



XVIII. ULUSAL MEKANİK KONGRESİ
26 - 30 Ağustos 2013, Celal Bayar Üniversitesi, Manisa

KAFES YAPILARIN MODİFİYE EDİLMİŞ YAPAY ARI KOLONİ ALGORİTMASI İLE OPTİMİZASYONU

S. Özgür Değertekin¹, Mehmet Ülker², M. Sedat Hayalioğlu¹

¹Dicle Üniversitesi, Mühendislik Fakültesi, İnşaat Mühendisliği Bölümü, 21280, Diyarbakır

²Fırat Üniversitesi, Mühendislik Fakültesi, İnşaat Mühendisliği Bölümü, 23119, Elazığ

ÖZET

Bu çalışmada, kafes yapıların optimizasyonunda modifiye edilmiş yapay arı koloni algoritması (MABC) kullanılmıştır. Yapay arı koloni algoritması (ABC); arı kolonilerindeki çalışma mekanizması ve arıların nektar elde etmek için izledikleri stratejilerin taklit edilmesini esas alan sezgisel optimizasyon yöntemidir. Optimizasyon probleminin formülasyonunda; deplasman ve gerilme sınırlayıcıları altında minimum ağırlıklı kafes yapı'nın elde edilmesi amaçlanmaktadır. Yöntemin etkinliği daha önce standart yapay arı koloni algoritması ve farklı sezgisel optimizasyon yöntemleriyle kullanılarak optimum tasarımı yapılmış 200 elemanlı, 29 tasarım değişkenli kafes yapı üzerinde test edilmiştir. Yapılan kıyaslamalar; modifiye edilmiş yapay arı koloni algoritmasının, standart yapay arı koloni algoritması ve diğer optimizasyon yöntemlerine göre daha iyi bir tasarım elde ettiğini göstermiştir.

Anahtar Kelimeler: Kafes yapılar, Modifiye edilmiş yapay arı koloni algoritması, Optimizasyon

GİRİŞ

Sezgisel optimizasyon yöntemleri son yıllarda farklı mühendislik problemlerinin çözümünde yaygın olarak kullanılmaktadır. Bu yöntemlerin ana felsefesi; ele alınan problem ile doğadaki bir süreç veya yaşam biçimi arasında benzerlik kurmaktır. Son yirmi yılda giderek artan bir şekilde rağbet gören sezgisel optimizasyon yöntemlerinin en önemlileri arasında genetik algoritmalar, tabu arama, tavlama benzeşimi, armoni arama, parçacık küme optimizasyonu ve yapay arı koloni algoritması sayılabilir.

Genetik algoritmalar; doğada güçlü bireylerin hayatta kalması esasını hesaplamalı yöntemlere uygulamaktadır [1]. Tabu arama; insan beyninin kısa ve uzun süreli hafıza çalışma

mekanizmalarını taklit etmektedir [2,3]. Tavlama benzeşimi; katı cisimlerdeki tavlama işlemi ile optimizasyon problemleri arasında benzeşim kurmaktadır [4]. Armoni arama; müzisyenlerin en iyi armoniyi bulmak için uyguladıkları işlemleri optimizasyon problemlerinde optimum çözümü bulmaya uygulamaktadırlar [5]. Parçacık küme optimizasyonu; hayvan sürülerinin davranışları ile optimizasyon problemleri arasında benzerlik kurmaktadır [6].

Yapay arı koloni algoritması; yukarıda sayılan yöntemlerden daha sonra geliştirilen nispeten yeni bir sezgisel optimizasyon yöntemi olup arı kolonilerindeki çalışma mekanizması ve arıların nektar elde etmek için izledikleri stratejilerin taklit edilmesi esasına dayanır. Yöntem ilk olarak nümerik optimizasyon problemleri için kullanılmış [7] ve kısa sürede farklı uygulama alanları bulmuştur [8-10].

