

## GELİŞTİRİLMİŞ YAPAY ARI KOLONİ ALGORİTMASI İLE KAFES VE DÜZLEMSEL ÇELİK YAPILARIN OPTİMUM TASARIMI

Özer SEVİM<sup>1\*</sup>, Mustafa SÖNMEZ<sup>2</sup>

<sup>1</sup> Kırıkkale Üniversitesi, Mühendislik Fakültesi, İnşaat Mühendisliği Bölümü, 71450 Kırıkkale

<sup>2</sup> Aksaray Üniversitesi, Mühendislik Fakültesi, İnşaat Mühendisliği Bölümü, 68100 Aksaray

### ÖZET

Son yıllarda sürü zekâsına dayalı optimizasyon teknikleri (karınca kolonisi optimizasyonu ve parçacık sürü optimizasyonu) araştırmacılar tarafından büyük ilgi görmüştür. Bu algoritmalar değişik mühendislik problemlerinin çözümü için kullanılmıştır.

Bu çalışmada Yapay Arı Koloni Algoritması (ABC) kullanılarak düzlem ve kafes çelik çerçevelerin ayrık optimizasyonu sunulmuştur. ABC algoritmasının kullanma bilirliğini gösterilmesi için 168 elemana ve 20 tasarım değişkenine çıkabilen iki farklı yapısal örnek çözülmüştür ve sonuçlar literatürdeki sezgisel arama teknikleri ile elde edilen sonuçları ile kıyaslanmıştır. Sonuçların düzlemsel ve kafes çelik çerçeve yapılarının tasarımı için ABC algoritmasının çok etkili ve güçlü bir teknik olduğu gösterilmiştir.

**Anahtar kelimeler:** Yapay Arı Koloni Algoritması, Ayrık Optimizasyon, Çelik Çerçeve Tasarımı

## OPTIMUM DESIGN OF TRUSS AND PLANAR STEEL STRUCTURES WITH IMPROVED ARTIFICIAL BEE COLONY ALGORITHM

### ABSTRACT

Over the past few years, swarm intelligence based optimization techniques such as ant colony optimization and particle swarm optimization have received considerable attention from researchers. These algorithms have been used in the solution of various engineering problems.

This study presents an Artificial Bee Colony Algorithm (ABC) technique for discrete optimization of planar steel frames and truss steel frames. In order to demonstrate the effectiveness of algorithm, two problems with up to 168 members and 20 design variables were solved and results were compared with those obtained using other well-known meta-heuristic search techniques. The results demonstrate that the artificial bee colony algorithm (ABC) is very effective and robust for planar steel frames and truss steel frame design.

**Keywords:** Artificial Bee Colony Algorithm, Discrete Optimization, Steel Frame Design

## 1. GİRİŞ

Dünya nüfusunun hızla artması, kaynakların sınırlı olması insanları farklı arayışlara yönlendirmiştir. Bu arayışlardan biriside istenilen şartları sağlayan en ekonomik yapıların tasarlanması olmuştur. En ekonomik yapıların tasarımında sadece mühendislik sezgisi yeterli olmadığından, belli bir yöntem (metodoloji) kullanılarak tasarım yapılması gerekliliği ortaya çıkmıştır. Örnek olarak, inşaat mühendisliğinde bir yapıyı taşıyıcı sistemi tasarlanırken ilk önce öntasarımın yapılır. Öntasarımın amacı ilk gerilmeleri bularak ön tasarımdaki kabullerin doğruluğunu kontrol etmektir. Ön tasarımdaki ilk kabullerin tam doğru ve en ekonomik olma olasılığı yok denecek kadar azdır. Ön tasarımda seçilen kesitler küçük ise gerilme ve yer değiştirme şartları sağlanmadığı için kesitler artırılır, eğer gerilme ve yer değiştirme değerleri kesitin taşıyabileceğinden çok daha küçük ise sistemin daha ekonomik olması için kesitlerin azaltılması gereklidir. Anlaşılacağı gibi bu işlemler yinelemeli (iteratif) işlemlerdir. Genel olarak bu yinelemeli işlemler eğer bir yöntem (metodoloji) kullanılarak yapılır ise buna optimizasyon denir.

Mühendislik tasarımda kullanılan optimizasyon teknikleri klasik (gradyan tabanlı) metotlar ve modern sezgisel (gradyan tabanlı olmayan) metotlar olarak iki ana başlıkta incelenebilir. Araştırmacılar çeşitli mühendislik problemlerini gradyan tabanlı optimizasyon yöntemleri kullanarak çözmek için çaba göstermişler, ancak gerçekte hayattaki mühendislik problemleri belli bir çözüm yöntemine uyacak şekilde modellemek kolay olmadığından dolayı pratik uygulama alanı bulamamıştır. Bu nedenle klasik optimizasyon tekniklerinin bu yetersizliklerini aşabilmek için problemlerden ve modelden bağımsız olan, doğadan esinlenen sezgisel optimizasyon algoritmaları geliştirilmiştir. Bu teknikler hem etkin hem de daha esnek olduğundan her tür problem gereksinimlerine göre uyarlanabilmektedir.

Son yıllardaki bilgisayar teknolojisindeki gelişmeler; çok daha hızlı işlemciler ve çok daha yüksek kapasiteli bilgisayarlar ve bu gelişmelere paralel olarak programlama teknolojisindeki gelişmeler mühendislik tasarımında bilgisayar kullanımını vazgeçilmez hale getirmiştir. Eskiden sadece analiz için programlar geliştirilirken artık tasarım yapan programlarda geliştirilmeye başlanmıştır. Hazır paket programların birçoğunda ismi artık CAD (Computer-Aided Design-Bilgisayar destekli tasarım) şeklinde verilmeye başlanmıştır.

### 1.1. Bu Çalışmanın Amacı

Artan hammadde ve işçilik maliyetleri, hammadde bulunmasında yaşanabilen problemler ve çelik fiyatlarındaki ani yükselişler çelik taşıyıcı sistemlerin ekonomik olarak boyutlandırılması zorunluluğunu ortaya çıkarmıştır. Bu amaçla çelik yapıların optimum tasarımı konusunda farklı optimizasyon yöntemlerini kullanan önemli çalışmalar yapılmıştır.

Bu çalışmanın amacı Yapay Arı Kolonisi (Artificial Bee Colony) optimizasyon algoritması kullanılarak kafes ve çerçeve sistemlerin optimum tasarımlarını yapacak bilgisayar programı geliştirmek ve ABC'nin çelik çerçevelerin tasarımı için etkili bir yöntem olduğunu ortaya kaymaktır. Yapay arı koloni algoritması (ABC-Artificial Bee Colony) Erciyes Üniversitesinden Derviş Karaboğa [1][2] tarafından geliştirilmiştir. Sonmez [3][4] ABC algoritmasını uzay kafelerin optimum tasarımı için kullanmıştır. Sonra Değertekin [5] ve Aydogdu vd., [6] ABC algoritmasını düzlem çerçevelerin optimizasyonunda tarafından kullanılmıştır Düzlem ve uzay çelik çerçeve sistemlerin tasarımında tasarım değişkenleri olarak sadece eleman kesitleri ele alınmıştır. Profiller hazır profil tablolarından seçilecektir. Yani bu çalışmada ayrık (discrete) optimizasyon tekniği kullanılacaktır. Bu çalışmada ticari hazır paket programlarından Sap2000 yapı analiz programı hazırlanan bilgisayar programına entegre edilmiştir. Programlama dili olarak nesne tabanlı C# programı seçilmiştir. C# programlama dilinin seçilmesindeki en önemli etken, .Netframework çevresinde hazır paket programlar ile uyumlu çalışmasıdır. Sonuç olarak, çubuk kuvvetleri ve deplasmanlar Sap2000 programı yardımı ile hesaplanmış olup optimizasyon işlemi ise C# program dili kullanılarak geliştirilmiştir [7].

