# YAPISAL HASAR TESPİTİNDE İKİ ÖNEMLİ EVRİMSEL OPTİMİZASYON YÖNTEMİNİN KARŞILAŞTIRILMASI

**Hakan GÖKDAĞ[1], Ali Rıza YILDIZ[2]**
[1]hakan.gokdag@btu.edu.tr Bursa Teknik Üniversitesi, Makine Mühendisliği Bölümü, 16190 Bursa
[2]aliriza.yildiz@btu.edu.tr Bursa Teknik Üniversitesi, Makine Mühendisliği Bölümü, 16190 Bursa

## ÖZET

Bu çalışmada iyi bilinen iki evrimsel temelli optimizasyon algoritması bir titreşim temelli hasar tespiti problemine uygulanmıştır. Bu bağlamda yapının sonlu eleman (SE) modeli oluşturulmuş, bulunan doğal frekans ve titreşim modları ile modal esneklik hesaplanmıştır. Hasarlı haldeki yapıya ait frekans ve modlarla hesaplanan esneklik ile herhangi bir hasar konfigürasyonu için yapının SE modeliyle hesaplanan esneklik farkıyla bir amaç fonksiyonu tanımlanmıştır. Bunu minimize etmek suretiyle yapıdaki hasar konfigürasyonu belirlenmiştir. Bu amaçla, daraltma çarpanı kullanan parçacık sürüsü optimizasyonu (CPSO) ve yapay arı kolonisi (ABC) algoritmaları kullanılmıştır. Benchmark problemlerde ABC iyi sonuç vermesine rağmen hasar tespiti uygulamasında CPSO'nun daha iyi olduğu görülmüştür.

**Anahtar Sözcükler:** Hasar tespiti, Titreşim, Parçacık Sürüsü Optimizasyonu, Yapay Arı Kolonisi Optimizasyonu

## ABSTRACT

In this work, two well-known evolutionary based algorithms are applied to a vibration based damage detection problem. In this regard, the finite element model of the structure is composed, and then modal flexibility is obtained by the computed natural frequencies and vibration modes. The modal flexibility obtained by the natural frequencies and vibration modes of the damaged structure is subtracted from the one obtained by the finite element model of the structure for a damage configuration, so that an objective function is defined. To minimize this, the particle swarm optimization with constriction factor, i.e. CPSO, and the artificial bee colony optimization are employed. Although ABC is superior according to the simulations with benchmark problems, the CPSO is observed to be better in the damage detection application considered in this work.

**Keywords:** Damage detection, Vibration, Partical Swarm Optimization, Artificial Bee Colony Optimization

## 1. GİRİŞ

Yapılarda erken hasar tespiti emniyet ve maliyet açısından önemli bir konudur. Bu amaçla çok sayıda çalışma yapılmış ve çeşitli hasar tespit yöntemleri geliştirilmiştir. Bunlar genel olarak tahribatlı ve tahribatsız olarak sınıflandırılırlar. Tahribatsız yöntemler de lokal ve global olarak ikiye ayrılır. Lokal yöntemler içinde ultrasonik, termal kızılötesi, radyografi, Eddy akımları gibi testler yaygın uygulamaya sahiptir. Bunların ortak bir yetersizliği karmaşık ve büyük çaplı yapılara uygulanmalarının zor olmasıdır. Diğer bir dezavantaj hasar bölgesinin önceden bilinmesini

gerektirmeleridir. Bu gibi kısıtlar global özellikteki yöntemlerin gelişmesine katkı sağlamıştır. Titreşim özelliklerindeki değişimi esas alan yöntemler (kısaca titreşim temelli yöntemler) global özelliktedirler [1-3]. Titreşim temelli yöntemlerin amacı işletme yükleri sebebiyle yapıda zamanla meydana gelen lokal katılık kayıplarının olup olmadığını, varsa yerini ve derecesini belirlemektir. Bunun için hasar sebebiyle doğal frekanslar, sönüm oranları, titreşim modları ve bunlardan türetilen indislerdeki (modal esneklik, modal şekil değiştirme enerjisi, frekans cevabı fonksiyonu vs) değişimlerden yararlanılır. Bu alanda geliştirilen bir grup yöntem model güncelleme [1] adıyla anılır. Böyle bir yöntemin temeli hasar tespiti işlemini bir optimizasyon problemine dönüştürüp çözerek hasar bilgisini elde etmektir. Amaç fonksiyonu, yukarıda bahsedilen titreşim özelliklerinin hasarlı haldeki değerleri ve yapının matematik modeli ile hesaplanan değerlerinin farkıyla oluşturulur. Tasarım değişkenleri de hasarın tanımına göre değişir; hasar çatlak şeklinde tanımlanmışsa bunun yeri ve derinliği, eleman katılığında azalma olarak modellenmişse bu azalma yüzdesi tasarım değişkenidir. Amaç fonksiyonu belli kısıtlar altında minimize edilerek hasarlı duruma karşılık gelen tasarım değişkenleri bulunur, böylece hasar yeri ve derecesi yaklaşık olarak ortaya çıkarılır [1,4,5].

