durumdaki işlevselliği test edilmeli ve çalışmaların kapsamı arttırılmalıdır. Sunulan vekil model tekniğinin yumuşak kısıtlarda bir hayli etkili olabileceği düşünülmektedir.

# KAYNAKLAR DİZİNİ

Adeli, H. ve Park, H. S. 1995, Optimization of space structures by neural dynamics, *Neural Networks* 8.5, 769–781. DOI: 10.1016/0893-6080(95)00026-v. URL: https://doi.org/10.1016/0893-6080(95)00026-v.

Ahrari, A., Atai, A. A. ve Deb, K. 2015, Simultaneous topology, shape and size optimization of truss structures by fully stressed design based on evolution strategy, *Engineering Optimization* 47.8, 1063–1084. ISSN: 10290273. DOI: 10.1080/0305215X.2014.947972.

Açıkgöz, T. ve Özbaşaran, H. 2020, Numerical experiments on surrogate-assisted optimization of a planar truss, *6th International Congress on Engineering, Architecture and Design*, 928–935.

Ballal, T. M. ve Sher, W. D. 2003, Artificial neural network for the selection of buildable structural systems, *Engineering, Construction and Architectural Management* 10.4, 263–271. ISSN: 09699988. DOI: 10.1108/09699980310489979.

Balling, R. J., Briggs, R. R. ve Gillman, K. 2006, Multiple Optimum Size/Shape/Topology Designs for Skeletal Structures Using a Genetic Algorithm, *Journal of Structural Engineering* 132.7, 1158–1165. ISSN: 0733-9445. DOI: 10.1061/(asce)0733-9445(2006)132:7(1158).

Camp, C. V. ve Bichon, B. J. 2004, Design of Space Trusses Using Ant Colony Optimization, *Journal of Structural Engineering* 130.5, 741–751. ISSN: 0733-9445. DOI: 10.1061/(asce)0733-9445(2004)130:5(741).

Charalampakis, A. E. ve Tsiatas, G. C. 2019, Critical Evaluation of Metaheuristic Algorithms for Weight Minimization of Truss Structures, *Frontiers in Built Environment* 5.October, 1–17. ISSN: 22973362. DOI: 10.3389/fbuil.2019.00113.

Chong, H., YAP, H. J., Tan, S., Yap, K. ve Wong, S. 2021, Advances of metaheuristic algorithms in training neural networks for industrial applications, *Soft Computing*. DOI: 10.1007/s00500-021-05886-z.

# KAYNAKLAR DİZİNİ (devam)

Chou, J.-S., Chong, W. K. ve Bui, D.-K. 2016, Nature-Inspired Metaheuristic Regression System: Programming and Implementation for Civil Engineering Applications, *Journal of Computing in Civil Engineering* 30.5, 04016007. ISSN: 0887-3801. DOI: 10.1061/(asce)cp.1943-5487.0000561.

Deb, K. ve Gulati, S. 2001, Design of truss-structures for minimum weight using genetic algorithms, *Finite Elements in Analysis and Design* 37.5, 447–465. ISSN: 0168874X. DOI: 10.1016/S0168-874X(00)00057-3.

Degertekin, S. O., Lamberti, L. ve Ugur, I. B. 2019, Discrete sizing/layout/topology optimization of truss structures with an advanced Jaya algorithm, *Applied Soft Computing Journal* 79, 363–390. ISSN: 15684946. DOI: 10.1016/j.asoc.2019.03.058. URL: https://doi.org/10.1016/j.asoc.2019.03.058.

Degertekin, S. 2012, Improved harmony search algorithms for sizing optimization of truss structures, *Computers & Structures* 92-93, 229–241. DOI: 10.1016/j.compstruc.2011.10.022. URL: https://doi.org/10.1016/j.compstruc.2011.10.022.

Degertekin, S., Lamberti, L. ve Ugur, I. 2018, Sizing, layout and topology design optimization of truss structures using the Jaya algorithm, *Applied Soft Computing* 70, 903–928. DOI: 10.1016/j.asoc.2017.10.001. URL: https://doi.org/10.1016/j.asoc.2017.10.001.