## 2. Yapay Arı Koloni (ABC) Algoritması

### 2.1. Sürü Zekası

Sürü terimi birlikte davranış sergileyen karınca, arı kolonileri, balık, kuş ve böcek toplulukları için kullanılmaktadır. Bir sürüdeki bireyler herhangi bir danışma işlemi olmadan hareket ederler. Herhangi bir ilişkileri olmadan kendi alanlarındaki kurallar vasıtasıyla kendi kendine organize olabilen bireyler arasındaki etkileşim sürü zekası adı verilen birlikte davranış zekasının ortaya çıkmasına sebep olmaktadır. Kendi kendine

organize olabilme sürü sisteminin temel özelliğidir[8]. Bonabeau [9] kendi kendine organize olabilmeyi dört özellik ile karakterize etmiştir.

1. Pozitif geri besleme: Elverişli durumların oluşmasını destekleyen davranışsal kurallar sayesinde oluşur. Pozitif geri beslenmeye örnek olarak arıların dans etmeleri ile birlikte diğer arıların bir işe yönlendirilmeleri verilebilir.
2. Negatif geri besleme: Sürünün birlikte uyumlu şekilde hareketlerini sağlamak için pozitif geri beslemeyi dengeler nitelikte çalışan kurallar ile ortaya çıkmaktadır. Herhangi bir yiyecek tüketiminde, iş konusunda ve benzeri durumlarda doyuma gitmeyi önlemek için negatif geri besleme mekanizması gerekmektedir.
3. Dalgalanmalar: Bireylerin yanlışları, yapılan işi değiştirmeleri yeni çözümlerin bulunmasını sağladığından yaratıcılık ve yenilik açısından önem taşımaktadır.
4. Çoklu etkileşimler: Herhangi bir sürüdeki bazı bireyler arasında bilgi alışverişi sürü içerisinde bilgi alışverişi de sağlar.

Bal arıları kovanlarda sürüler halinde yaşar ve her türlü işbirliğinde topluluk zekâlarıyla cevap veren doğada buluna sürülerden birisidir. Bal arıları fotoğrafik hafızaya, uzay çağı sensörlerine, yön bulma sistemine, sezgisel kavrama yeteneğine, herhangi bir iş birliğinde grup olarak karar verme özelliğine sahiptirler. Bu özellikler araştırmacıları arıların davranışlarını modellemeleri yönünde teşvik edici olmuştur.

## 2.2. Doğadaki Arıların Davranışı

Bir bal arısı kolonisi tek bir kraliçe arı, binlerce dişi işçi arı ve yüzlerce erkek arıdan oluşur. Bal arıları koloni halinde yaşayan sosyal böceklerdir. Kraliçenin görevi erkek arılar ile çiftleşerek yeni koloniler oluşturmaktır. Aynı zamanda kraliçe arı yumurtlama özelliğine sahip tek arıdır. Erkek arılar yumurtaları döleyerek dişi işçi arıların oluşmasına yardım eder. Bir kolonideki tüm görevler; gıda işleme, bu gıdaların depolanması, hücrelerin temizliği, petek inşa etmek ve kovana koruma gibi görevler henüz 3 haftalık olmamış dişi arılar tarafından yapılır. Kovan içerisine çalışan bu işçi arılar yaklaşık üç haftalık olduğunda, kovan içerisindeki tüm işlerini bırakırlar ve çiçeklerden nektar toplayan birer izci arı olurlar.

Bir bal arısı yaklaşık olarak kovandan 12 kilometre uzağa nektar bulmak için gidebilir ancak genellikle bu mesafe 3 km'yi geçmez. Eğer bir arı çiçeklerin birinde nektar keşfederse, bu nektarı alır, kovana götürür ve gerekli yere depo ettikten sonra "sallanma dansı" (Waggle Dance) ile nektarın bulunduğu yeri diğer izci arılara iletir[1][3]. Şekil 1.'de bir kovan, üç gıda kaynağı ve bu gıda kaynağını keşfeden arılar tarafından gerçekleştirilen sallanma dansı şematik olarak gösterilmiştir.



Şekil 1. Bal arılarının sallanma dansı (Waggle Dance)

Sekiz rakamına benzeyen sallanma dansı, kovan içerisindeki diğer arılara üç önemli bilgi verir. Bunlar;

1. Sekize benzeyen şeklin ortasındaki düz çizgi besin kaynağının yönünü,
2. Sekize benzeyen şeklin ortasındaki düz çizginin uzunluğu besin kaynağının kovandan ne kadar uzakta olduğunu,
3. Sekize benzeyen şeklin ortasındaki düz çizginin alındığı süre ise nektar kalitesi hakkında bilgi verir.

Dans eden arıları seyreden kovandaki diğer arılara izleyiciler (onlookers) denir. Bu izleyici arıların bazıları dans ederek bilgi veren arıları takip ederken, bazıları da bunları izlemeden kendi bildikleri yöne doğru giderler. Kimseyi takip etmeden istedikleri yönde giden arılara izci arı (scout bee) denir. Kovanda nektar arayan bir arı aşağıdaki yolların birini izlemektedir.

1. En son keşfettiği besin kaynağını terk ederek yeni bir besin kaynağı arar,
2. En son keşfettiği bir besin kaynağına tekrar gider,
3. En son keşfettiği besin kaynağına dönmeden önce kovandaki diğer arıları bilgilendirmek için sallanma dansı yapar [3][4][7]

### 2.3. Arı Koloni Algoritması

ABC algoritmasında, arılar tarafından bulunan her bir besin kaynağının yeri, tasarım değişkenlerinin değerini ve bu besin kaynağındaki nektarın miktarı ise amaç fonksiyonunun değerini temsil etmektedir. Her bir arı en çok nektarı olan çiçeği bulmak için kovandan ayrıldığında rastgele bir yöne ve uzaklığa gider. Bu ilk noktaların koordinatları aşağıdaki gibi tüm arama alanı üzerinde rastgele başlatılır:

$${}^0x_j = \min x_j + \lambda(\max x_j - \min x_j), \quad \min x_j \leq x_j \leq \max x_j \quad (1)$$

Yukarıdaki eşitlikte,  $\lambda$  değeri 0 ile 1 arasında rastgele bir sayıdır. Sağ alt indis her tasarım değişkeninin değişken numarasını, sol üst indis değişkenin alabileceği maksimum, minimum değeri ve sol alt indis ise çözümün hangi besin kaynağında gerçekleştiğini göstermektedir. Denklem 1’de verilen  $\max x_j$  ve  $\min x_j$  değerleri tasarım değişkeninin önceden bilinen tasarım sınırlarıdır. Eğer tasarım değişkenleri sürekli ise Denklem 1’ de verildiği gibi gösterilir. Eğer tasarım değişkenleri sürekli değil ise ver bir listeden seçilecek ise bu durumda Denklem 1 aşağıdaki şekilde yazılabilir.

$${}^0I_j = \min I_j + \lambda(\max I_j - \min I_j), \quad \min I_j \leq I_j \leq \max I_j \quad (2)$$

Denklem (2)’deki  $I$  tasarım değişkenlerinin listedeki yerini (indisini) göstermektedir. Her bir arı için ilk koordinatlar seçildikten sonra, bu arıların bu koordinatlardaki besin kaynaklarından buldukları nektarın miktarı tespit edilecektir. Bu hesap işlemi optimizasyonda amaç fonksiyonunun değerinin hesaplanması anlamına gelir. Amaç fonksiyonu hesaplandıktan sonra en iyi nektarı bulan arılar aynı bölgeye giderken, bu arılara istihdam edilen (employed) arılar denir. İyi nektar bulamayan arılar ise iyi nektar bulanları takip edeceklerdir ve bu arılara da izleyici (onlooker) arılar denilecektir. Genellikle istihdam edilen arı sayısı ve izleyici arıların sayısı eşit olarak alınır ama bu zorunlu değildir. Burada dikkat edilmesi gereken nokta ise eğer ağırlık optimizasyonu yapılıyor ise en hafif sistem en iyi nektarı, eğer kar optimizasyonu yapılıyor ise en büyük değer en iyi nektarı gösterecektir.