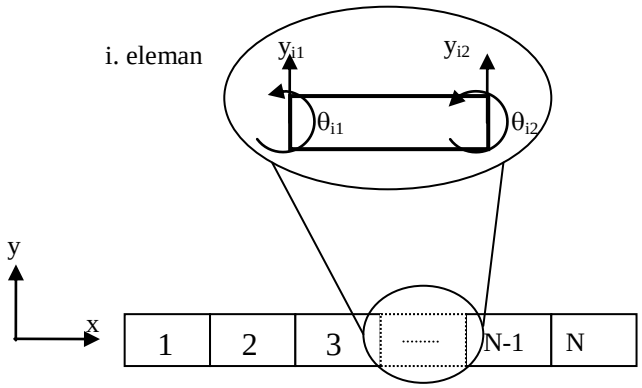
Son yıllarda optimizasyon problemlerinin çözümünde, türev hesaplamayı gerektirmemesi ve lokal minimuma yakalanma riski az olduğundan, genellikle evrimsel yöntemler tercih edilir. Konuyla ilgili literatüre bakıldığında çoğunlukla genetik algoritmaların (GA) kullanıldığı görülür [6-9]. Bunun yanında, parçacık sürüsü optimizasyonu (particle swarm optimization - PSO) ve yapay arı kolonisi optimizasyonu (artificial bee colony - ABC) gibi daha güncel yaklaşımların kullanıldığı çalışmaların son zamanlarda literatürde yer aldığını görüyoruz. Örneğin, Begambre ve Laier [10] PSO ve Simplex algoritmalarını hibritleştirerek bir yöntem geliştirmişler ve bunu iki ucu serbest bir kiriş ve 10 uzuvlu kafes sistemi gibi yapılarda hasar tespiti problemlerine tatbik etmişlerdir. Amaç fonksiyonu yapının matematiksel modeliyle hesaplanan ve hasarlı durumdaki frekans cevabı fonksiyonlarının farkıyla tanımlanmıştır. Seyedpoor [11] MDLAC (multiple damage location assurance criterion) ve modal şekil değiştirme indisi gibi titreşim temelli hasar indislerini kullanarak bir hasar tespit yaklaşımı sunmuştur. Bu indislerden biri ile hasarlı elemanlar yaklaşık olarak tespit edilmekte, daha sonra diğeri ile tanımlı amaç fonksiyonu PSO algoritması ile minimize edilerek hasarlı elemanlar ve hasar dereceleri daha hassas bir şekilde belirlenmiştir. Öte yandan, Moradi ve ark. [12] çatlaklı bir kirişin matematik modeli ile hesaplanan ilk birkaç doğal frekansı ve bunların hasarlı haldeki değerleri farkını kullanarak bir amaç fonksiyonu tanımlamıştır. Bunu PSO ve arı kolonisi yaklaşımları ile minimize ederek çatlak yeri ve derinliğini tespit etmişler, böylece PSO ve arı kolonisi algoritmalarının performanslarını karşılaştırmışlardır. Gökdağ ve Yıldız [5] sıkça kullanılan dört farklı modal parametre temelli indisin performansını karşılaştırmışlardır. Çözücü algoritma olarak PSO kullanılmış, esneklik temelli indisin diğerlerine nispetle daha iyi olduğu görülmüştür.

Bu çalışma önceki çalışmanın [5] devamı niteliğindedir. Köprü türü bir yapının sonlu eleman modelinden yararlanılarak PSO ve ABC algoritmalarının performansı karşılaştırılmıştır. Amaç fonksiyonu olarak modal esneklik kullanılmıştır. Klasik PSO algoritması yerine daraltma çarpanı kullanan başka bir PSO algoritması [13] kullanılmıştır. Hesaplamalar MATLAB ortamında gerçekleştirilmiştir. PSO ile ilgili kodlar tarafımızdan yazılmış, MATLAB ortamında yazılan ABC algoritmasının v2 sürümü ise ilgili siteden [14] indirilmiştir.

# 2. TEORİ

## 2.1 Formülasyon

İncelediğimiz yapı bir kiriş olup bunun sonlu eleman modeli Şekil 1'de gösterilmiştir. Buna göre, N elemana bölünmüş bir kirişte bir elemanın iki düğüm noktası ve her düğüm noktasında biri dönme diğer öteleme olmak üzere iki hareket serbestliği vardır.




Şekil 1. Kirişin sonlu eleman modeli

Eleman uzunluğu $L_e$ olsun. Timoshenko modeli dikkate alınırsa elemanın kütle ve katılık matrisleri aşağıdaki gibi hesaplanır [5]:

$$\mathbf{m_e} = \rho A\int_0^{L_e} \mathbf{L}\mathbf{L}^{\mathrm{T}}\mathrm{d}x + \rho I\int_0^{L_e} \mathbf{S}\mathbf{S}^{\mathrm{T}}\mathrm{d}x\,, \quad \mathbf{k_e} = EI\int_0^{L_e} \mathbf{S'}\mathbf{S'}^{\mathrm{T}}\mathrm{d}x + kGA\mathbf{V}\mathbf{V}^{\mathrm{T}} \qquad (1)$$

Burada $\rho$, *A*, *E*, *G*, *I* ve *k* sırasıyla yoğunluk, kesit alanı, elastisite modülü, kayma modülü, alan atalet momenti ve kesit şekil faktörüdür. **L**, **S** ve **V** vektörleri de aşağıdaki gibi tanımlıdır:

$$\mathbf{L} = \left[\left(1-x/L_e\right) \quad \left(x/2-x^2/2L_e\right) \quad x/L_e \quad \left(x^2/2L_e-x/2\right)\right]^{\mathrm{T}},\ \mathbf{S} = \left[0 \quad \left(1-x/L_e\right) \quad 0 \quad x/L_e\right]^{\mathrm{T}}$$

$$\mathbf{S'} = \mathrm{d}\mathbf{S}/\mathrm{d}x\,,\ \mathbf{V} = \left[-1/L_e \quad -1/2 \quad 1/L_e \quad 1/2\right]^{\mathrm{T}}\,,\ \text{T: Transpoz}$$

Her elemanın kütle ve katılık matrisleri uygun bir şekilde birleştirilir ve sınır şartları uygulanırsa toplam kütle (**M**) ve katılık (**K**) matrisleri elde edilir. Sönüm ihmal edildiğinde özdeğer problemi aşağıdaki gibi derlenir:

$$\left(\mathbf{K}-\omega^2\mathbf{M}\right)\mathbf{X} = \mathbf{0} \qquad (2)$$

Bunun çözümü ile i. doğal frekans ($\omega_i$) ve mukabil mod şekli ($\mathbf{X}_i$) bulunur. Kiriş N elemana bölünürse N+1 adet düğüm noktası oluşur. Her düğüm noktasında 2 serbestlik olduğundan, hesaplanabilecek nicelik sayısı en fazla 2(N+1) dir; i=1,2,…,2(N+1). Her elemanın katılık matrisinin $(1-g_i)$ gibi bir reel sayı ile çarpıldığını varsayalım. Burada $0 \le g_i < 1$ olsun. Şu halde $g_i = 0$ ilgili elemanın hasarsız olduğunu gösterir. $g_i = 1$ halinde eleman tamamen hasarlanmış demektir. Yani $g_i$ sayısı eleman katılığındaki azalmayı ifade eder. Şimdi, hesaplanan $N_M$ adet doğal frekans ve mukabil mod şekilleri ile bunların hasarlı haldeki karşılıklarını kullanarak aşağıdaki gibi bir amaç fonksiyonu oluşturalım [15]:

$$F(\mathbf{g}) = \left\| \mathbf{X}^c\mathbf{\Lambda}^c\mathbf{X}^{c^{\mathrm{T}}} - \mathbf{X}^d\mathbf{\Lambda}^d\mathbf{X}^{d^{\mathrm{T}}} \right\| \qquad \mathbf{0} \le \boldsymbol{g}_i < \mathbf{1} \qquad (3)$$