Demirkan, O., Olceroglu, E., Basdogan, I. ve Özgüven, H. N. 2013, A New Design Approach for Rapid Evaluation of Structural Modifications Using Neural Networks, *Journal of Mechanical Design* 135.2. DOI: 10.1115/1.4023156. URL: https://doi.org/10.1115/1.4023156.

Deng, J. 2006, Structural reliability analysis for implicit performance function using radial basis function network, *International Journal of Solids and Structures* 43.11-12, 3255–3291. DOI: 10.1016/j.ijsolstr.2005.05.055. URL: https://doi.org/10.1016/j.ijsolstr.2005.05.055.

# KAYNAKLAR DİZİNİ (devam)

Dimou, C. K. ve Koumousis, V. K. 2009, Reliability-Based Optimal Design of Truss Structures Using Particle Swarm Optimization, *Journal of Computing in Civil Engineering* 23.2, 100–109. DOI: 10.1061/(asce)0887-3801(2009)23:2(100). URL: https://doi.org/10.1061/(asce)0887-3801(2009)23:2(100).

Du, X., Xu, H. ve Zhu, F. 2021, Understanding the Effect of Hyperparameter Optimization on Machine Learning Models for Structure Design Problems, *Computer-Aided Design* 135, 103013. DOI: 10.1016/j.cad.2021.103013. URL: https://doi.org/10.1016/j.cad.2021.103013.

Echard, B., Gayton, N. ve Lemaire, M. 2011, AK-MCS: An active learning reliability method combining Kriging and Monte Carlo Simulation, *Structural Safety* 33.2, 145–154. DOI: 10.1016/j.strusafe.2011.01.002. URL: https://doi.org/10.1016/j.strusafe.2011.01.002.

Fang, J., Sun, G., Qiu, N., Kim, N. H. ve Li, Q. 2016, On design optimization for structural crashworthiness and its state of the art, *Structural and Multidisciplinary Optimization* 55.3, 1091–1119. DOI: 10.1007/s00158-016-1579-y. URL: https://doi.org/10.1007/s00158-016-1579-y.

Feury, C. ve Geradin, M. 1978, Optimality criteria and mathematical programming in structural weight optimization, *Computers & Structures* 8.1, 7–17. DOI: 10.1016/0045-7949(78)90155-4. URL: https://doi.org/10.1016/0045-7949(78)90155-4.

Forrester, A. I. J., Sobester, A. ve Keane, A. J. 2008, *Engineering Design via Surrogate Modelling - A Practical Guide.* Wiley, I–XVIII, 1–210. ISBN: 978-0-470-06068-1.

Garson, G. D. 1991, Interpreting Neural-Network Connection Weights, *AI Expert* 6.4, 46–51. ISSN: 0888-3785.

Geem, Z. W., Kim, J. H. ve Loganathan, G. 2001, A New Heuristic Optimization Algorithm: Harmony Search, *SIMULATION* 76.2, 60–68. DOI: 10.1177/003754970107600201. URL: https://doi.org/10.1177/003754970107600201.

# KAYNAKLAR DİZİNİ (devam)

Guo, Z. ve Bai, G. 2009, Application of Least Squares Support Vector Machine for Regression to Reliability Analysis, *Chinese Journal of Aeronautics* 22.2, 160–166. DOI: 10.1016/s1000-9361(08)60082-5. URL: https://doi.org/10.1016/s1000-9361(08)60082-5.

Hajela, P. ve Berke, L. 1991, Neurobiological computational models in structural analysis and design, *Computers & Structures* 41.4, 657–667. DOI: 10.1016/0045-7949(91)90178-o. URL: https://doi.org/10.1016/0045-7949(91)90178-o.

Hanna, S. 2007, Inductive machine learning of optimal modular structures: Estimating solutions using support vector machines, *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 21.4, 351–366. DOI: 10.1017/s0890060407000327. URL: https://doi.org/10.1017/s0890060407000327.