İstihdam edilen arılar dans ederek işsiz arıları (seyirci arılar) kendi buldukları besin kaynağına gelmeleri için ikna etmeye çalışırlar. Dans eden arıları, nektarların kalitesine göre izleyici arıların takip edeceği düşünülür ise az veya hiç nektar olmayan besin kaynağını daha az işsiz arı seçerken, bol nektar kaynağını ziyaret edecek işsiz arıların sayısı da buna bağlı olarak fazla olacağı kabul edilir. Bu işsiz arıların nektar miktarı ile orantılı bir olasılığa göre besin kaynağını seçmesi anlamına gelir.  $i$ ’inci besin kaynağını arıların seçme olasılığı aşağıdaki şekilde hesaplanır:

$$p_i = \frac{(f_i(x))^{-1}}{\sum_{m=1}^n (f_m(x))^{-1}} \quad (3)$$

Bu formül ile işsiz arıların kaç tanesinin  $i$ ’inci besin kaynağına gideceği bulunur. Bu adımdan sonra  $i$ ’inci besin kaynağına bir adet bu kaynaktan istihdam edilen arı ve olasılık ile hesaplanan sayıda arı bu bölgeye gönderilir. Bu bölgedeki her bir arının yeni yeri aşağıdaki formül ile hesaplanır:

$${}^{yeni}I_j = {}^{isth}I_j + \phi({}^{isth}I_j - {}^{isth}I_j) \quad (4)$$

Yukarıdaki eşitlikte,  $\phi$  değeri -1 ile 1 arasında rastgele bir reel sayıdır. Sağ alt indis ( $j$ ) tasarım değişkeninin adını gösterirken sol alt indisler ( $i$ ) besin kaynağının numarasını temsil eder, ( $k$ ) ise 1 ile  $SN$  arasında  $i$  ye eşit olmayan bir tam sayıyı temsil eder. Bu formül diğer gıda kaynağı arıların arama faaliyetlerini kontrol etmek için kullanılır.  ${}^{isth}I_j$  ABC yakınsama davranışlarında önemli bir rol oynar. Bulunduğu yerde besin seviyesi

önceki en iyi besin kaynağına göre daha iyi ise, bulunduğu konum iyi besin kaynağı özelliği kazanır yani  $I_j = \frac{yeni I_j}{k I_j}$  olur. Aksi takdirde, eski konumu en iyi besin kaynağı olarak korunur.

Fonksiyon optimizasyonunda ABC algoritmasının nasıl kullanılacağı Kaynak [7]'de ayrıntılı olarak ele alınmıştır.

### 3.YAPAY ARI KOLONİ OPTİMİZASYONUNUN ÇELİK YAPILARA UYGULANMASI

En genel hali ile optimizasyon problemi aşağıdaki gibi tanımlanabilir;

$f(\vec{x})$  fonksiyonunu minimum yapan ve

$$g_j(\vec{x}) \leq 0 \quad j = 1, 2, 3, \dots, m$$

$$l_k(\vec{x}) = 0 \quad k = 1, 2, 3, \dots, p$$

Şartlarını sağlayan  $\vec{x} = \{x_1, \dots, x_i, \dots, x_n\}$ 'lerin bulunmasıdır.

Buradaki  $\vec{x}$ ,  $n$  boyutlu bir vektördür ve tasarım değişkenleri vektörü (design variable vector) denir. Tasarım değişkenleri vektöründeki her bir eleman tek bir tasarım değişkenini temsil eder;  $f(\vec{x})$  amaç fonksiyonu (objective function);  $g_j(\vec{x})$  eşitsizlik kısıtları (inequality constraints) ve  $l_k(\vec{x})$  eşitlik kısıtları (equality constraints) olarak tanımlanır.  $n$ ,  $m$  ve  $p$  ise birbiri ile ilişkisi olmayan tamsayılardır; sırası ile tasarım değişkenlerinin, eşitsizlik kısıtlarının ve eşitlik kısıtlarının sayısını göstermek için kullanılmıştır. Yukarıda verilen kısıtlar sınırlı optimizasyon probleminde vardır. Bazı problemlerde sınırlar yoktur ve bu tür problemler de sınırsız optimizasyon denir. Sınırsız optimizasyon aşağıdaki gibi gösterilebilir.

$f(\vec{x})$  fonksiyonunu minimum yapan

$$\vec{x} = \{x_1, \dots, x_i, \dots, x_n\}$$

Yukardaki gibi problemler sınırsız optimizasyon problemi olarak adlandırılır.

$$\beta_{ij}(I) = \frac{\delta_{jl}}{\delta_{ju}} - 1 \leq 0 \quad (5)$$

$$\beta_{ij}(Z) = \frac{\Delta_{jil}}{\Delta_{ju}} - 1 \leq 0 \quad (6)$$

$\delta_{jl}$  : j. kattaki l. yüklemesinden meydana gelen yer değiştirme ve  $\Delta_{jil}$  l yüklemesi için j ile i noktaları arasındaki yer değiştirme farkı olarak hesaplanmıştır.

$\frac{P_u}{\emptyset P_n} \geq 0.2$  durumu için

$$\beta_{il}(I) = \left( \frac{P_u}{\emptyset P_n} \right)_{il} + \frac{8}{9} \left( \frac{M_{ux}}{\emptyset_b M_{nx}} + \frac{M_{uy}}{\emptyset_b M_{ny}} \right)_{il} - 1.0 \quad i = 1, \dots, nm, l = 1, \dots, nl \quad (7)$$

$\frac{P_u}{\emptyset P_n} < 0.2$  durumu için

$$\beta_{il}(I) = \left( \frac{P_u}{2\emptyset P_n} \right)_{il} + \left( \frac{M_{ux}}{\emptyset_b M_{nx}} + \frac{M_{uy}}{\emptyset_b M_{ny}} \right)_{il} - 1.0 \quad i = 1, \dots, nm, l = 1, \dots, nl \quad (8)$$

Burada  $P_u$  çekmede veya basınçta eksenel kuvvet,  $P_n$  çekmede veya basınçta taşınabilecek maksimum kuvvet,  $M_{ux}$  geometrik olarak lineer olmayan yani ikinci dereceden etkileri içeren kuvvetli eksenindeki eğilme momenti,  $M_{nx}$  kuvvetli eksenindeki taşıyabileceği maksimum eğilme momentidir.  $M_{uy}$  geometrik olarak lineer olmayan yani ikinci dereceden etkileri içeren zayıf eksenindeki eğilme momentidir (iki boyutlu çerçevelerde  $M_{uy} = 0$  dır).  $M_{ny}$  zayıf eksenindeki taşıyabileceği maksimum eğilme momenti,  $\emptyset$  ise dayanım azaltma katsayısıdır, basınç için 0.85, çekme için 0.90 ve eğilme için 0.90 olur.