Burada **X** i. sütunu i. mod şeklini içeren modal matris (i=1,2,…$N_M$), $\mathbf{\Lambda}$ köşegen elemanları doğal frekanslar olan köşegen matris, üst "c" ilgili niceliğin yapının sonlu eleman modeli ile hesaplandığını, "d" ise hasarlı halde kaydedilen değer olduğunu gösterir. **g** vektörü eleman katılıklarındaki azalmayı ifade eden $g_i$ sayılarını içerir. Şu halde (3) eşitliği yapının hasarlı haldeki modal esnekliği $\mathbf{X}^d\mathbf{\Lambda}^d\mathbf{X}^{d^{\mathrm{T}}}$ ve herhangi bir hasar konfigürasyonu için sonlu eleman modeli ile hesaplanan modal esneklik farkının matris normunu ifade eder. Bu amaç fonksiyonunu minimize

eden $\mathbf{g}$ vektörü aranan hasar konfigürasyonunu verir. $\mathbf{g}$ tasarım değişkenleri vektörünün sınırları ise $\mathbf{0} \leq \mathbf{g}_i < \mathbf{1}$ şeklindedir. (3) ile verilen optimizasyon problemini çözmek için PSO ve ABC algoritmaları kullanılacaktır. Bunlarla ilgili ayrıntılar sonraki başlıklarda verilmiştir.

## 2.2 Optimizasyon Algoritmaları

### 2.2.1 PSO

PSO algoritması kuş ve balık sürülerinin yiyecek ararken gösterdikleri toplu davranışlardan esinlenerek geliştirilmiştir [16]. PSO algoritmasında N adet parçacıktan oluşmuş bir sürü mevcuttur. Parçacıklar tasarım uzayındaki muhtemel çözüm noktalarını temsil ederler. Parçacıkların ilk değerleri ve hızları tasarım uzayında rastgele örnekleme ile elde edilir. Her iterasyonda parçacığın konumu bir önceki iterasyondaki kendi en iyi noktası $p_{ij}^k$ ve sürünün en iyi noktası $p_{gi}^k$ ile güncellenerek ayarlanır. Bu işlem aşağıdaki eşitliklerle gerçekleştirilir:

$$\begin{aligned} v_{ij}^{k+1} &= v_{ij}^k + c_1 R_1 (p_{ij}^k - x_{ij}^k) + c_2 R_2 (p_{gj}^k - x_{ij}^k) \\ x_{ij}^{k+1} &= x_{ij}^k + v_{ij}^{k+1}, \quad i = 1,2,..,N, \;\; j = 1,2,...,m, \;\; k = 1,2,...,K_{\max} \end{aligned} \tag{4}$$

Burada $v_{ij}^k$ k. iterasyonda i. parçacığın j. boyutunun hızı, aynı şekilde $x$ konumu, $R_1$ ve $R_2$ (0,1) aralığında üniform dağılımlı random sayılardır. $c_1$ ve $c_2$ sırasıyla bilişsel ve sosyal ivme katsayıları olup parçacıkların lokal en iyi ve sürünün en iyi konumlarına doğru hızlarını düzenler. Clerc ve Kennedy [13] PSO algoritmasında hızların iyi ayarlanamamasından kaynaklanan "sürü patlaması" (swarm explosion) denilen olumsuzluğu bertaraf etmek ve algoritmanın yakınsama hızını arttırmak üzere daraltma (constriction) çarpanı ($\chi$) içeren yeni bir yaklaşım geliştirmiştir. Buna göre parçacıkların hızları aşağıdaki gibi güncellenir.

$$v_{ij}^{k+1} = \chi \left( v_{ij}^k + c_1 R_1 (p_{ij}^k - x_{ij}^k) + c_2 R_2 (p_{gj}^k - x_{ij}^k) \right) \tag{5}$$

Eşitlikteki sabitler arasındaki ilişkiler [13]: $\chi = 2\left( \left| 2 - \varphi - \sqrt{\varphi^2 - 4\varphi} \right| \right)^{-1}$, $\varphi = c_1 + c_2$ ve $\varphi > 4$

Çalışmamızda PSO'nun bu versiyonunu kullanacağız ve "constriction" ismine atfen bunu CPSO ile göstereceğiz. Burada ilgili sabitler $c_1 = c_2 = 2.05$ olarak seçilmiş, $\chi = 0.7298$ hesaplanmıştır.

### 2.2.2 ABC

ABC algoritması Karaboğa [17] tarafından geliştirilen popülasyon tabanlı bir algoritmadır. Mutasyon, çaprazlama gibi harici parametrelere ihtiyaç duymayan, basit, esnek ve PSO'ya kıyasla daha az parametreli bir optimizasyon yöntemidir. Bu algoritmada yapay arılar söz konusu olup bunlar üç gruba ayrılır: 1-İşçi arılar, 2-Gözcü arılar, 3- Kâşif arılar. Kolonideki arıların yarısı işçi diğer yarısı ise gözcü arı olarak seçilir. Her bir nektar kaynağı için bir işçi arı bulunur. Bu durumda işçi arıları sayısı nektar kaynağı sayısına eşittir [18]. Algoritmanın başlangıcında yine ilk değerler rastgele olarak üretilir. Her iterasyonda işçi arılar kendi yiyecek kaynaklarındaki nektar miktarını arttırmaya çalışırlar. Sonra her gözcü arı nektar miktarına göre yiyecek kaynağı seçer. Yiyecek kaynağı tükenen işçi arı kâşif arı olur. Bu kâşif arı yeni yiyecek kaynakları aramaya başlar. Yiyecek kaynağının konumu optimizasyon probleminde bir çözüme karşılık gelir. Nektar miktarı ise çözümün fitness değerini ifade eder. Algoritmanın işlem adımları aşağıdaki gibidir [19]:

1) Popülasyon büyüklüğü N olmak üzere tasarım uzayında rastgele örnekleme yaparak N/2 adet çözüm noktası oluştur.