Bu çalışmanın amacı, standart arı koloni algoritması (ABC) üzerinde birtakım değişiklikler yaparak modifiye edilmiş arı koloni algoritması (MABC) adı verilen daha güçlü ve daha hızlı bir yöntem geliştirmektir. Yöntemin etkinliğini test etmek için literatürde standart ABC ve farklı sezgisel optimizasyon yöntemleriyle optimize edilmiş 200 elemanlı, 29 tasarım değişkenli düzlem bir kafes yapı kullanılmış ve elde edilen sonuçlar kıyaslanmıştır.

OPTİMUM TASARIM PROBLEMİ

Kafes yapıların optimizasyon problemi şu şekilde formüle edilebilir:

$$\min W(x) = \sum_{i=1}^{ng} x_i \sum_{k=1}^{mk} \rho_k L_k \quad (1)$$

$$x = [x_1, x_2, \dots, x_{ng}] \quad x_i^l \leq x \leq x_i^u, \quad i=1,2,\dots,ng \quad (2)$$

burada x tasarım değişkenlerini içeren bir vektör, x_i ; i 'nci tasarım değişkenine ait enkesit değeri, ng tasarım değişkenlerinin sayısı (kafes yapıdaki grup sayısı), x_i^l ve x_i^u enkesitlerin alabileceği en küçük ve en büyük değerler, mk ; k 'nci gruptaki toplam eleman sayısı, $W(x)$ kafes yapının ağırlığı, ρ_k ve L_k ; i 'nci gruptaki k 'nci elemana ait özgül ağırlık ve uzunluk değerleridir. Normalize edilmiş tasarım sınırlayıcıları şu şekilde ifade edilebilir:

$$g_{nl}^s(x) = \frac{|\sigma_{nl}|}{|\sigma_{nu}|} - 1 \leq 0, \quad 1 \leq n \leq nm, \quad 1 \leq l \leq nl \quad (3)$$

$$g_{jl}^d(x) = \frac{|d_{jl}|}{|d_{ju}|} - 1 \leq 0, \quad 1 \leq j \leq ndof, \quad 1 \leq l \leq nl \quad (4)$$

bu bağıntılarda $g_{nl}^s(x)$ ve $g_{jl}^d(x)$ sırasıyla eleman gerilme ve düğüm deplasman sınırlayıcılarına ait fonksiyonlar, σ_{nl} l 'nci yükleme durumunda n 'nci elemandaki gerilme değeri, σ_{nu} herhangi bir elemandaki gerilmenin alabileceği en büyük değer, d_{jl} l 'nci yükleme durumunda j 'nci serbestlik derecesi için düğüm noktasındaki deplasman değeri, d_{ju}

herhangi bir düğüm noktasındaki deplasmanın alabileceği en büyük değer, nl yükleme durumu sayısı, $ndof$ serbestlik derecesi sayısıdır.

Optimizasyon süreci sonunda (3) ve (4) bağıntılarıyla verilen sınırlayıcıları sağlayan farklı kafes yapılardan minimum ağırlıklı olan nihai optimum tasarım olarak tayin edilir. Bu süreçte, (3) ve (4) bağıntılarıyla verilen sınırlayıcıları sağlamayan tasarımlar elenir. Bu eleme işlemi için daha önce kafes yapıların optimizasyonuna başarıyla uygulanmış bir seçim mekanizması kullanılmıştır [11]. Bu seçim mekanizması şu kuralları içerir:

- Sınırlayıcıları sağlayan kafes yapı tasarımları, sınırlayıcıları sağlamayanlara tercih edilir,
- Sınırlayıcıları çok küçük bir değerde (optimizasyon sürecinin başında 0,01 optimizasyon sürecinin sonunda 0,001 mertebesinde) sağlamayan tasarımlar sınırlayıcıları sağlıyor olarak kabul edilir,
- Sınırlayıcıları sağlayan iki kafes yapı tasarımı arasında daha hafif olan kafes yapı tasarımı tercih edilir,
- Sınırlayıcıları sağlamayan iki kafes yapı tasarımı arasında sınırlayıcıları daha az ihlal eden tasarım tercih edilir.