Eğer Denklem 7 ve 8 deki eksenel kuvvet basınç ise,  $P_n$  taşınabilecek maksimum kuvvet,  $\phi = \phi_c$  basınç için dayanım faktörüdür ve 0.85 değerindedir. Elemanların taşıyacağı maksimum kuvvet aşağıda Denklem 9' da verilmiştir.

$$P_n = A_g \times F_{cr} \quad (9)$$

$$F_{cr} = (0.658^{\lambda_c^2}) F_y \quad (10)$$

$$\lambda_c \leq 1.5 \text{ ise} \quad (11)$$

$$F_{cr} = \left( \frac{0.877}{\lambda_c^2} \right) F_y \quad \lambda_c > 1.5 \text{ ise}$$

$$\lambda_c = \frac{KL}{r\pi} \sqrt{\frac{F_y}{E}} \quad (12)$$

Burada  $A_g$  elemanın en kesit alanı,  $K$  etkili uzunluk faktörü (narinliğe bağlı katsayı),  $E$  eleman malzemesinin Elastisite Modülü;  $r$  atalet yarıçapı,  $L$  eleman uzunluğu,  $F_y$  çeliğin akma gerilmesidir. Etkili uzunluk faktörü ( $K$ ) yanal ötelenmesi önlenmemiş sistemler için Denklem 13 ile hesaplanabilir.

$$K = \sqrt{\frac{1.6G_A G_B + 4.0(G_A + G_B) + 7.50}{G_A + G_B + 7.50}} \quad (13)$$

$G_A$  ve  $G_B$  sınırlama faktörü Denklem 14 ile hesaplanabilir.

$$G = \frac{\sum(I_c/L_c)}{\sum I_g/L_g} \quad (14)$$

Burada  $I_c$  kolon atalet momenti,  $L_c$  kolon uzunluğu,  $I_g$  kiriş atalet momenti,  $L_g$  ise kiriş uzunluğudur. Burada toplama işlemi ise sadece eğilme düzleminde bulunan ve düğüme bağlanan tüm kiriş ve kolonun toplamını ifade etmektedir.

İkinci dereceden etkilerin işleme dahil edilmesi için LRFD-2001 aşağıdaki hesapları göz önüne almaktadır.

$$M_u = \beta_1 M_{nt} + \beta_2 M_{lt} \quad (15)$$

$$\beta_1 = \frac{C_m}{(1 - P_u/P_{e1})} \geq 1.0 \quad (16)$$

$$\beta_2 = \frac{1}{1 - \frac{\sum P_u}{\sum H} \left( \frac{\Delta_{oh}}{L} \right)} \geq 1.0 \quad (17)$$

$M_{nt}$  = yanal deplasmanların önlendiği çerçevelerdeki momentler,

$M_{lt}$  = sadece yanal kuvvetlerden meydana gelen çerçevelerdeki momentler,

$P_u$  = eleman eksenel kuvvetidir,

$C_m$  = yanal ötelenme durumuna göre kullanılan katsayı,

$P_{e1} = \frac{KL}{r}$  katsayısına göre değişen kuvvettir.

$\sum P_u$  = kattaki bütün kolonların aksel kuvvetlerinin toplamıdır,

$\Delta_{oh}$  = belirtilen yükler altında kattaki yer değiştirmedir,

$\sum H$  = kata maruz kalan bütün yatay yüklerin toplamıdır,

$K$  = etkili uzunluk faktörüdür.

Denklem 15'deki  $M_{nt}$  ve  $M_{lt}$ 'yi hesaplamak için ilk önce yanal deplasmanları durdurmak için mesnetler yerleştirilmelidir. Daha sonra bu mesnetler meydana gelen kuvvetler ve yanal kuvvetler birlikte etki ettirilerek  $M_{lt}$ 'ler hesaplanmalıdır. Bu da işlemleri çok artıracığı için TS500'de verilen başka bir alternatif yöntem kullanılmıştır. Bu yöntemde göre  $M_u$  Denklem 18'de ifade edilmiştir.

Lineer olmayan analiz için Denklem 7 ile ikinci dereceden etkiler göz önüne alınmıştır. Kiriş ve kolon elemanları LRFD' de bulunan 267 adet W kesitlerinden seçilmiştir.

$$M_u = \beta_1 \times \beta_2 \times M \quad (18)$$

#### 4. SAYISAL ÖRNEKLER

Yapay arı koloni algoritmasının etkinliğini göstermek için aşağıda belirtilen iki adet tasarım örneği kullanılmıştır. Bu örnekler;

1. Tek yükleme tipine maruz kalan üç açıklıklı, 24 katlı ve 168 elemandan oluşan düzlem çelik çerçeve yapısı,
2. Tek yükleme tipine maruz kalan on elemanlı kafes yapısı,

Tasarım örneklerinin optimizasyonu için kullanılan algoritma Bölüm 2.' de sunulmuştur. Kafes yapının, düzlem çerçevelerin optimum tasarımları için C# programı kullanılarak bir optimizasyon algoritması geliştirilmiştir.

C# programı nesne tabanlı olduğu için geliştirilen program üç ana sınıfa (class) bölünmüştür. Bu sınıflar *Data*, *Bee* ve *Program* dır. *Data* sınıfı tasarım örneği için gerekli sınırları, malzeme özelliklerini ve tasarım değişkenlerinin özelliklerini içermektedir. Bu sınıfta yapı ile ilgili bilgiler mevcuttur. *Bee* sınıfı ise arının davranışını modellemek için geliştirilmiştir. Her bir arının adı, tasarım değişkenlerini ve arılar arasındaki yiyecek arama gibi davranışları modellemek için geliştirilmiştir. Eğer arının besin bulduğu yer uygun bölgede değil ise yine bu sınıf tarafından ceza fonksiyonu ile birlikte amaç fonksiyonları da yine burada hesaplanır. *Program* sınıfı ise Sap2000 programını çağırarak verilen kesit ve geometriye ait yapının göstereceği yer değiştirmeleri ve gerilmeleri bulmak için kullanılmıştır.

Optimizasyon programı çalıştırılmadan önce arı sayısı, Maksimum döngü ve Limit sayıları önceden tespit edilmelidir. Bu çalışmada sunulan bütün örneklerde;

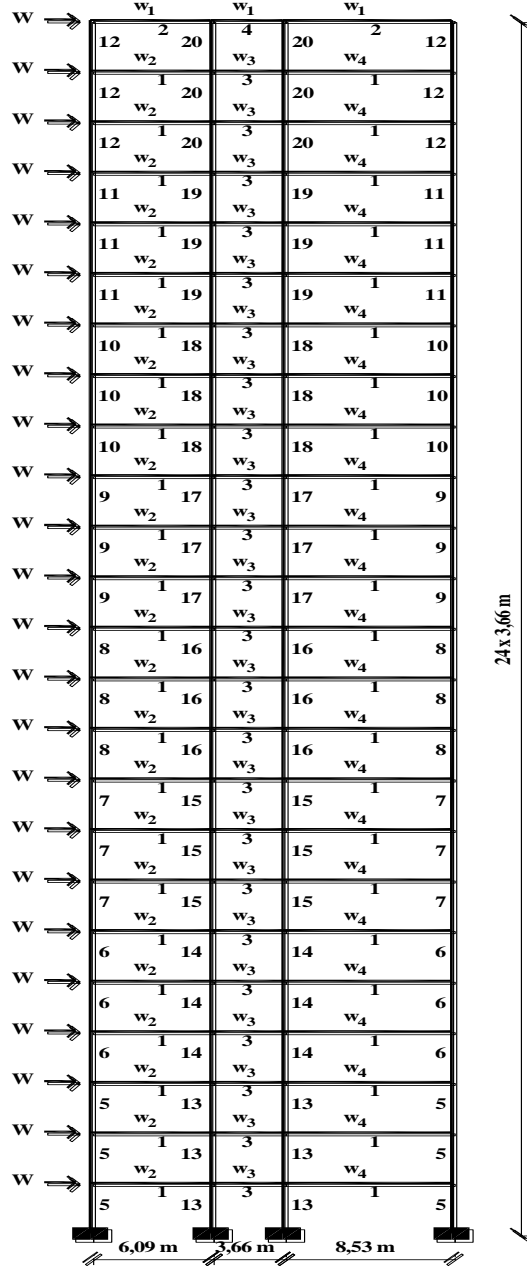
1. Bir kolonideki arı sayısı  $N=50$ ,
2. Maksimum döngü sayısı  $MNC=750$ ,
3. Limit değeri ise  $MNC/3$  alınmıştır.