2) Her $x_i$ çözümü için (6) eşitliğini kullanarak bir $v_i$ noktası tanımla.
3) Çözümlerin olasılık değerlerini (7) eşitliği ile hesapla.
4) Rulet tekeri seçim kuralına göre gözcü arıları yiyecek kaynaklarına konumlandır ve 2. adımdaki işlemleri yap.
5) -Varsa şayet- terk edilen çözümleri belirle ve bunların yerine rastgele sayılar üreterek yeni aday noktalar oluştur.
6) En iyi çözümü kaydet.
7) 2. ve 6. adımlar arasındaki işlemleri tekrarla belli bir iterasyon sayısı için tekrarla.

$$v_{ij} = x_{ij} + \varphi_{ij}(x_{ij} - x_{kj}) \qquad \text{i=1,2,... N/2} \qquad (6)$$

$$P_i = \frac{fit_i}{\sum_{n=1}^{N/2} fit_n} \qquad (7)$$

burada $\varphi_{ij}$ (-1,1) arasında üniform dağılımlı random sayı, k ve j rastgele seçilmiş birbirinden farklı indislerdir. $fit$ çözüm noktalarının fitness değeri olup aşağıdaki gibi hesaplanır:

$$fit_i = \begin{cases} \dfrac{1}{1+F_i} & F_i \le 0 \ \text{ise} \\ 1+|F_i| & F_i > 0 \ \text{ise} \end{cases} \qquad (8)$$

Burada *F* amaç fonksiyonunun değeridir. 5. adımda terk edilen çözümlerin olup olmadığını belirlemek üzere bir "limit" sayısı geliştirilmiştir. Bu sayı algoritmadaki tek değişkendir. Bunun işçi arıların sayısı ile tasarım uzayının boyutunun çarpımına eşit alınması önerilmiştir; limit = (N/2) x m. Buna göre belli bir noktada limit sayısı kadar denemeye rağmen bir iyileşme olmuyorsa bu nokta atılıp yerine yeni bir aday nokta üretilerek işlemlere devam edilir.

## 3. UYGULAMA

### 3.1 Kodların Test Edilmesi

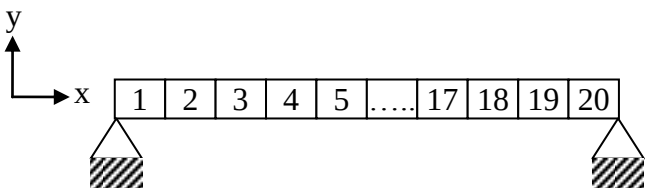
Bu amaçla iki farklı benchmark fonksiyonu dikkate alıyoruz. Bunlar Rosenbrock ve Sphere olarak bilinen tek optimum noktaya sahip (*unimodal*) fonksiyonlardır (Tablo 1). Bu fonksiyonlardaki değişken sayısı *m*=30, iterasyon sayısı 1000, değişkenlerin alt ve üst sınırları [+20,-20] olarak belirlenmiştir. İstatistiksel değerlendirme yapmak için algoritmalar 20 şer defa çalıştırılmış, en iyi, ortalama, standart sapma ve en kötü sonuçlar tabloda gösterilmiştir. Elde edilen sonuçlar ABC yönteminin mevcut problemler için daha iyi olduğunu göstermektedir. İlgili kaynaklarda (mesela [20]) da bu doğrultuda sonuçlar yer almaktadır.

Tablo 1. Kodların test edilmesi

| Fonksiyon | En Kötü | Ortalama | En iyi | Standart Sapma |
|---|---|---|---|---|
| $f_1$ | 240.48 | 74.30 | 16.09 | 60.54 |
| $f_2$ | 5.99 | 2.29 | 0.13 | 2.02 |
| $f_1 = \sum_{i=1}^{m-1}\left(100(x_i^2 - x_{i+1})^2 + (1-x_i)^2\right)$ (Rosenbrock) | | | | |
| $f_2 = \sum_{i=1}^{m} x_i^2$ (Sphere) | | | | |