Bu seçim mekanizmasında (a) ve (d) kuralları optimizasyon işleminin sınırlayıcıları sağlayan tasarımlara yönelmesini sağlamakta, (c) kuralı sınırlayıcıları sağlayan tasarımlar içinde daha iyi (daha hafif) olan tasarıma yönlendirme yapmakta, (b) kuralı ise global optimumun sınırlayıcılara ait sınır değerlere yakın olması olasılığını araştırmaktadır. Burada önemle vurgulanması gereken nokta, (b) kuralı uygulanmakla birlikte optimizasyon süreci sonunda tayin edilen optimum tasarım her zaman için sınırlayıcıları sağlayan tasarım olacaktır. Çok düşük değerde dahi olsa sınırlayıcıları sağlamayan tasarımın optimum tasarım olarak atanması kabul edilmeyecektir.

MODİFİYE EDİLMİŞ ARI KOLONİ ALGORİTMASI (MABC)

Herhangi bir arı kolonisinde işçi arılar, gözcü arılar ve kaşif arılar olmak üzere üç ayrı grubu bulunmaktadır. İşçi arılar, çevredeki besin kaynaklarından en iyi nektarları toplayarak koloninin bulunduğu kovana getirmekle sorumludurlar. Daha fazla nektar toplayan işçi arılar bu durumu kovan çevresinde daha fazla dans ederek belli ederler. Gözcü arılar, işçi arıların danslarını gözlemleyerek fazla miktarda nektar toplayan işçi arıların besin kaynaklarına yönelirler. Kaşif arılar ise, arı kovanı yakınında olabilecek nektar kaynaklarını rastgele olarak araştırırlar.

Standart arı koloni algoritması (ABC) üç aşamadan oluşmaktadır. Birinci aşamada; yapay arı kolonisi rasgele olarak üretilir. Kolonideki her bir besin kaynağı (x_i), ($i=1,2,\dots,ntb$); ng boyutunda bir vektörle temsil edilebilir. Burada; ntb besin kaynağı sayısı (kolonideki arı sayısı), ng ise tasarım değişkeni sayısıdır. Rasgele oluşturulan kolonideki arıların ilk yarısı işçi arı, ikinci yarısı ise gözcü arı olarak görevlendirilir.

İkinci aşamada; daha iyi bir besin kaynağı bulmak amacıyla koloninin ilk yarısını oluşturan her bir işçi arı mevcut i 'nci konumunu aşağıdaki denklemi kullanarak değiştirir:

$$x_{ij}^{new} = x_{ij}^{old} + \phi_{ij} (x_{ij}^{old} - x_{kj}) \quad (5)$$

bu denklemde $i, k \in \{1,2,\dots,ntb\}$ ve $j \in \{1,2,\dots,ng\}$ rasgele seçilen indisler olup $i \neq k$ olmalıdır. ntb kolonideki işçi arı sayısını, j tasarım değişkenini, ϕ_{ij} ise $[-1,1]$ aralığında değişen rasgele bir reel sayıdır. İşçi arının mevcut konumu değiştikten sonra yeni konumundaki besin

kaynağına ait nektar miktarı hesaplanır. Eğer yeni konum için nektar miktarı bir önceki konuma ait nektar miktarından büyükse, işçi arı yeni konuma yerleşir. Aksi halde işçi arı önceki konumuna geri döner. Tüm işçi arılar için bu işlemler tamamlandıktan sonra işçi arılar mevcut konumlarına ait bilgileri ve nektar miktarlarını arı kovana önündeki dans alanında gözcü arılarla paylaşırlar. Daha uzun süre dans alanında kalan işçi arı gözcü arılara, nektar miktarının fazla olduğunu ve bulunduğu yeni besin kaynağının daha iyi olduğu mesajını verir. Gözcü arılar işçi arıların verdikleri mesajları dikkate alarak aşağıda verilen olasılık değerine göre besin kaynaklarını seçerler:

$$p_i = \frac{obj_i}{\sum_{n=1}^{neb} obj_n} \quad (6)$$

burada obj_i i 'nci çözüme ait amaç fonksiyonudur. Bu değer i 'nci konumdaki besin kaynağına ait nektar miktarıyla orantılıdır. Her bir işçi arıya ait olasılık değerleri hesaplandıktan sonra, gözcü arılar seçtikleri işçi arıların konumunu şu şekilde değiştirirler:

$$x_{ij}^{new} = x_{ij}^{old} + \phi_{ij} (x_{ij}^{old} - x_{ij}) \quad (7)$$

bu bağıntıda $l \in \{1,2,\dots,nob\}$ ve $i \in \{1,2,\dots,neb\}$ rasgele seçilen indisler, ϕ_{ij} $[-1,1]$ aralığında değişen rasgele bir reel sayı, nob gözcü arı sayısıdır ($nob=neb$). Eğer yeni besin kaynağına ait nektar miktarı önceki besin kaynağına ait nektar miktarından fazla ise, gözcü arı yeni besin kaynağına ait konuma yerleşir. Aksi halde önceki konumuna geri döner. Bu işlemler her bir gözcü arı için tekrarlanarak ikinci aşama tamamlanır. Bu şekilde hem işçi arılar hem de gözcü arılar daha iyi besin kaynaklarını araştırarak koloninin mevcut durumunu daha iyi bir seviyeye çıkarmayı amaçlamaktadırlar.

Son aşamada ise kolonide belli sayıda denemeye rağmen daha iyi bir besin kaynağı bulamayan işçi veya gözcü arı konumunu terk ederek kâşif arıya dönüşür. Gözcü veya işçi arılara daha iyi bir besin kaynağı bulmaları için müsaade edilen deneme sayısı limit değer (lv) olarak adlandırılır. Kaşif arı tarafından x_{ij} bu besin kaynağının yerine geçecek yeni besin kaynağı şu şekilde elde edilir:

$$x_{ij} = x_{j\min} + rand(0,1)(x_{j\max} - x_{j\min}) \quad j=1,2,\dots,ng \quad (8)$$

burada $x_{j\min}$ ve $x_{j\max}$ sırasıyla j 'nci tasarım değişkeninin alabileceği en büyük ve en küçük değerlerdir. Üçüncü aşamanın tamamlanması ile yapay arı koloni algoritmasındaki bir iterasyon bitirilmiş olur. Yukarıda bir iterasyon için açıklanan işlemler maksimum iterasyon sayısı kadar tekrarlanarak optimizasyon işlemi bitirilir.

Kafes yapıların optimizasyonu ile yapay arı koloni algoritması yöntemi arasındaki benzeşim şu şekilde kurulabilir: arı kolonisindeki işçi ve gözcü arıların toplam sayısı optimizasyon süresince kullanılacak kafes yapı sayısını, besin kaynağındaki nektar miktarı kafes yapı tasarımına ait amaç fonksiyonunun değerini göstermektedir.

Bu çalışmada, yukarıda açıklanan standart ABC'nin performansını arttırmak için önerilen MABC yönteminde ilave olarak şu stratejiler kullanılmıştır:

Standart ABC’de mevcut besin kaynağından yeni besin kaynağına gidilirken rasgele olarak sadece bir tasarım değişkeni değiştirilmektedir. Buna karşın önerilen MABC’de tüm tasarım değişkenleri değiştirilmektedir.

İkinci modifikasyon; mevcut besin kaynağı ile yeni besin kaynağının kıyaslanması konusundadır. Standart ABC’de yeni besin kaynağı ile önceki besin kaynağı karşılaştırılmaktadır. MABC’de ise yeni besin kaynağı ile kolonideki en az nektar miktarına sahip besin kaynağı kıyaslanmaktadır.

Son modifikasyonunda ise eğer en iyi besin kaynağındaki arı limit değer boyunca belli bir gelişim göstermiyorsa bile bu arının kaşif olarak atanmaması hususudur. Birinci ve ikinci modifikasyon optimizasyon sürecinin hızlandırılması amacıyla uygulanan stratejiler olup üçüncü strateji ise en iyi tasarımın koruma altına alınmasına yöneliktir.