Sonuçların istatistiksel değişkenlerden etkilenmemesi için her bir örnek program 5 kez çalıştırılmıştır. Sonra da en iyi, en kötü ve ortalama sonuçları tablolar halinde verilmiştir.

Literatürdeki birinci sayısal örnek de İngiliz Birim Sistemi kullanıldığı için burada birim dönüşümleri yapılmadan orijinal halleri ile verilmiş ve standart birim sistemdeki karşılıkları ise parantez içinde verilmiştir.

##### 4.1. Üç Açıklıklı Yirmi Dört Katlı Düzlem Çelik Çerçeve

Şekil 2'de üç açıklıklı yirmi dört katlı düzlem çelik çerçevenin boyutları, yükleme şekli, elemanların grup numaraları gösterilmiştir. Bu çelik çerçeve Camp vd., [10] tarafından Yapay Karınca optimizasyon (ACO) metodu ile, Degertekin [11] tarafından Harmoni Arama (HS) ile, Kaveh ve Talatahari [12] tarafından Geliştirilmiş Karınca Koloni Optimizasyon (IACO) metodu ile ve Togan [13] tarafından Öğretme-Öğrenme Tabanlı Optimizasyon (TLBO) metodu kullanılarak optimum tasarımı yapılmıştır.



Şekil 2. Üç açıklıklı 24 katlı çerçeve sisteminin boyutlandırılması, yükleme şekli ve elemanların gruplandırılması

Çelik çerçeve sistem 100 düğüm noktası ve 168 elemandan oluşturulmuştur. Kirişler dört ayrı grup, kolonlar ise on altı ayrı gruptan oluşmak üzere çelik çerçeve toplam yirmi gruptan oluşmuştur. Çelik çerçeve tek tip yüklemekten oluşmaktadır. Yüklemeler için  $W = 25.63 \text{ kN}$  (5761.85 lb),  $w_1 = 4.38 \text{ kN/m}$  (300 lb/ft) ,

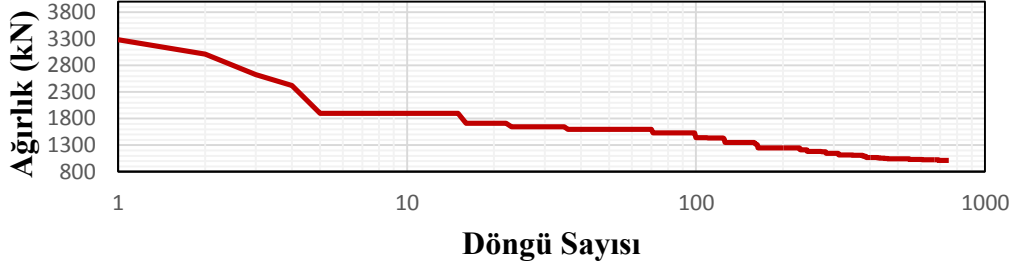


$w_2 = 6.36$  kN/m (436 lb/ft) ,  $w_3 = 6.92$  kN/m (474 lb/ft) ve  $w_4 = (5.95$  kN/m) 408 lb/ft olarak verilmiştir. Malzeme için Elastisite Modülü  $E = 200$  GPa (29732) ksi , akma gerilmesi  $f_y = 248,2$  MPa (33,4 ksi) alınmıştır. Sistemde gerilme kısıtları Amerikan Çelik Yapılar Enstitüsü-Yük ve Dayanım Faktörü Tasarımı (AISC-LRFD)[17] şartnamesinden alınmıştır. Maksimum yer değiştirme sınırı bina yüksekliğinin 300’de birinden daha az olması gerektiğine uygun olarak  $\delta_{ju} = 29,20$  cm (11.52 in) ve katlar arası maksimum yer kat yüksekliğinin 300’de biri olarak  $\Delta_{ju} = 1.216$  cm (0.48 in) olarak kullanılmıştır. Mesnetlenme biçimi ankastredir. Optimum tasarım sonucu minimum tasarım ağırlığı, 1033.7 kN ortalama ağırlık ise 1054.05 kN bulunmuştur. Yapılan program sonucunda yapay arı koloni algoritması için gerilme ihlali çıkmayıp, 0.0082 deplasman ihlali çıkmıştır.

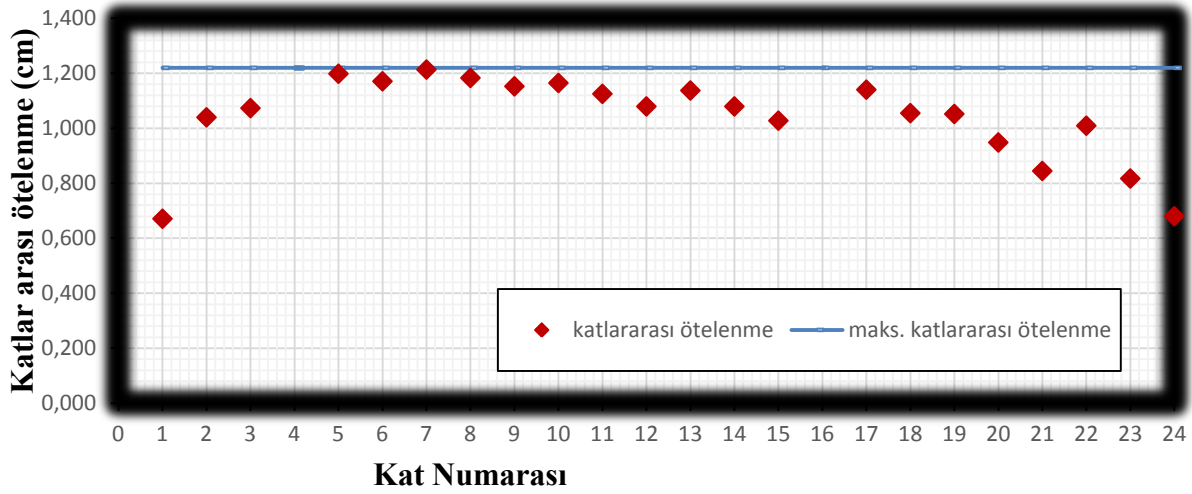
**Tablo 1.** Üç açıklıklı yirmi dört katlı çerçeve sisteminin optimum sonuçları