## 3.2 Hasar Tespiti Uygulaması

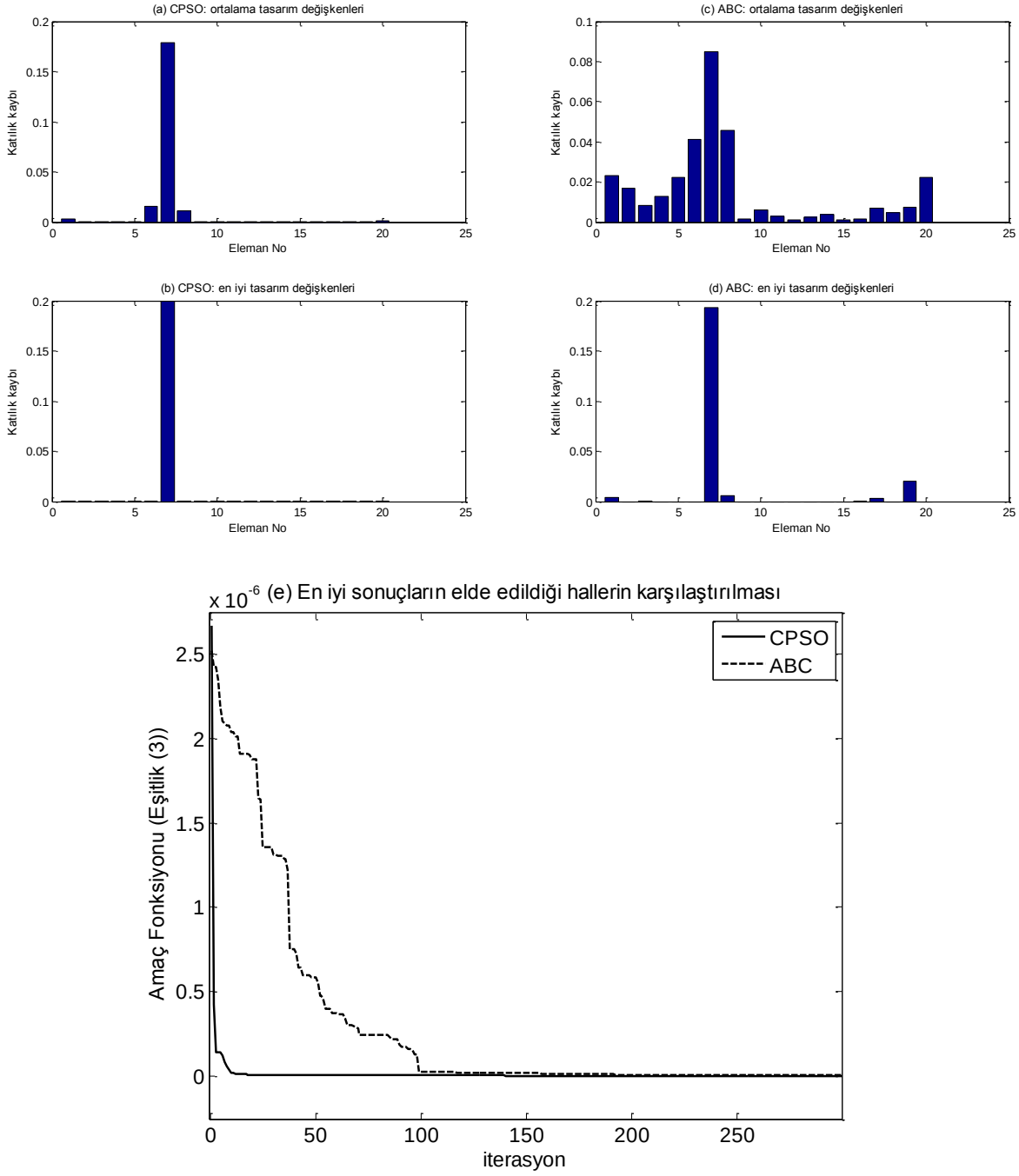
Dikkate aldığımız iki ucu basit mesnetli kirişin sonlu eleman modeli ve özellikleri Şekil 2'de gösterilmiştir. İki farklı hasar durumu dikkate alınmıştır. Hasar1 halinde 7 nolu elemanda %20 katılık kaybı, Hasar2 halinde 10 ve 15 nolu elemanlarda sırasıyla %20 ve %30 katılık kaybı vardır. İlk beş doğal frekans ve titreşim modu kullanılarak amaç fonksiyonu (Eşitlik (3)) hesaplanmıştır. Önceki tecrübelere dayanarak [5] popülasyon sayısı 50 ve iterasyon sayısı 300 alınmıştır. İstatistikî değerlendirme yapmak için algoritmalar 30'ar defa çalıştırılmıştır.




Boy: 50m
Kesit: 0.5 x 1 $m^2$
Yoğunluk: 7860 $kg/m^3$
Elastisite Modülü: 210 GPa

Şekil 2. Kirişin sonlu eleman modeli ve özellikleri

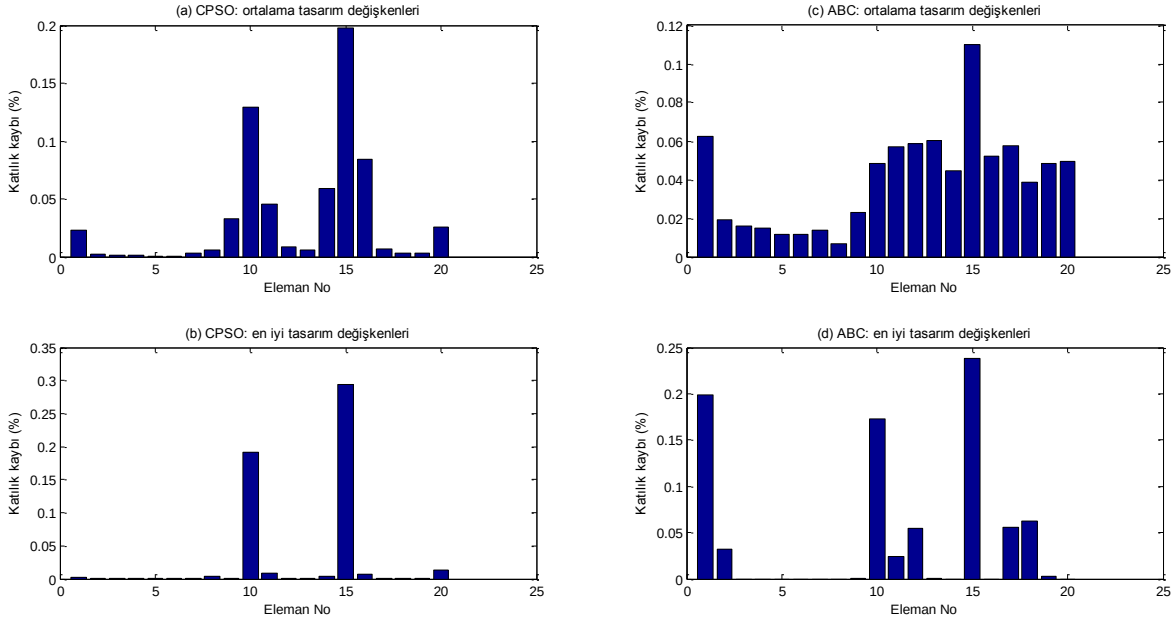
Optimum noktaya karşılık gelen tasarım değişkenlerinin ortalama değerleri ve en iyi durumdaki tasarım değişkenlerinin değerleri Şekil 3'de gösterilmiştir. Hasar1 halinde sadece 7 nolu elemanda %20 oranında katılık kaybı vardır. Şu halde şekle baktığımızda sadece bu eleman için çubuk değerinin 0.2, diğerlerinde sıfır olması gerekir. Bu duruma en yakın sonuç Şekil 3-b'de görülmektedir. Buna göre CPSO ile en iyi çözümün elde edildiği grafik gerçeğe en yakın sonucu vermiştir. ABC ile de buna yakın sonuç elde edilmiştir (Şekil 3-d). Fakat bazı elemanlarda sıfırdan farklı değerler mevcuttur. Ortalama değerler anlamında da CPSO'nun daha iyi olduğu görülmektedir (Şekil 3-a ve 3-c).




