

Yapay Arı Koloni Algoritması İle Uzay Çelik Yapıların Optimum Tasarımı

*¹Özer Sevim, ²Mustafa Sönmez, ³Rıfat Sezer

¹İnşaat Mühendisliği Bölümü, Mühendislik Fakültesi, Kırıkkale Üniversitesi, Türkiye.

²İnşaat Mühendisliği Bölümü, Mühendislik Fakültesi, Aksaray Üniversitesi, Türkiye.

³İnşaat Mühendisliği Bölümü, Mühendislik Fakültesi, Selçuk Üniversitesi, Türkiye.

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 uzay çelik çerçevelerin ayrık optimizasyonu sunulmuştur. ABC algoritmasının kullanma bilirliliğini gösterilmesi için uzay çelik çerçeve içeren yapısal örnek çözülmüştür. Sonuçlarla uzay 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, Sürü Zekâsı, C# Programlama

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 space steel frames. In order to demonstrate the effectiveness of algorithm, including space frame problem were solved. The results demonstrate that the artificial bee colony algorithm (ABC) is very effective and robust for space steel frame design.

Keywords: Artificial Bee Colony Algorithm, Discrete Optimization, Steel Frame Design, Swarm Intelligence, C# Programming

1. Giriş

İnsan nüfusunun hızla artması, yaşanılacak yerlerin sınırlı olması insanları farklı arayışlara yönlendirmiştir. Bu arayışlar ile birlikte teknolojinin de gelişmesiyle en yüksek, en dayanıklı ve en ekonomik fonksiyonel yapıların tasarlanması mühendislerin en önemli amaçlarından birisi olmuştur. İnşaat mühendisliğinde ön tasarımın 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. Bu yinelemeli işlemlerin bir yöntem (metodoloji) kullanılarak yapılmasına optimizasyon denir.

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

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 uzay çerçeve sistemlerin optimum tasarımlarını yapacak bilgisayar programı geliştirmek ve ABC'nin çelik çerçevelerin tasarımı için etkili bir yöntem olduğunu ortaya koymaktır. Yapay arı koloni algoritması (ABC-Artificial Bee Colony) 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 Uzay çelik çerçeve sistemlerin tasarımında tasarım değişkenleri olarak sadece eleman kesitleri ele alınmıştır. Profiller hazır profil tablolarından seçilecektir. Bu çalışmada ticari hazır paket programlarından SAP2000 yapı analiz programı hazırlanan bilgisayar programına entegre edilmiştir. Programlama dili olarak C# programı seçilmiştir. 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 zekâsı adı verilen birlikte davranış zekâsının ortaya çıkmasına sebep olmaktadır. Kendi kendine organize olabilme sürü sisteminin temel özelliğidir[8]. Bonabeau [9] kendi kendine organize olabilmeyi dört özellik ile karakterize etmiştir.

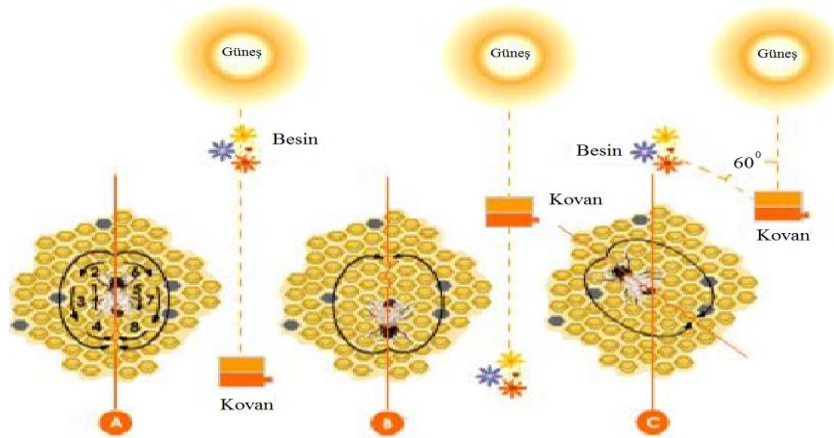
1. Pozitif geri besleme: Elverişli durumların oluşmasını destekleyen davranışsal kurallar sayesinde oluşur. Pozitif geri beslenmeye örnek olarak arıların dans etmeleri ile birlikte diğer arıların bir işe yönlendirilmeleri verilebilir.