| Eleman grup no.   | AISC W kesitler |               |                          |            |            |
|-------------------|-----------------|---------------|--------------------------|------------|------------|
|                   | ACO Camp vd.    | HS Degertekin | IACO Kaveh ve Talatahari | TLBO Togan | Bu Çalışma |
| 1                 | W30 × 90        | W30 × 90      | W30 × 99                 | W30 × 90   | W30 × 90   |
| 2                 | W8 × 18         | W10 × 22      | W16 × 26                 | W8 × 18    | W12 × 14   |
| 3                 | W24 × 55        | W18 × 40      | W18 × 35                 | W24 × 62   | W21 × 55   |
| 4                 | W8 × 21         | W12 × 16      | W14 × 22                 | W6 × 9     | W12 × 30   |
| 5                 | W14 × 145       | W14 × 176     | W14 × 145                | W14 × 132  | W14 × 257  |
| 6                 | W14 × 132       | W14 × 176     | W14 × 132                | W14 × 120  | W14 × 145  |
| 7                 | W14 × 132       | W14 × 132     | W14 × 120                | W14 × 99   | W14 × 120  |
| 8                 | W14 × 132       | W14 × 109     | W14 × 109                | W14 × 82   | W14 × 109  |
| 9                 | W14 × 68        | W14 × 82      | W14 × 48                 | W14 × 74   | W14 × 74   |
| 10                | W14 × 53        | W14 × 74      | W14 × 48                 | W14 × 53   | W14 × 53   |
| 11                | W14 × 43        | W14 × 34      | W14 × 34                 | W14 × 34   | W14 × 34   |
| 12                | W14 × 43        | W14 × 22      | W14 × 30                 | W14 × 22   | W14 × 22   |
| 13                | W14 × 145       | W14 × 145     | W14 × 159                | W14 × 109  | W14 × 159  |
| 14                | W14 × 145       | W14 × 132     | W14 × 120                | W14 × 99   | W14 × 132  |
| 15                | W14 × 120       | W14 × 109     | W14 × 109                | W14 × 99   | W14 × 120  |
| 16                | W14 × 90        | W14 × 82      | W14 × 99                 | W14 × 90   | W14 × 109  |
| 17                | W14 × 90        | W14 × 61      | W14 × 82                 | W14 × 68   | W14 × 99   |
| 18                | W14 × 61        | W14 × 48      | W14 × 53                 | W14 × 53   | W14 × 53   |
| 19                | W14 × 30        | W14 × 30      | W14 × 38                 | W14 × 34   | W14 × 53   |
| 20                | W14 × 26        | W14 × 22      | W14 × 26                 | W14 × 22   | W14 × 22   |
| Ağırlık(kN)       | 970.05          | 945,38        | 956,84                   | 893,23     | 1033,70    |
| Ortalama ağ.(kN)  | *               | *             | *                        | *          | 1054,05    |
| Analiz sayısı (#) | 15500           | 14651         | 3500                     | 12000      | 37500      |
| Gerilme ihlali    | 1.680           | 1.858         | 4.714                    | 8.836      | 0          |
| Deplasman ihlali  | 0.053           | 2.457         | 4.846                    | 6.401      | 0.0082     |

\*Değerler önceki çalışmalarda verilmemiştir.



Şekil 3. Üç açıklıklı 24 katlı çerçeve sistemi için ağırlık ve döngü sayısı ilişkisi



Şekil 4. Üç açıklıklı 24 katlı çerçeve için katlar arası yer değiştirmeler

#### 4.2. On Elemanlı Çelik Kafes Sistemi

Şekil 5.'de kafes sisteminin boyutları ve yükleme şekli gösterilmiştir. Bu kafes sistem ayrık tasarım değişkenli olarak genetik algoritma (GA) kullanılarak Rajeev ve Krishnamoorthy [14], tavlama benzetimi (SA) kullanılarak Kripka [15], hiyerarşi parçacık sürü optimizasyonu algoritması (HPSO) kullanılarak Li vd. [16], karınca koloni algoritması (ACO) kullanılarak Camp ve Bichon [10], yapay arı kolonisi algoritması (ABC) kullanılarak Sonmez [4] tarafından çalışılmıştır.

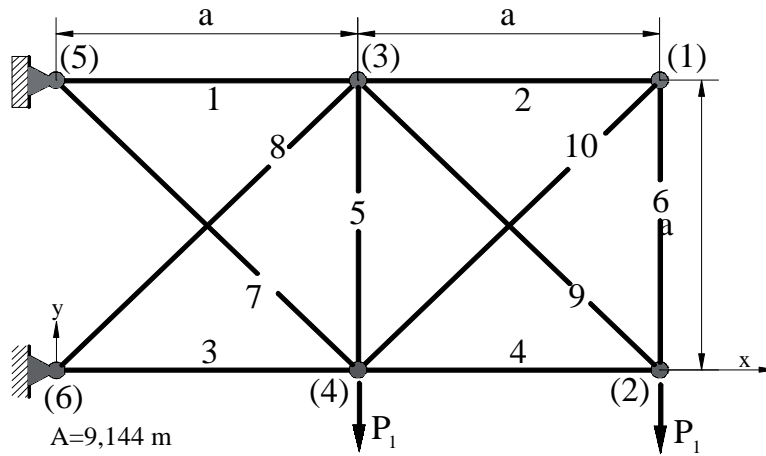
Bu örnek literatürde birçok araştırmacı tarafından kullanılmıştır.  $P_1 = 444,82$  kN (100 kips) olarak verilmiştir. Her bir kesit elemanı için Elastisite Modülü  $E = 68,971$  GPa (10000 ksi) ve malzeme yoğunluğu  $\rho = 2768$  kg/m<sup>3</sup> (0,10 lb/in.<sup>3</sup>) olduğu varsayılmıştır. Her bir eleman için muhtemel en kesit alanlar  $L = \{10.45, 11.61, 12.84, 13.74, 15.35, 16.90, 16.97, 18.58, 18.90, 19.94, 20.19, 21.81, 22.39, 22.90, 23.42, 24.77, 24.97, 25.03, 26.97, 27.23, 28.97, 29.61, 30.97, 32.06, 33.03, 37.03, 46.58, 51.42, 74.19, 87.10, 89.68, 91.61, 100.00, 103.23, 109.03, 121.29, 128.39, 141.94, 147.74, 170.97, 193.55, 216.13\}$  (cm<sup>2</sup>) olmak üzere 42 tasarım değişkeni kullanılmıştır. Her iki yönde olmak üzere her bir düğüm noktasının yer değiştirmesi  $\pm 5.08$  cm (

$\pm 2$  in.) değerinden az olmalıdır. Sistem için müsaade edilen gerilme değeri ise  $\pm 172.25\text{Mpa}$  ( $\pm 25$  ksi) olarak verilmiştir. Gerilme kısıtı ve yer değiştirme kısıtı aşağıdaki gibi ifade edilmiştir.

$$\sigma_{m,l} = \frac{\sigma_{m,l}}{\sigma_{m,müsaade}} - 1 \leq 0, \quad m = 1, 2, \dots, n \quad (19)$$

$$U_{k,l} = \frac{u_{k,l}}{u_{k,müsaade}} - 1 \leq 0, \quad k = 1, 2, \dots, n_n \quad (20)$$

ABC algoritmasının kullanılarak on elemanlı kafes sistemin tasarımı için geliştirilen program, arı sayısı  $N=50$  ve maksimum döngü sayısı  $MNC=750$  alınarak birbirinden bağımsız 5 kere çalıştırılmıştır. Amaç fonksiyonun değerleri elde edilmiştir.