Şekil 3. Hasar1 için sonuçlar

Hasar2 durumunda problemin biraz daha zor olması dolayısıyla iterasyon sayısı 400'e çıkarılmıştır. Bu halde sonuçlar Şekil 4'de gösterilmiştir. Hasar2 durumunda 10 nolu elemandaki katılık azalması % 20, 15 nolu elemandaki de %30'dur. Bu durumda çubuk grafiklerine bakıldığında sadece bu elemanlara ait değerler sırasıyla 0.2 ve 0.3 olmalı, diğerlerininki sıfır olmalıdır. Buna en yakın sonuç Şekil 4'de (b) grafiğinde görülmektedir. (d) grafiğine bakıldığında 1 nolu elemanda %20 katılık kaybı var gibi yanıltıcı bir sonuç görülmektedir. Dolayısıyla CPSO'nun performans bakımından ABC'den iyi olduğu anlaşılmaktadır. Ortalama değerler anlamında kıyaslama yaptığımızda da CPSO'nun daha iyi olduğunu anlıyoruz. (a) grafiğinde en yüksek genlikler 10 ve 15 nolu elemanlardadır. Bunların değerlerinin 0.2 ve 0.3'den farklı olması sonuçların hasar yerini belirleme

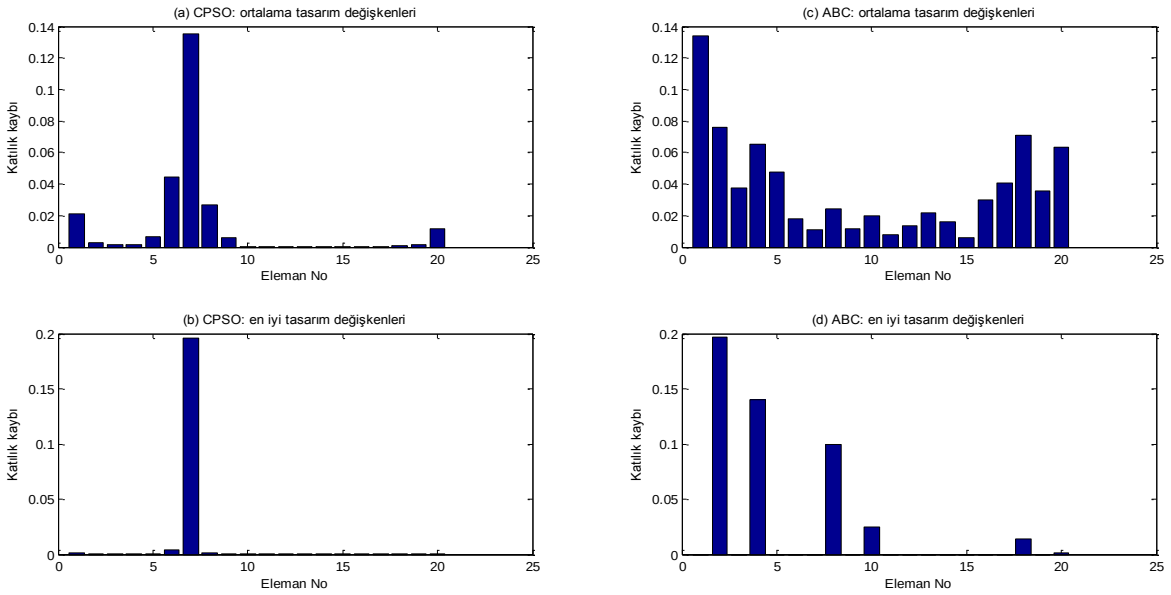
anlamında yeterli, hasar derecesini tespitte önceki duruma göre daha kötü olduğunu göstermektedir. Fakat bu durumda da CPSO ABC'dan iyidir. Ortalama değerlere göre CPSO'nun daha iyi olması bunun daha "robust" olduğunu ima etmektedir.



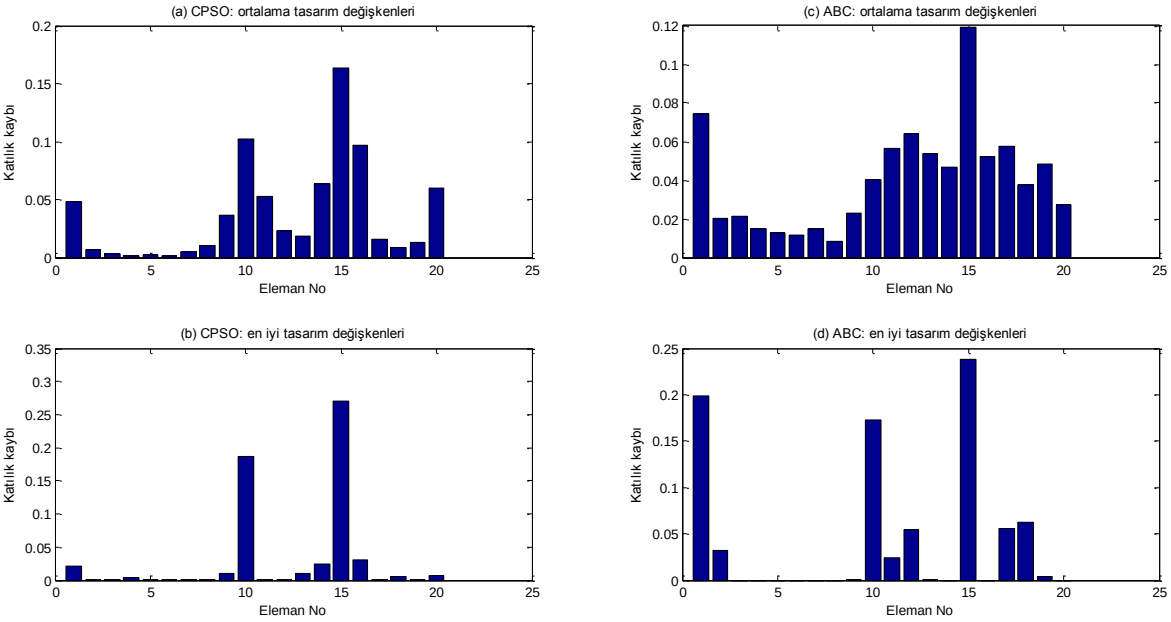

Şekil 4. Hasar2 için sonuçlar

## 3.2.1 İterasyon Sayısına Göre Karşılaştırma

Mevcut problem için programların yakınsama hızlarını karşılaştırmak üzere farklı iterasyon sayılarındaki sonuçlar incelenmiştir. Örneğin Hasar1 için iterasyon sayısı 100 alındığında Şekil 5'de görüldüğü üzere CPSO'nun daha iyi sonuç verdiği görülmektedir. ABC ise yukarıdakilerden daha kötü olup bu halde hasar bilgisini açığa çıkarma anlamında yetersiz kalmıştır. Benzer bir sonuç Hasar2 için elde edilmiştir (bkz. Şekil 6). Bu hasar durumu için iterasyon sayısı 200'e düşürüldüğünde ABC'nun yine yetersiz olduğu, CPSO'nun hasar bilgisini yeterince açığa çıkardığı görülmektedir.