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.

Bu özellikler araştırmacıları arıların davranışlarını modellemeleri yönünde teşvik edici olmuştur.

2.2. Doğadaki arıların davranışı

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 kovana, üç gıda kaynağı ve bu gıda kaynağını keşfeden arılar tarafından gerçekleştirilen sallanma dansı şematik olarak gösterilmiştir.



Şekil 1. Bal arılarının sallanma dansı (Waggle Dance)

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 kolonisi 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, λ 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 veriler 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 fonksiyonun değerinin hesaplaması 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.

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:

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

Yukarıdaki eşitlikte, ϕ 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 arılarının arama faaliyetlerini kontrol etmek için kullanılır. ${}^{isth}x_j$ ABC yakınsama davranışlarında önemli bir rol oynar. Bulduğu yerde besin seviyesi önceki en iyi besin kaynağına göre daha iyi ise, bulunduğu konum iyi besin kaynağı özelliği kazanır yani ${}^{isth}I_j = {}^{yeni}I_j$ olur. Aksi takdirde, eski konumu en iyi besin kaynağı olarak korunur.

3. Sayısal Örnek

Yapay arı koloni algoritmasının etkinliğini göstermek için aşağıda belirtilen tasarım örneği kullanılmıştır. Bu örnek;

1. Dört yükleme tipine maruz kalan tek açıklıklı, 4 katlı ve 32 elemandan oluşan uzay çelik çerçeve yapısı.

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 sınır şartlarını, 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 sayısı 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.

3.1. Tek açıklıklı dört katlı uzay çelik çerçeve

Şekil 2.'de tek açıklıklı dört katlı uzay çelik çerçevenin boyutları gösterilmiştir. Bu çelik çerçeve ilk kez bu çalışmada kullanılarak optimum tasarım için kullanılmıştır. Gerilme kıstasları Amerikan Çelik Yapılar Enstitüsü - Yük ve Dayanım Faktörü Tasarımı (AISC-LRFD, 2001) [14] şartnamesinden alınmıştır. Katlar arası yer değiştirme kısıtı ise $\delta < h/300$ şeklinde alınmıştır. Maksimum yer değiştirme kısıtı $\Delta < H/400$ şeklinde alınmıştır (Δ çelik çerçeve üzerinde oluşan maksimum yer değiştirme). Kiriş ve kolon elemanları için LRFD[14] 'de sunulan 267 adet W kesitlerinden 64 adet W kesiti seçilmiştir. Seçilen kesitler aşağıda verilmiştir.

$L=\{W8 \times 15, W8 \times 18, W8 \times 21, W8 \times 24, W8 \times 28, W8 \times 31, W8 \times 35, W8 \times 40, W8 \times 48, W8 \times 58, W8 \times 67, W10 \times 12, W10 \times 15, W10 \times 17, W10 \times 19, W10 \times 22, W10 \times 26, W10 \times 30, W10 \times 33, W10 \times 39, W10 \times 45, W10 \times 49, W12 \times 14, W12 \times 16, W12 \times 19, W12 \times 22, W12 \times 26, W12 \times 30, W12 \times 35, W12 \times 40, W12 \times 45, W12 \times 50, W12 \times 53, W12 \times 58, W14 \times 22, W14 \times 26, W14 \times 30, W14 \times 34, W14 \times 38, W14 \times 43, W14 \times 48, W14 \times 53, W14 \times 61, W14 \times 68, W16 \times 26, W16 \times 31, W16 \times 36, W16 \times 40, W16 \times 45, W16 \times 50, W16 \times 57, W16 \times 67, W18 \times 35, W18 \times 40, W18 \times 46, W18 \times 50, W18 \times 55, W21 \times 44, W21 \times 50, W21 \times 57, W21 \times 62, W24 \times 55, W24 \times 62, W24 \times 68\}$. Çelik çerçeve sistem 20 düğüm noktası ve 32 elemandan oluşturulmuştur. Bu uzay çelik çerçeve altı gruptan oluşmaktadır. Kolonlar iki ayrı gruptan oluşmuştur. Kirişler ise dört ayrı gruptan

oluşturmuştur. İlk iki katın tüm kolonları birinci grubu, üç ve dördüncü kattaki tüm kolonlar ise ikinci grubu oluşturmaktadır. Bir ve ikinci kat x doğrultusundaki tüm kirişler üçüncü grubu, bir ve ikinci kattaki y doğrultusundaki tüm kirişler dördüncü grubu, üç ve dördüncü kattaki x doğrultusundaki tüm kirişler beşinci grubu, üç ve dördüncü kattaki y doğrultusundaki tüm kirişler ise altıncı ve son grubu oluşturmaktadır.