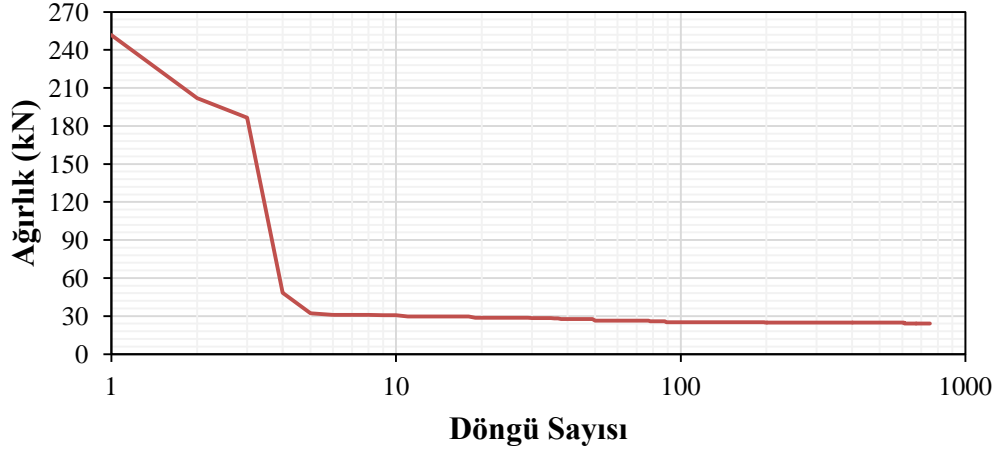


Şekil 5. On elemanlı kafes sistemi

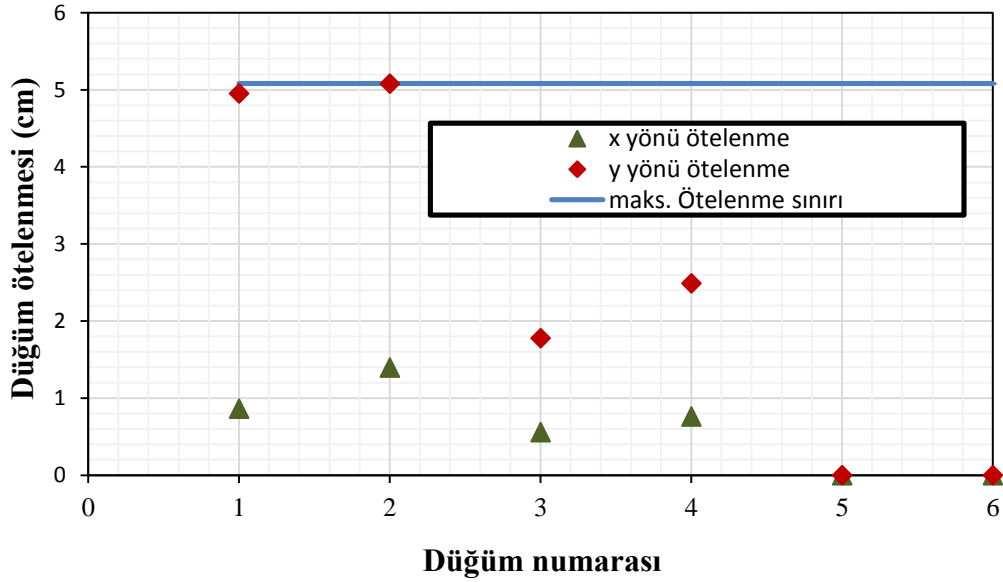
Tablo 2. Çelik kafes sisteminin optimum sonuçları

| Değişkenler       |          | Optimum en kesit alanları ( $cm^2$ ) |        |        |        |        |            |
|-------------------|----------|--------------------------------------|--------|--------|--------|--------|------------|
| Numara            | Değişken | GA                                   | SA     | HPSO   | ACO    | ABC    | Bu Çalışma |
| 1                 | $A_1$    | 216.13                               | 216.13 | 193.55 | 216.13 | 216.13 | 216.13     |
| 2                 | $A_2$    | 10.45                                | 10.45  | 10.45  | 10.45  | 10.45  | 10.45      |
| 3                 | $A_3$    | 141.94                               | 147.74 | 147.74 | 147.74 | 147.74 | 147.74     |
| 4                 | $A_4$    | 100.00                               | 91.61  | 87.10  | 91.61  | 91.61  | 89.68      |
| 5                 | $A_5$    | 10.45                                | 10.45  | 10.45  | 10.45  | 10.45  | 10.45      |
| 6                 | $A_6$    | 10.45                                | 10.45  | 10.45  | 10.45  | 10.45  | 10.45      |
| 7                 | $A_7$    | 91.61                                | 51.42  | 51.42  | 51.42  | 51.42  | 89.68      |
| 8                 | $A_8$    | 128.39                               | 147.74 | 147.74 | 147.74 | 147.74 | 147.74     |
| 9                 | $A_9$    | 128.39                               | 141.94 | 141.94 | 141.94 | 141.94 | 103.23     |
| 10                | $A_{10}$ | 16.90                                | 10.45  | 11.61  | 10.45  | 10.45  | 10.45      |
| En iyisi (kN)     |          | 24,70                                | 24,16  | 24,34  | 24,16  | 24,16  | 24,09      |
| Ortalaması (kN)   |          | *                                    | *      | *      | *      | 24,24  | 24,39      |
| En kötüsü (kN)    |          | *                                    | *      | *      | *      | 24,36  | 25,13      |
| Analiz sayısı (#) |          | *                                    | *      | 50000  | 10000  | 25800  | 35000      |
| Kısıt ihlali      |          | Yok                                  | Yok    | Yok    | Yok    | Yok    | Yok        |

Tablo 2 'de görüldüğü gibi minimum tasarım ağırlığı 24.09 kN, ortalama ağırlık ise 24.39 kN bulunmuştur.

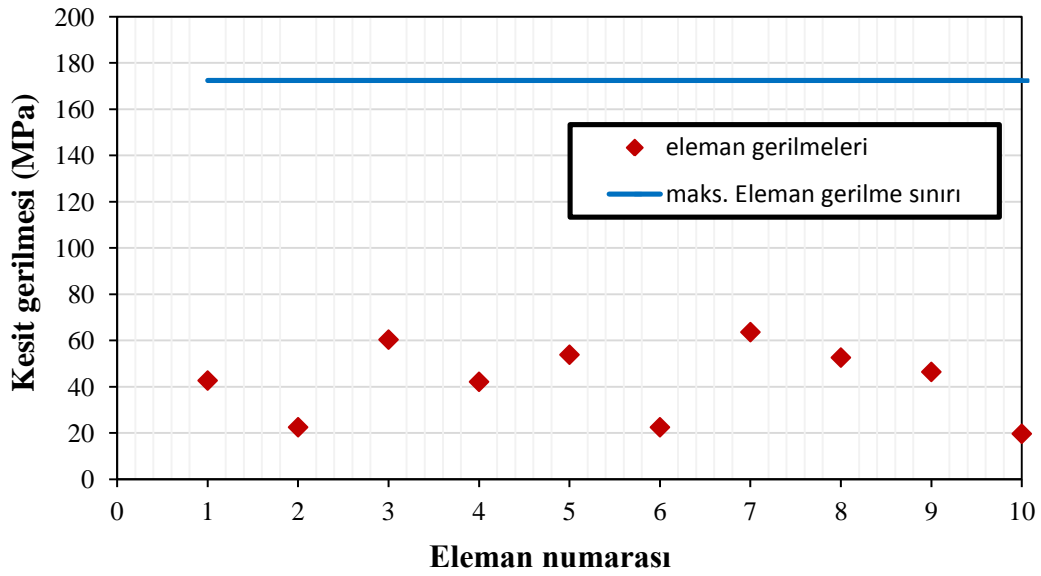


Şekil 6. Çelik kafes sistemi ağırlık ve döngü sayısı ilişkisi



Şekil 7. Çelik kafes sistemi için x ve y yönü için katlar arası yer değiştirmeler

Şekil 7.'de on elemanlı kafes sisteminde düğüm noktalarının x ve y doğrultusundaki yer değiştirilmeleri mutlak değerce gösterilmiştir. Yer değiştirmelerin hepsi müsaade edilen 5,08 cm ( $\pm 2$ ) in değerinden küçük veya eşit olduğu gözlenmiştir.