Şekil 5. Hasar1 için sonuçlar (iterasyon sayısı: 100)

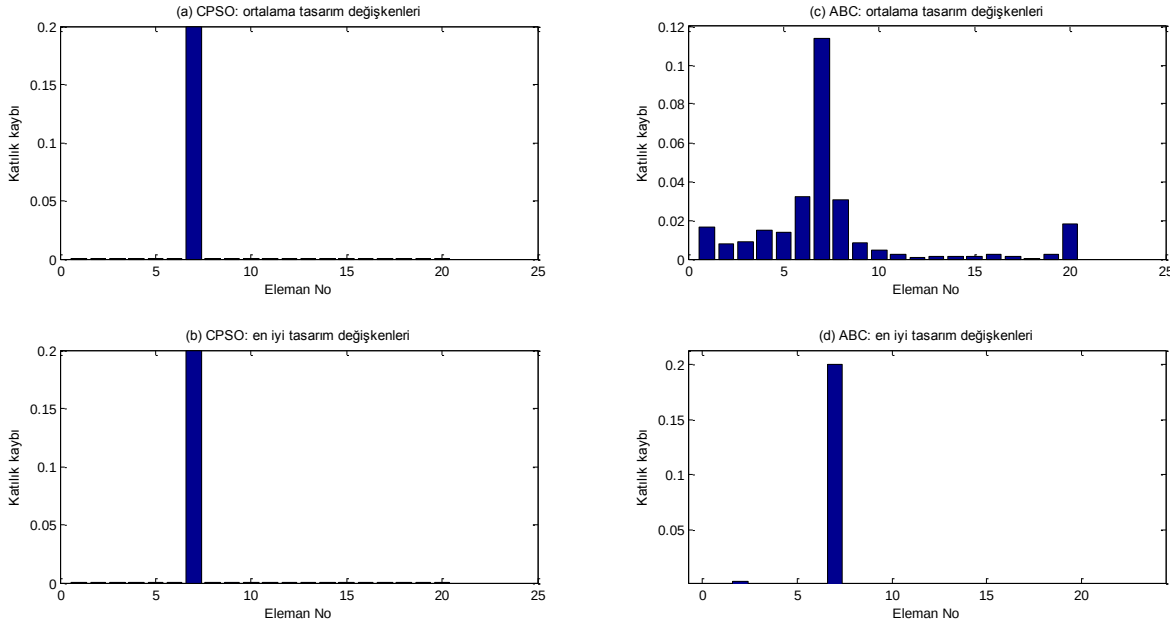



Şekil 6. Hasar2 için sonuçlar (iterasyon sayısı: 200)

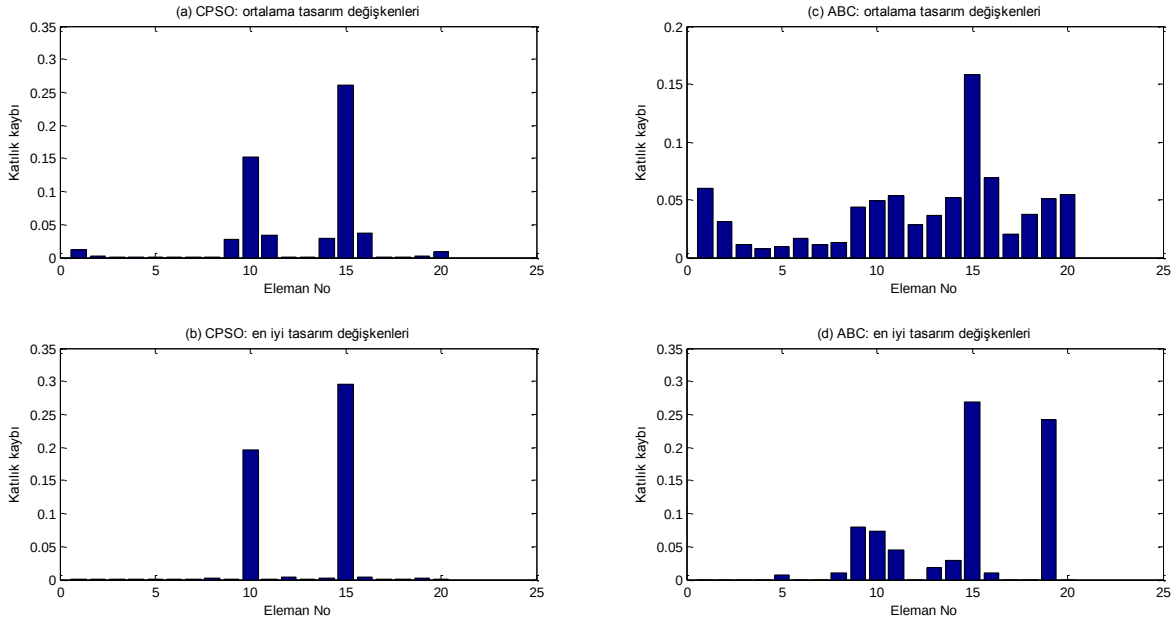
## 3.2.2 Popülasyon Büyüklüğüne Göre Karşılaştırma

Popülasyon büyüklüğünün sonuçların iyileşmesine katkısını anlamak üzere popülasyon büyüklüğü 50'den 100'e çıkarılmıştır. İterasyon sayısı yine 300 olmak üzere Hasar1 için Şekil 7'deki sonuçlar elde edilmiştir. Şekil 3 ile karşılaştırıldığında sonuçlarda nispeten iyileşme görülüyor. Fakat bu halde de CPSO'nun daha iyi olduğu anlaşılmaktadır. Hasar2 için iterasyon sayısı yine 400 alınarak popülasyon sayısı 100 olduğunda Şekil 8'deki sonuçlar elde edilmiştir. CPSO ile ilgili sonuçlarda

(Şekil 4 ile kıyaslandığında) nispi bir iyileşme görülmekte, fakat ABC algoritması için aynı şey söylenememektedir.