KAFES YAPILARIN MODİFİYE EDİLMİŞ YAPAY ARI KOLONİ ALGORİTMASI İLE OPTİMİZASYONU

Bu çalışmada önerilen MABC şu adımlardan oluşmaktadır:

Adım 1: Algoritmaya ait parametrelerin kolonideki arı sayısı (ntb) ve limit değeri (lv) değerlerini ata. İterasyon sayacını sıfırla $it=0$.

Adım 2: Farklı kafes yapıları içeren başlangıç kolonisini rasgele olarak üret. Kolonideki bir kafes yapı tasarımındaki her bir tasarım değişkeni en büyük ve en küçük sınır değerler arasında aşağıdaki bağıntı ile üretilir:

$$x_i^j = x_{i_{\min}}^j + rand(0,1)(x_{i_{\max}}^j - x_{i_{\min}}^j) \quad i=1,2,\dots,ng, \quad j=1,2,\dots,ntb \quad (9)$$

Kolonideki kafes yapı tasarımları açık bir şekilde aşağıdaki matris formunda gösterilebilir.

$$x_i^j = \begin{bmatrix} x_1^1 & x_2^1 & \cdot & \cdot & x_{ng}^1 \\ x_1^2 & x_2^2 & \cdot & \cdot & x_{ng}^2 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ x_1^{ntb} & x_2^{ntb} & \cdot & \cdot & x_{ng}^{ntb} \end{bmatrix} \quad i=1,2,\dots,ng, \quad j=1,2,\dots,ntb \quad (10)$$

Matristeki her bir satır kolonideki bir kafes yapı tasarımını gösterir. Tasarımlar amaç fonksiyonu değerlerine göre artan biçimde ilk sıradan ntb 'nci sıraya kadar sıralanırlar. Burada ntb kolonideki toplam arı sayısıdır (Toplam kafes yapı sayısı). Koloninin ilk yarısı işçi arılar (neb), ikinci yarısı ise gözcü arılar olarak (nob) belirlenir.

Adım 3: Kolonideki kafes yapıların analizini yaptırarak, eleman kuvvetlerini ve düğüm deplasman değerlerini tespit et. (2)-(4) denklemleri yardımıyla $W(x)$ amaç fonksiyonunu hesapla. Kafes yapıları amaç fonksiyonu değeri küçükten büyüğe olacak şekilde sırala. Bu durumda kolonideki en iyi tasarım ilk sırada ($W(x^{eni})$), en kötü tasarım ise ($W(x^{enköti})$) en son sırada olacaktır.

Adım 4: Tasarım sayacını arttır, $it=it+1$.

Adım 5: Koloninin ilk yarısından $W(x^r)$ amaç fonksiyonu değerine sahip bir kafes yapı tasarımını rasgele seç (x^r ; $r=1,2,\dots,neb$). Denklem (5)'i kullanarak bu kafes yapıyı değiştir. Yeni kafes yapıyı analiz ederek, amaç fonksiyonunu ($W(x^{yeni})$) hesapla.

Adım 6: Eğer $W(x^{yeni}) \leq W(x^{enkötü})$ ise, yeni tasarım en kötü tasarımın yerini alır. Aksi halde önceki tasarım aynen kalır. Kolonideki kafes yapılar amaç fonksiyonu değerlerine göre tekrar sıralanır.

Adım 7: 5. ve 6. adımlar koloninin ilk yarısındaki tüm kafes tasarımlar için tekrarlanır.