Uzay çelik çerçeve ölü yük (D), hareketli yük (L), çatı hareketli yükü (L_r) ve rüzgâr yükü (W) içeren I: 1.4D, II: 1.2D + 1.6L + 0.5L_r, III: 1.2D + 1.6L_r + 0.5L, IV: 1.2D + 1.3W + 0.5L + 0.5L_r kombinasyonlarından oluşan dört farklı tip yükleme ile yüklenmiştir. Bu yüklemeler ve kısıtlar Amerikan Çelik Yapılar Enstitüsü - Yük ve Dayanım Faktörü Tasarımı (AISC-LRFD)[14] şartnamesinden alınmıştır.

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

$$\beta_{il}(I) = \left(\frac{P_u}{\phi P_n} \right)_{il} + \frac{8}{9} \left(\frac{M_{ux}}{\phi_b M_{nx}} + \frac{M_{uy}}{\phi_b M_{ny}} \right)_{il} - 1.0 \quad i = 1, \dots, nm, l = 1, \dots, nl \quad (5)$$

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

$$\beta_{il}(I) = \left(\frac{P_u}{2\phi P_n} \right)_{il} + \left(\frac{M_{ux}}{\phi_b M_{nx}} + \frac{M_{uy}}{\phi_b M_{ny}} \right)_{il} - 1.0 \quad i = 1, \dots, nm, l = 1, \dots, nl \quad (6)$$

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 eksendeki eğilme momenti, M_{nx} kuvvetli eksendeki taşıyabileceği maksimum eğilme momentidir. M_{uy} geometrik olarak lineer olmayan yani ikinci dereceden etkileri içeren zayıf eksendeki eğilme momentidir (iki boyutlu çerçevelerde $M_{uy} = 0$ dir). M_{ny} zayıf eksendeki taşıyabileceği maksimum eğilme momenti, ϕ ise dayanım azaltma katsayısıdır, basınç için 0.85, çekme için 0.90 ve eğilme için 0.90 olur.

Eğer Denklem 5 ve 6 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 7' de verilmiştir.

$$P_n = A_g \times F_{cr} \quad (7)$$

$$F_{cr} = (0.658^{\lambda_c^2}) F_y \quad \lambda_c \leq 1.5 \text{ ise} \quad (8)$$

$$F_{cr} = \left(\frac{0.877}{\lambda_c^2} \right) F_y \quad \lambda_c > 1.5 \text{ ise} \quad (9)$$

$$\lambda_c = \frac{KL}{r\pi} \sqrt{\frac{F_y}{E}} \quad (10)$$

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 11 ile hesaplanabilir.

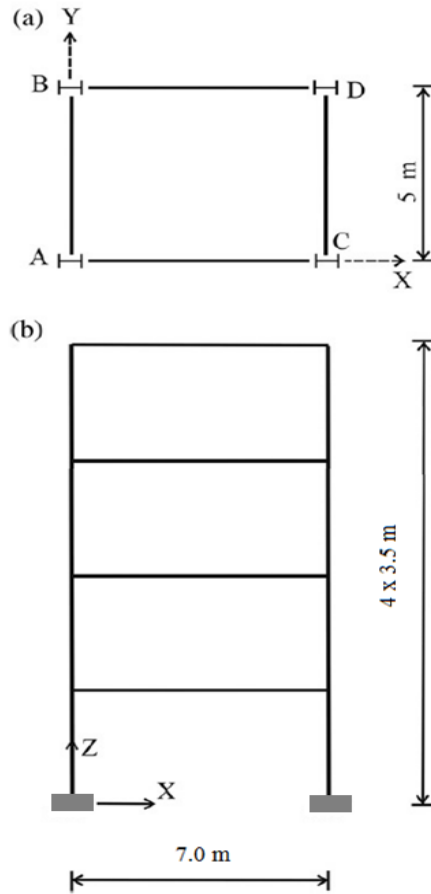
$$K = \sqrt{\frac{1.6G_A G_B + 4.0(G_A + G_B) + 7.50}{G_A + G_B + 7.50}} \quad (11)$$