Haykin, S. S. 2009, *Neural networks and learning machines*. Third. Upper Saddle River, NJ: Pearson Education.

Holland, J. H. 1975, *Adaptation in Natural and Artificial Systems*. second edition, 1992. Ann Arbor, MI: University of Michigan Press.

Im, J. ve Park, J. 2013, Stochastic structural optimization using particle swarm optimization, surrogate models and Bayesian statistics, *Chinese Journal of Aeronautics* 26.1, 112–121. DOI: 10.1016/j.cja.2012.12.022. URL: https://doi.org/10.1016/j.cja.2012.12.022.

Kallioras, N. A., Kazakis, G. ve Lagaros, N. D. 2020, Accelerated topology optimization by means of deep learning, *Structural and Multidisciplinary Optimization* 62.3, 1185–1212. DOI: 10.1007/s00158-020-02545-z. URL: https://doi.org/10.1007/s00158-020-02545-z.

Karaboga, D. ve Basturk, B. 2008, On the performance of artificial bee colony (ABC) algorithm, *Applied Soft Computing* 8.1, 687–697. DOI: 10.1016/j.asoc.2007.05.007. URL: https://doi.org/10.1016/j.asoc.2007.05.007.

# KAYNAKLAR DİZİNİ (devam)

Kaveh, A., Gholipour, Y. ve Rahami, H. 2008, Optimal Design of Transmission Towers Using Genetic Algorithm and Neural Networks, *International Journal of Space Structures* 23.1, 1–19. DOI: 10.1260/026635108785342073. URL: https://doi.org/10.1260/026635108785342073.

Kaveh, A. ve Talatahari, S. 2009, Size optimization of space trusses using Big Bang–Big Crunch algorithm, *Computers & Structures* 87.17-18, 1129–1140. DOI: 10.1016/j.compstruc.2009.04.011. URL: https://doi.org/10.1016/j.compstruc.2009.04.011.

Kaymaz, I. 2005, Application of kriging method to structural reliability problems, *Structural Safety* 27.2, 133–151. DOI: 10.1016/j.strusafe.2004.09.001. URL: https://doi.org/10.1016/j.strusafe.2004.09.001.

Kennedy, J. ve Eberhart, R. 1995, Particle swarm optimization, *Proceedings of ICNN'95 - International Conference on Neural Networks*. **volume** 4, 1942–1948 vol.4. DOI: 10.1109/ICNN.1995.488968.

Kingma, D. P. ve Ba, J. 2017, *Adam: A Method for Stochastic Optimization*. arXiv: 1412.6980 [cs.LG].

Kirkpatrick, S., Gelatt, C. D. ve Vecchi, M. P. 1983, Optimization by Simulated Annealing, *Science* 220.4598, 671–680. ISSN: 0036-8075. DOI: 10.1126/science.220.4598.671. eprint: https://science.sciencemag.org/content/220/4598/671.full.pdf. URL: https://science.sciencemag.org/content/220/4598/671.

Krempser, E., Bernardino, H. S., Barbosa, H. J. ve Lemonge, A. C. 2012, Differential evolution assisted by surrogate models for structural optimization problems, *Civil-Comp Proceedings* 100, 1–19. ISSN: 17593433. DOI: 10.4203/ccp.100.49.

Lagaros, N. D. ve Papadrakakis, M. 2004, Learning improvement of neural networks used in structural optimization, *Advances in Engineering Software* 35.1, 9–25. DOI: 10.1016/s0965-9978(03)00112-1. URL: https://doi.org/10.1016/s0965-9978(03)00112-1.

# KAYNAKLAR DİZİNİ (devam)

Lamberti, L. 2008, An efficient simulated annealing algorithm for design optimization of truss structures, *Computers & Structures* 86.19-20, 1936–1953. DOI: 10.1016/j.compstruc.2008.02.004. URL: https://doi.org/10.1016/j.compstruc.2008.02.004.

Lee, J., Jeong, H. ve Kang, S. 2007, Derivative and GA-based methods in metamodeling of back-propagation neural networks for constrained approximate optimization, *Structural and Multidisciplinary Optimization* 35.1, 29–40. DOI: 10.1007/s00158-007-0110-x. URL: https://doi.org/10.1007/s00158-007-0110-x.