Adım 8: 5. ve 6. adımdan elde edilen bilgiler koloninin ikinci yarısı ile paylaşılır. p_i olasılığı ile koloninin ilk yarısından bir kafes tasarım seçilir. Denklem (6) maksimizasyon problemlerine uygundur. Bu çalışmada ise amaç kafes yapı ağırlığını minimize etmektir. Bu amaçla denklem (6) şu şekilde modifiye edilebilir:

$$p_i = \frac{1/W(x^i)}{\sum_{n=1}^{neb} 1/W(x^n)} \quad i=1,2,\dots,neb \quad (11)$$

Adım 9: Denklem (7) ile kafes yapı tasarımında değişiklik yapılarak yeni bir tasarım elde edilir. Eğer $W(x^{yeni}) \leq W(x^{enkötü})$ ise, yeni tasarım en kötü tasarımın yerine geçer. Aksi halde önceki tasarım aynen kalır. Kolonideki kafes yapılar amaç fonksiyonu değerlerine göre tekrar sıralanır.

Adım 10: 8. ve 9. adımlar koloninin ikinci yarısındaki her tasarım için tekrarlanır.

Adım 11: Kolonideki en iyi tasarım dışındaki herhangi bir tasarım lv limit değeri boyunca gelişme göstermiyorsa, bu tasarımdan vazgeçilir. Denklem (8) kullanılarak rasgele yeni bir tasarım üretilir. Sınırlayıcıları sağlayan minimum ağırlıklı kafes yapı tasarımı mevcut optimum tasarım olarak kaydedilir.

Adım 12: Önceden belirlenen kafes yapı analizi süresinde kolonideki mevcut optimum kafes yapı tasarımından daha hafif bir tasarım elde edilmemişse, optimizasyon süreci bitirilir. Mevcut optimum nihai optimum tasarım olarak atanır. Aksi halde adım 4'e gidilerek işlemlere devam edilir.

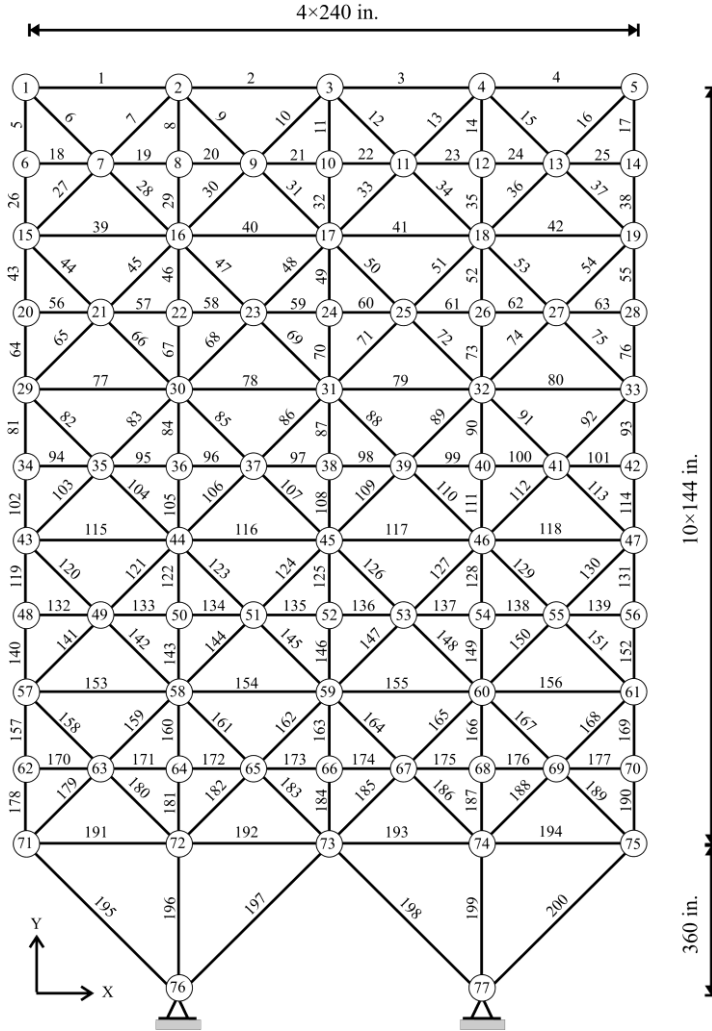
TEST PROBLEMİ

Bu çalışmada öne sürülen MABC yönteminden elde edilen sonuçları kıyaslamak için literatürde armoni arama (HS) [12], çok seviyeli tavlama benzeşimi (CMLPSA) [13], hibrid paracık küme karınca koloni optimizasyonu (HPSACO) [14] ve standart yapay arı koloni algoritması (ABC) [9] yöntemleriyle optimize edilmiş 200 elemanlı, 29 tasarım değişkenli düzlem kafes yapı kullanılmıştır.