G_A ve G_B sınırlama faktörü Denklem 12 ile hesaplanabilir.

$$G = \frac{\sum(I_c/L_c)}{\sum I_g/L_g} \quad (12)$$

Lineer olmayan analiz için Denklem 13 ile ikinci dereceden etkiler göz önüne alınmıştır. Kiriş ve kolon elemanları LRFD[14]' de bulunan 267 adet W kesitlerinden seçilmiştir[14].

$$M_u = \beta_1 \times \beta_2 \times M \quad (13)$$



Şekil 2. Dört katlı otuz iki elemanlı uzay çerçeve: (a) plan, (b) yandan görünüş

Dört farklı yükleme için ölü yük (D) = 0.4 kPa, hareketli yük (L) = 0.3 kPa, çatı hareketli yükü (L_r) = 0.2 kPa ve rüzgar yükü (W) Uniform Building Şartnamesine göre $p = C_e \times C_q \times q_s \times I_w$ formülü ile hesaplanmıştır. Burada p rüzgar basıncını, C_e yüksekliğe bağlı rüzgar

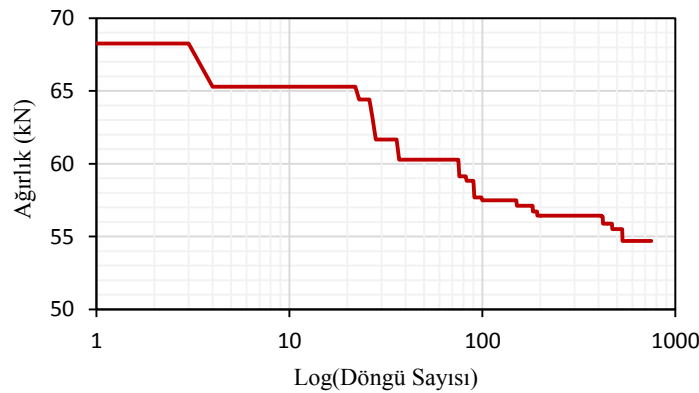
faktörü, C_q basınç katsayısı, q_s rüzgarın durgunluk katsayısı, I_w ise rüzgar önem katsayısıdır. Uzay çelik çerçevenin D yüklemesine maruz kaldığı varsayıldığında C_e katsayısı Uniform Building Şartnamesi (Tablo 16-G) 14 m bina yüksekliğine karşı gelen değer olarak 1,653 alınmıştır. C_q basınç için 0,8 emme için $-0,5$ alınmıştır. Rüzgar önem katsayısı I_w 1 olarak alınmıştır. Rüzgâr durgunluk katsayısı q_s ise 129 km/sa temel rüzgar hızı için 0,785 kPa seçilmiştir.

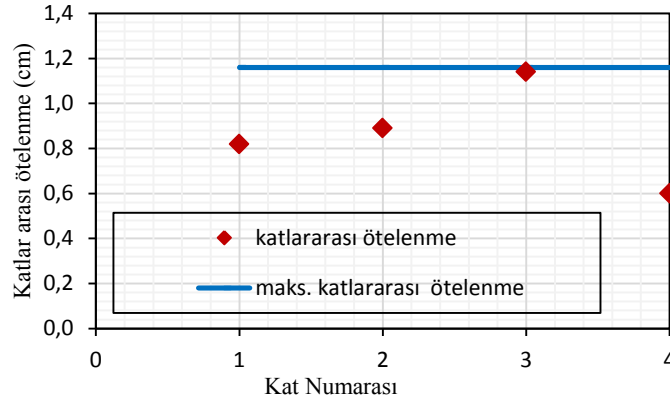
Program birbirinden bağımsız olarak beş kere çalıştırılmıştır, optimum tasarım için en iyi sonuçlar Tablo 1 'de gösterilmiştir. Malzeme için Elastisite Modülü $E = 200$ GPa, malzeme akma gerilmesi $f_y = 248,2$ MPa alınmıştır. Maksimum yer değiştirme sınırı 3,5 cm ve katlar arası maksimum yer değiştirme sınırı tüm katlar için 1,16 cm olarak verilmiştir. Mesnetlenme biçimi ankastre verilmiştir.