Lee, K. S. ve Geem, Z. W. 2004, A new structural optimization method based on the harmony search algorithm, *Computers & Structures* 82.9-10, 781–798. DOI: 10.1016/j.compstruc.2004.01.002. URL: https://doi.org/10.1016/j.compstruc.2004.01.002.

Lee, S., Ha, J., Zokhirova, M., Moon, H. ve Lee, J. 2017, Background Information of Deep Learning for Structural Engineering, *Archives of Computational Methods in Engineering* 25.1, 121–129. DOI: 10.1007/s11831-017-9237-0. URL: https://doi.org/10.1007/s11831-017-9237-0.

Luh, G. C. ve Lin, C. Y. 2011, Optimal design of truss-structures using particle swarm optimization, *Computers and Structures* 89.23-24, 2221–2232. ISSN: 00457949. DOI: 10.1016/j.compstruc.2011.08.013. URL: http://dx.doi.org/10.1016/j.compstruc.2011.08.013.

Martini, K. 2011, Harmony Search Method for Multimodal Size, Shape, and Topology Optimization of Structural Frameworks, *Journal of Structural Engineering* 137.11, 1332–1339. ISSN: 0733-9445. DOI: 10.1061/(asce)st.1943-541x.0000378.

Messner, M. C. 2019, Convolutional Neural Network Surrogate Models for the Mechanical Properties of Periodic Structures, *Journal of Mechanical Design* 142.2. DOI: 10.1115/1.4045040. URL: https://doi.org/10.1115/1.4045040.

# KAYNAKLAR DİZİNİ (devam)

Miguel, L. F. F., Lopez, R. H. ve Miguel, L. F. F. 2013, Multimodal size, shape, and topology optimisation of truss structures using the Firefly algorithm, *Advances in Engineering Software* 56, 23–37. ISSN: 09659978. DOI: 10.1016/j.advengsoft.2012.11.006. URL: http://dx.doi.org/10.1016/j.advengsoft.2012.11.006https://linkinghub.elsevier.com/retrieve/pii/S0965997812001536.

Mortazavi, A. ve Toğan, V. 2016, Simultaneous size, shape, and topology optimization of truss structures using integrated particle swarm optimizer, *Structural and Multidisciplinary Optimization* 54.4, 715–736. DOI: 10.1007/s00158-016-1449-7. URL: https://doi.org/10.1007/s00158-016-1449-7.

Mote, H ve Kumar, S. R. S. 2019, Use of Artificial Neural Network for Initial design of steel structures, *IOP Conference Series: Materials Science and Engineering* 660, 012064. DOI: 10.1088/1757-899x/660/1/012064. URL: https://doi.org/10.1088/1757-899x/660/1/012064.

Müller, T. E. ve Klashorst, E. van der. 2017, A quantitative comparison between size, shape, topology and simultaneous optimization for truss structures, *Latin American Journal of Solids and Structures* 14.12, 2221–2242. ISSN: 16797825. DOI: 10.1590/1679-78253900.

Nguyen, T.-H. ve Vu, A.-T. 2020, Using Neural Networks as Surrogate Models in Differential Evolution Optimization of Truss Structures, *Computational Collective Intelligence*. Springer International Publishing, 152–163. DOI: 10.1007/978-3-030-63007-2_12. URL: https://doi.org/10.1007/978-3-030-63007-2_12.

Niutta, C. B., Wehrle, E. J., Duddeck, F. ve Belingardi, G. 2018, Surrogate modeling in design optimization of structures with discontinuous responses, *Structural and Multidisciplinary Optimization* 57.5, 1857–1869. DOI: 10.1007/s00158-018-1958-7. URL: https://doi.org/10.1007/s00158-018-1958-7.