Şekil 7. Hasar1 için sonuçlar (Popülasyon büyüklüğü 100)




Şekil 8. Hasar2 için sonuçlar (Popülasyon büyüklüğü 100)

## 4. SONUÇ

Bu çalışmada son yıllarda yaygın kullanılan iki optimizasyon algoritmasının etkinliği hasar tespiti problemi için karşılaştırılmıştır. Bunlardan biri daraltma çarpanı kullanan PSO, yani CPSO, diğeri ABC yöntemidir. Benchmark problemlere tatbik edildiklerinde ABC yönteminin daha iyi sonuç verdiği

görülmüştür. Fakat hasar tespiti problemi için yapılan karşılaştırmada CPSO belirgin bir şekilde üstün gelmiştir. [12] nolu çalışmada tam olarak aynı olmamakla birlikte buradakine benzer bir sonuç görülmektedir. Aynı problem için simülasyonla elde edilmiş veriler kullanıldığında PSO, diğer durumda arı kolonisi yaklaşımının daha iyi olduğu gösterilmiştir. Üstelik ilgili çalışmada PSO'nun klasik versiyonu kullanılmıştır. Öte yandan, ABC'de tek değişken limit sayısı olup bunun farklı kombinasyonlarının sonucu etkilediği bilinmektedir (örneğin bkz. [19]). Dolayısıyla, farklı limit değerleri ile problemin çözümü üzerinde durulabilir. Ayrıca, yapılan çalışmada ölçüm verilerine karışan parazit etkisi ihmal edilmiştir. Hâlbuki ölçülen doğal frekans ve mod şekillerine parazit karışması kaçınılmazdır. İleride bu gibi etkilerin dikkate alındığı başka bir çalışma yapmak planlanmaktadır.

## 5. KAYNAKÇA

[1] DOEBLING, S.W., FARRAR, C.R., PRIME, M.B., VE SHEVITZ, D.W. Damage identification and health monitoring of structural and mechanical systems from changes in their vibration characteristics: A Literature Review. Rep. LA-13070-MS, UC-900, Los Alamos National Laboratory, USA, (1996).

[2] ALVANDI, A. VE CREMONA, C., Assessment of vibration-based damage identification techniques, Journal of Sound and Vibration, 292, 179-202, (2006).

[3] YAN, Y. J., CHENG, L., WU, Z. Y. VE YAM, L. H., Development in vibration-based structural damage detection, Mechanical Systems and Signal Processing 21, 2198-2211, (2007).

[4] NOBAHARI, M. VE SEYEDPOOR, S. M., Structural damage detection using an efficient correlation based index and a modified genetic algorithm, Mathematical and Computer Modelling, 53, 1798-1809, (2011).

[5] GÖKDAĞ, H. VE YILDIZ, A.R. Structural damage detection using modal parameters and particle swarm optimization, Materials Testing, 54, 1-6, (2012).

[6] GOMES, H. VE SILVA, N., Some comparisons for damage detection on structures using genetic algorithms and modal sensitivity method, Applied Mathematical Modelling, 32, 2216-2232, (2008).

[7] PERERA, R., FANG, S. E. VE HUERTA, C., Structural crack detection without updated baseline model by single and multiobjective optimization, Mechanical Systems and Signal Processing, 23, 752-768, (2009).

[8] FRISWELL, M. I., PENNY, J. E. T. VE GARVEY, S. D., A combined genetic algorithm and eigensensitivity algorithm for the location of damage in structures, Computers and Structures, 69, 547-556, (1998).

[9] VAKIL-BAGHMISHEH, M. , PEIMANI, M., SADEGHI, M. VE ETTEFAGH M., Crack detection in beam-like structures using genetic algorithms, Applied Soft Computing, 8, 1150-1160, (2008).

[10] BEGAMBRE, O. VE LAIER, J. E., A hybrid particle swarm optimization – simplex algorithm (PSOS) for structural damage identification, Advances in Engineering Software, 40, 883-891, (2009).

[11] SEYEDPOOR, S. M., A two stage method for structural damage detection using a modal strain energy based index and particle swarm optimization, International Journal of Nonlinear Mechanics, 47, 1- 8, (2012).

[12] MORADI, S., RAZI, P. VE FATAHI, L., On the application of the bees algorithm to the problem of crack detection of beam-type structures, Computers and Structures, 89, 2169-2175, (2012).

[13] CLERC, M. VE KENNEDY, J., The particle swarm-explosion, stability, and convergence in a multidimensional complex space, IEEE Trans. on Evol. Comput, 6(1), 58-73, (2002).

[14] http://mf.erciyes.edu.tr/abc/software.htm (Erişim tarihi: 01.07.2012)

[15] MERUANE, V. VE HEYLEN, W., An hybrid real genetic algorithm to detect structural damage using modal properties, Mechanical Systems and Signal Processing, 25, 1559-1573, (2011).

[16] KENNEDY, J. VE EBERHART, R., Particle swarm optimization, Proc. of the 4th IEEE Int. Conf. on Neural Netw., 4, 1942-1948, (1995).

[17] KARABOĞA, D., An idea based on honey bee swarm for numerical optimization, Technical Report TR06, Computer Engineering Department, Erciyes University, Turkey, 2005.

[18] ÖZTÜRK, A., ÇOBANLI, S., DUMAN, S., TOSUN, S. VE DÖŞOĞLU, K., Yapay arı kolonisi algoritması ile elektrik güç sistemi optimal yakıt maliyetinin belirlenmesi, 6th International Advanced Technologies Symposium (IATS'11), 16-18 May 2011, Elazığ, Turkey.

[19] NARASIMHAN, H., Parallel artificial bee colony (PABC) algorithm, IEEE World Congress on Nature & Biologically Inspired Computing, pp 301-306, Coimbatore, (2009).

[20] AKAY, B. VE KARABOĞA, D., A modified artificial bee colony algorithm for real-parameter optimization, Information Sciences, 192, 120-142, (2012).