Tablo 1. Dört katlı otuz iki elemanlı uzay çerçevenin optimum sonuçları

Grup sayısı	ABC Bu çalışma
1	W 16 × 26
2	W 12 × 16
3	W 21 × 44
4	W 12 × 14
5	W 16 × 26
6	W 10 × 12
Ağırlık (kN)	54,69
En üst kat ötelenme (cm)	3,45
Katlar arası maks. Ötelenme (cm)	1,14
Ortalama ağırlık (kN)	54,97
Analiz sayısı (#)	37500
Standart sapma	0,28

Tablo 1 'de görüldüğü gibi minimum tasarım ağırlığı 54,69 kN, ortalama ağırlık ise 54,97 kN, en üst kat ötelenmesi 3,45 cm, katlar arası maksimum yer değiştirme, 1,14 cm, standart sapma ise 0,28 bulunmuştur.





Şekil 4. Dört katlı otuz iki elemanlı uzay çerçeve için katlar arası yer değiştirmeler

4. Tartışma ve Sonuçlar

Bu çalışmada direk arama optimizasyon algoritmalarından birisi olan yapay arı koloni algoritmasının (ABC) ayrık değişkenli üç boyutlu çerçevenin optimum tasarımı için nasıl kullanılacağı üzerinde durulmuştur. 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.

Optimum tasarımı için modellenmesi zor olan 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 iyi sonuçlar verdiği görülmüştür. SAP2000 ile çalışabilmiştir.
2. Yapay arı koloni algoritması dikkat çekici bir yönü de lokal minimumlarda takılmayarak %100 başarı ile global optimuma yaklaşmıştır.
3. Kısıtlar için statik penaltı fonksiyonu kullanılmıştır. Statik penaltı değeri olarak 10 gibi yüksek bir değer seçilerek Deb'in [15] seçim mekanizmasına yakın bir algoritma oluşturulmuştur. Başka penaltı fonksiyonları kullanılarak yapay arı koloni algoritmasının yakınsama hızını artırabilir.

Bu sonuçlara göre ayrık değişkenli uzay çerçevelerin optimum tasarımı için yapay arı koloni algoritmasının başarılı bir uygulama olduğu görülmüştür. İlerdeki çalışmalarda 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.

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 ve Akay B., “On the performance of artificial bee colony (ABC) algorithm”, *Applied Soft Computing*, 2008, 8(1), 687-697.
- [3] Sonmez, M., “Artificial Bee Colony algorithm for optimization of truss structures”, *Applied Soft Computing*, 2011, 2406-2418.
- [4] Sonmez, M., “Discrete optimum design of truss structures using artificial bee colony algorithm”, *Struct Multidisc Optim* 2011, 85-97.
- [5] Degertekin, S.O., “Improved harmony search algorithms for sizing optimization of truss structures”, *Computers and Structures*, 2012, 229-241.
- [6] Aydogdu, I., Akın, A. and Saka, M.P., “Optimum design of steel space frames by artificial bee colony algorithm”, 10th International Congress on Advances in Civil Engineering, Ankara, Turkey, 2012.
- [7] Sevim, O., ”Yapay Arı Koloni Algoritması ile 3 boyutlu çelik çerçevelerin optimum tasarımı” Aksaray Üniversitesi, Fen Bilimleri Enst. Aksaray, 2013.
- [8] Akay, B., “Nümerik optimizasyon problemlerinde yapay arı koloniisi (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*, 2004, 1245-1287.
- [11] Degertekin, S.O., “Optimum design of geometrically non-linear steel frames using artificial bee colony algorithm”, *Steel and Composite Structures*, 2012, 505-522.
- [12] Kaveh, A. ve Talatahari, S., “An improved ant colony optimization for the design of planar steel frames”, *Engineering Structures*, 2010, 864-873.
- [13] Togan, V., “Design of planar steel frames using Teaching–Learning Based Optimization”, *Engineering Structures*, 2012, 225-232.
- [14] ASIC, American Institute of Steel Construction Load & Resistance Factor Design, Volume 1, Second Edition, 1995.
- [15] Deb, K., “An efficient constraint handling method for genetic algorithms”, *Comput Methods Appl Mech Eng*, 2000, 186(2–4), 311–338.