# KAYNAKLAR DİZİNİ (devam)

Noilublao, N. ve Bureerat, S. 2013, Simultaneous Topology, Shape, and Sizing Optimisation of Plane Trusses with Adaptive Ground Finite Elements Using MOEAs, *Mathematical Problems in Engineering* 2013, 1–9. DOI: 10.1155/2013/838102. URL: https://doi.org/10.1155/2013/838102.

Nourbakhsh, M., Irizarry, J. ve Haymaker, J. 2018, Generalizable surrogate model features to approximate stress in 3D trusses, *Engineering Applications of Artificial Intelligence* 71, 15–27. DOI: 10.1016/j.engappai.2018.01.006. URL: https://doi.org/10.1016/j.engappai.2018.01.006.

Oliphant, T. E. 2007, Python for scientific computing, *Computing in Science and Engineering* 9.3, 10–20. ISSN: 15219615. DOI: 10.1109/MCSE.2007.58. URL: http://ieeexplore.ieee.org/document/4160250/.

Ong, Y. S., Nair, P. B. ve Keane, A. J. 2003, Evolutionary optimization of computationally expensive problems via surrogate modeling, *AIAA Journal* 41.4, 687–696. URL: https://eprints.soton.ac.uk/22421/.

Özbaşaran, H. 2021, Overall Performance Of Artificial Neural Networks As Surrogate Models For Structural Optimization, *7th International Congress on Engineering, Architecture and Design*, 638–645.

Papadrakakis, M. ve Lagaros, N. D. 2003, Soft computing methodologies for structural optimization, *Applied Soft Computing* 3.3, 283–300. DOI: 10.1016/s1568-4946(03)00040-1. URL: https://doi.org/10.1016/s1568-4946(03)00040-1.

Papadrakakis, M., Lagaros, N. D. ve Tsompanakis, Y. 1998, Structural optimization using evolution strategies and neural networks, *Computer Methods in Applied Mechanics and Engineering* 156.1-4, 309–333. DOI: 10.1016/s0045-7825(97)00215-6. URL: https://doi.org/10.1016/s0045-7825(97)00215-6.

Park, H. S. ve Adeli, H. 1997, Distributed Neural Dynamics Algorithms for Optimization of Large Steel Structures, *Journal of Structural Engineering* 123.7, 880–888. DOI: 10.

# KAYNAKLAR DİZİNİ (devam)

1061/(asce)0733-9445(1997)123:7(880). URL: https://doi.org/10.1061/(asce)0733-9445(1997)123:7(880).

Perez, R. ve Behdinan, K. 2007, Particle swarm approach for structural design optimization, *Computers & Structures* 85.19-20, 1579–1588. DOI: 10.1016/j.compstruc.2006.10.013. URL: https://doi.org/10.1016/j.compstruc.2006.10.013.

Pholdee, N. ve Bureerat, S. 2014, Comparative performance of meta-heuristic algorithms for mass minimisation of trusses with dynamic constraints, *Advances in Engineering Software* 75, 1–13. ISSN: 18735339. DOI: 10.1016/j.advengsoft.2014.04.005. URL: http://dx.doi.org/10.1016/j.advengsoft.2014.04.005.

Prayogo, D., Cheng, M. Y., Wu, Y. W. ve Tran, D. H. 2020, Combining machine learning models via adaptive ensemble weighting for prediction of shear capacity of reinforced-concrete deep beams, *Engineering with Computers* 36.3, 1135–1153. ISSN: 14355663. DOI: 10.1007/s00366-019-00753-w. URL: https://doi.org/10.1007/s00366-019-00753-w.

Rajan, S. D. 1995, Sizing, Shape, and Topology Design Optimization of Trusses Using Genetic Algorithm, *Journal of Structural Engineering* 121.10, 1480–1487. ISSN: 0733-9445. DOI: 10.1061/(ASCE)0733-9445(1995)121:10(1480). URL: http://ascelibrary.org/doi/10.1061/%28ASCE%290733-9445%281995%29121%3A10%281480%29.

Rogers, J. L. 1994, Simulating Structural Analysis with Neural Network, *Journal of Computing in Civil Engineering* 8.2, 252–265. DOI: 10.1061/(asce)0887-3801(1994)8:2(252). URL: https://doi.org/10.1061/(asce)0887-3801(1994)8:2(252).