Şekil 8. Çelik kafes sistemi için eleman gerilmeleri

Şekil 8’ de kafes için her bir elemandaki gerilme değerleri verilmiştir. Gerilme değerlerinin tamamı müsaade edilen  $\pm 25$  ksi değerinden düşük olduğu gözlemlenmiştir.

## 5. TARTIŞMALAR VE SONUÇLAR

Bu çalışmada direk arama optimizasyon algoritmalarından birisi olan yapay arı koloni algoritmasının (ABC) ayrık değişkenli iki ve üç boyutlu çerçevelerin optimum tasarımı için nasıl kullanılacağı üzerinde durulmuştur. ABC algoritması arıların yiyecek arama davranışlarına esinlenerek geliştirilmiş bir sezgisel direkt arama metodudur. Yapay arı koloni (ABC) algoritması nesne tabanlı programlama teknoloji kullanılarak C# programlama dili ile kodlanmıştır. Hazırlanan programa Sap2000 yapı analiz programı entegre edilmiş olup çubuk kuvvetleri ve deplasmanlar bu program yardımı ile hesaplanmıştır.

Literatürde sunulan üç açıklıklı yirmi dört katlı düzlem çerçeve yapay arı koloni (ABC) algoritması kullanılarak ağırlıkları minimize edilmiştir. Son olarak optimum tasarımı için modellenmesi zor olan bir uzay çelik çerçeve (tek açıklık ve dört katlı) sunularak yapay arı koloni algoritması ile ağırlıkları minimize edilmiştir. Tasarım kısıtları göz önüne alınarak yapılan optimum tasarım sonuçları yapay arı koloni (ABC) algoritmasının başarılı bir optimizasyon yöntemi olduğu gözlemlenmiştir.

İki tasarım örneğinin ABC algoritması ile optimum tasarımı sonucunda aşağıdaki sonuçlar elde edilmiştir.

1. ABC sonuçları ile diğer algoritmaların sonuçları karşılaştırılması sonucunda yapay arı koloni algoritmasının diğer optimizasyon algoritmalarına göre daha iyi sonuçlar verdiği görülmüştür.
2. Yapay arı koloni algoritması dikkat çekici bir yönü de lokal minimumlarda takılmayarak %100 başarı ile global optimuma yaklaşmıştır.
3. Yapay arı koloni algoritması ile en iyi tasarımı bulmak için yapısal analiz sayısı ve yakınsama hızı açısından diğer yöntemlerle karşılaştırıldığı zaman belirgin bir iyileşme gözlenmemiştir.
4. Kısıtlar için statik penaltı fonksiyonu kullanılmıştır. Statik penaltı değeri olarak 10 gibi yüksek bir değer seçilerek Deb’in [18] seçim mekanizmasına yakın bir algoritma oluşturulmuştur. Başka penaltı fonksiyonları kullanılarak yapay arı koloni algoritmasının yakınsama hızını artırabilir.

Bu sonuçlara göre ayrık değişkenli kafes sistemlerin, düzlem ve uzay çerçevelerin optimum tasarımı için yapay arı koloni algoritmasının başarılı bir uygulama olduğu görülmüştür. Literatürde bulunan çalışmalardan alınan ayrık değişkenli ve uygun geometriye sahip tasarım örnekleri, yapay arı koloni algoritması ile çok etkili bir şekilde çözülmüştür.

İlerdeki çalışmalarda ABC algoritmasının daha az sayıda yapısal çözümleme yapacak şekilde modellenmesine çalışılabilir. Analizlerde Sap2000 programı kullanıldığı için çok farklı tipteki yapıların optimizasyonu hazırlanan program ile yapılabilir. İlerdeki çalışmalarda sadece kesit optimizasyonu değil aynı zamanda şekil ve topoloji optimizasyon problemleri de çözülebilir.

## 6. KAYNAKLAR

- [1] KARABOĞA, D., “An idea based on honey bee swarm for numerical optimization”, Technical Report TR06, Erciyes University, Engineering Faculty, Computer, Engineering Department, 2005.
- [2] KARABOĞA, D and AKAY B., “On the performance of artificial bee colony (ABC) algorithm”, *Applied Soft Computing*, 8(1), 687-697, 2008.
- [3] SONMEZ, M., “Artificial Bee Colony algorithm for optimization of truss structures”, *Applied Soft Computing*, 2406-2418, 2011.
- [4] SONMEZ, M., “Discrete optimum design of truss structures using artificial bee colony algorithm”, *Struct Multidisc Optim*, 85-97, 2011.
- [5] DEGERTEKİN, S.O., “Improved harmony search algorithms for sizing optimization of truss structures”, *Computers and Structures*, 229-241, 2012.
- [6] AYDOĞDU, I., Akın, A. and Saka, M.P., “Optimum design of steel space frames by artificial bee colony algorithm”, 10<sup>th</sup> International Congress on Advances in Civil Engineering, Ankara, Turkey, 2012.
- [7] SEVİM, O., “Yapay Arı Koloni Algoritması ile 3 boyutlu çelik çerçevelerin optimum tasarımı” Aksaray Üniversitesi, Fen Bilimleri Enstitüsü, Aksaray, 2013.
- [8] AKAY, B., “Nümerik optimizasyon problemlerinde yapay arı koloni (artificial bee colony) algoritmasının performans analizi”, Doktora Tezi Kayseri Erciyes Üniversitesi, Fen Bilimleri Enstitüsü, Kayseri, 2009.
- [9] BONABEAU, E., DORIGO, M. and THERAULAZ, G., *Swarm intelligence: from natural to artificial systems*, Oxford University Press, Inc., New York, NY, USA, 1999.
- [10] CAMP, C.V. ve BICHON, B.J., “Design of space trusses using ant colony optimization”, *J Struct Eng*, 1245-1287, 2004.
- [11] DEGERTEKİN, S.O., “Optimum design of geometrically non-linear steel frames using artificial bee colony algorithm”, *Steel and Composite Structures*, 505-522, 2012.
- [12] KAVEH, A. and TALATAHARI, S., “An improved ant colony optimization for the design of planar steel frames”, *Engineering Structures*, 864-873, 2010.
- [13] TOGAN, V., “Design of planar steel frames using Teaching–Learning Based Optimization”, *Engineering Structures*, 225-232, 2012.
- [14] RAJEEV S. and KRISHNAMOORTHY, C.S., “Discrete optimization of structures using genetic algorithms”, *J Struct Eng, ASCE*, 118(5), 233-1251, 1992.
- [15] KRIPKA, M., 2004, *Discrete optimization of trusses by simulated annealing*, J. Braz. Soc. Mech. Sci. & Eng. vol.26 no.2 Rio de Janeiro.
- [16] LI, L.J., HUANG, Z.B. and LIU, F., A heuristic particle swarm optimization method for truss structures with discrete variables, *Comput Struct*, 87,435-444, 2009.
- [17] ASIC, American Institute of Steel Construction Load & Resistance Factor Design, Volume 1, Second Edition, 1995.
- [18] DEB, K., “An efficient constraint handling method for genetic algorithms”, *Comput Methods Appl Mech Eng*, , 186(2–4), 311–338, 2000.