Saka, M., Hasançebi, O. ve Geem, Z. 2016, Metaheuristics in structural optimization and discussions on harmony search algorithm, *Swarm and Evolutionary Computation* 28, 88–97. DOI: 10.1016/j.swevo.2016.01.005. URL: https://doi.org/10.1016/j.swevo.2016.01.005.

# KAYNAKLAR DİZİNİ (devam)

Samuel, A. L. 2000, Some studies in machine learning using the game of checkers, *IBM Journal of Research and Development* 44.1.2, 206–226. DOI: 10.1147/rd.441.0206. URL: https://doi.org/10.1147/rd.441.0206.

Santana Gomes, W. J. de ve Beck, A. T. 2013, Global structural optimization considering expected consequences of failure and using ANN surrogates, *Computers & Structures* 126, 56–68. DOI: 10.1016/j.compstruc.2012.10.013. URL: https://doi.org/10.1016/j.compstruc.2012.10.013.

Savsani, V. J., Tejani, G. G. ve Patel, V. K. 2016, Truss topology optimization with static and dynamic constraints using modified subpopulation teaching–learning-based optimization, *Engineering Optimization* 48.11, 1990–2006. ISSN: 10290273. DOI: 10.1080/0305215X.2016.1150468.

Sharifi, Y., Moghbeli, A., Hosseinpour, M. ve Sharifi, H. 2019, Neural networks for lateral torsional buckling strength assessment of cellular steel I-beams, *Advances in Structural Engineering* 22.9, 2192–2202. DOI: 10.1177/1369433219836176. URL: https://doi.org/10.1177/1369433219836176.

Sonmez, M. 2011, Artificial Bee Colony algorithm for optimization of truss structures, *Applied Soft Computing* 11.2, 2406–2418. DOI: 10.1016/j.asoc.2010.09.003. URL: https://doi.org/10.1016/j.asoc.2010.09.003.

— 2018, Performance Comparison of Metaheuristic Algorithms for the Optimal Design of Space Trusses, *Arabian Journal for Science and Engineering* 43.10, 5265–5281. ISSN: 21914281. DOI: 10.1007/s13369-018-3080-y. URL: https://doi.org/10.1007/s13369-018-3080-y.

Souza, R. R., Fadel Miguel, L. F., Lopez, R. H., Torii, A. J. ve Miguel, L. F. 2016, A backtracking search algorithm for the simultaneous size, shape and topology optimization of trusses, *Latin American Journal of Solids and Structures* 13.15, 2622–2651. ISSN: 16797825. DOI: 10.1590/1679-78253101.

# KAYNAKLAR DİZİNİ (devam)

Srinivas, V. ve Ramanjaneyulu, K. 2007, An integrated approach for optimum design of bridge decks using genetic algorithms and artificial neural networks, *Advances in Engineering Software* 38.7, 475–487. DOI: 10.1016/j.advengsoft.2006.09.016. URL: https://doi.org/10.1016/j.advengsoft.2006.09.016.

Su, G., Peng, L. ve Hu, L. 2017, A Gaussian process-based dynamic surrogate model for complex engineering structural reliability analysis, *Structural Safety* 68, 97–109. DOI: 10.1016/j.strusafe.2017.06.003. URL: https://doi.org/10.1016/j.strusafe.2017.06.003.

Swingler, K. 1996, *Applying neural networks - a practical guide.* Academic Press, I–XVIII, 1–303. ISBN: 978-0-12-679170-9.

Talatahari, S., Kheirollahi, M., Farahmandpour, C. ve Gandomi, A. H. 2012, A multi-stage particle swarm for optimum design of truss structures, *Neural Computing and Applications* 23.5, 1297–1309. DOI: 10.1007/s00521-012-1072-5. URL: https://doi.org/10.1007/s00521-012-1072-5.

Talatahari, S., Goodarzimehr, V. ve Taghizadieh, N. 2020, Hybrid teaching-learning-based optimization and harmony search for optimum design of space trusses, *Journal of Optimization in Industrial Engineering* 13.1, 177–194. ISSN: 24233935. DOI: 10.22094/JOIE.2019.1866904.1649.

Tang, W., Tong, L. ve Gu, Y. 2005, Improved genetic algorithm for design optimization of truss structures with sizing, shape and topology variables, *International Journal for Numerical Methods in Engineering* 62.13, 1737–1762. ISSN: 00295981. DOI: 10.1002/nme.1244.

Tashakori, A. ve Adeli, H. 2002, Optimum design of cold-formed steel space structures using neural dynamics model, *Journal of Constructional Steel Research* 58.12, 1545–1566. DOI: 10.1016/s0143-974x(01)00105-5. URL: https://doi.org/10.1016/s0143-974x(01)00105-5.

# KAYNAKLAR DİZİNİ (devam)

Techasen, T., Wansasueb, K., Panagant, N., Pholdee, N. ve Bureerat, S. 2018, Multiobjective Simultaneous Topology, Shape and Sizing Optimization of Trusses Using Evolutionary Optimizers, *IOP Conference Series: Materials Science and Engineering* 370.1. ISSN: 1757899X. DOI: 10.1088/1757-899X/370/1/012029.

Tejani, G. G., Savsani, V. J., Patel, V. K. ve Savsani, P. V. 2017, Size, shape, and topology optimization of planar and space trusses using mutation-based improved metaheuristics, *Journal of Computational Design and Engineering* 5.2, 198–214. DOI: 10.1016/j.jcde.2017.10.001. URL: https://doi.org/10.1016/j.jcde.2017.10.001.

Venkata Rao, R. 2016, Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems, *International Journal of Industrial Engineering Computations* 7.1, 19–34. ISSN: 19232934. DOI: 10.5267/j.ijiec.2015.8.004.

Wang, D, Zhang, W. ve Jiang, J. 2002, Truss shape optimization with multiple displacement constraints, *Computer Methods in Applied Mechanics and Engineering* 191.33, 3597–3612. DOI: 10.1016/s0045-7825(02)00297-9. URL: https://doi.org/10.1016/s0045-7825(02)00297-9.

Wu, C. Y. ve Tseng, K. Y. 2010, Truss structure optimization using adaptive multi-population differential evolution, *Structural and Multidisciplinary Optimization* 42.4, 575–590. ISSN: 1615147X. DOI: 10.1007/s00158-010-0507-9.

Wu, S. J. ve Chow, P. T. 1995, Integrated discrete and configuration optimization of trusses using genetic algorithms, *Computers and Structures* 55.4, 695–702. ISSN: 00457949. DOI: 10.1016/0045-7949(94)00426-4.

Yu, M., Li, X. ve Liang, J. 2019, A dynamic surrogate-assisted evolutionary algorithm framework for expensive structural optimization, *Structural and Multidisciplinary Optimization* 61.2, 711–729. DOI: 10.1007/s00158-019-02391-8. URL: https://doi.org/10.1007/s00158-019-02391-8.

# KAYNAKLAR DİZİNİ (devam)

Yu, Y., Hur, T., Jung, J. ve Jang, I. G. 2018, Deep learning for determining a near-optimal topological design without any iteration, *Structural and Multidisciplinary Optimization* 59.3, 787–799. DOI: 10.1007/s00158-018-2101-5. URL: https://doi.org/10.1007/s00158-018-2101-5.

Zhou, Z., Ong, Y. S., Nair, P. B., Keane, A. J. ve Lum, K. Y. 2007, Combining Global and Local Surrogate Models to Accelerate Evolutionary Optimization, *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)* 37.1, 66–76. DOI: 10.1109/tsmcc.2005.855506. URL: https://doi.org/10.1109/tsmcc.2005.855506.

Zhu, M., McKenna, F. ve Scott, M. H. 2018, OpenSeesPy: Python library for the OpenSees finite element framework, *SoftwareX* 7, 6–11. ISSN: 23527110. DOI: 10.1016/j.softx.2017.10.009.