Yöntemin kendi içindeki tutarlılığını tespit etmek amacıyla, rasgele üretilmiş farklı başlangıç tasarımları için geliştirilen program 20 sefer icra edilmiş ve 20 farklı nihai kafes yapı tasarımı elde edilmiştir. Bu tasarımlardan en ağır ve en hafif olanlarına ait değerler ile 20 farklı tasarım için hesaplanan ortalama ağırlık ve standart sapma değerleri Tablo'da sunulmuştur.

200 Elemanlı Kafes Yapı

Yapıların optimizasyonunda test problem olarak yaygın olarak kullanılan 200 elemanlı düzlem kafes Şekil 1’de verilmiştir.



Şekil 1. 200 elemanlı kafes yapı

Kafes yapı Tablo 1’de verilen 29 eleman grubundan oluşmaktadır. Malzeme için elastisite modülü ve özgül ağırlığı değerleri sırasıyla 206.85 GPa ve 7835 kg/m³’dür. Optimizasyon işleminde sınırlayıcı olarak sadece gerilme sınırlayıcısı kullanılmış olup bu değer hem basınç hem de çekme durumu için 68.95 MPa’dır. Tasarım değişkenlerinin alabileceği minimum enkesit değeri 6.4516 cm² olarak belirlenmiştir.

Kafes yapıya etki eden üç farklı yükleme durumu mevcuttur. I. Yükleme Durumunda: 1,6,15,20,29,34,43,48,57,62 ve 71 no’lu düğüm noktalarına 4.448 kN’luk yük pozitif X doğrultusunda etki etmekte, II. Yükleme Durumunda: 1, 2, 3, 4, 5, 6, 8, 10, 12, 14, 15, 16, 17, 18, 19, 20, 22, 24, 26, 28,29 30, 31, 32, 33, 34,36, 38, 40, 42, 43, 44, 45, 46, 47, 48, 50, 52, 54, 56, 57, 58, 59, 60, 61, 62, 64, 66, 68, 70, 71, 72, 73, 74 ve 75 no’lu düğüm noktalarına

44.48 kN'luk yük negatif Y doğrultusunda etki etmekte, III. Yükleme Durumunda: I. ve II. yükleme durumları beraber etki etmektedir.

MABC'de ntb ve lv değerleri 80 ve 1500 olarak atanmıştır. Bu çalışmada ileri sürülen MABC'den elde edilen ve literatürde hazır bulunan HS [12], CMLPSA [13], HPSACO [14] ve ABC'ye [9] ait sonuçlar Tablo 2'de verilmiştir.

Tablo 1. 200 elemanlı kafes yapı için tasarım tasarım değişkenleri

Tasarım değişkeni	Eleman numarası	Tasarım değişkeni	Eleman numarası
1	1,2,3,4	16	82,83,85,86,88,89,91,92,103,104,106,107,109,110,112,113
2	5,8,11,14,17	17	115,116,117,118
3	19,20,21,22,23,24	18	119,122,125,128,131
4	18,25,56,63,94,101,132,139,170,177	19	133,134,135,136,137,138
5	26,29,32,35,38	20	140,143,146,149,152
6	6,7,9,10,12,13,15,16,27,28,30,31,33,34,36,37	21	120,121,123,124,126,127,129,130,141,142,144,145,147,148,150,151
7	39,40,41,42	22	153,154,155,156
8	43,46,49,52,55	23	157,160,163,166,169
9	57,58,59,60,61,62	24	171,172,173,174,175,176
10	64,67,70,73,76	25	178,181,184,187,190
11	44,45,47,48,50,51,53,54,65,66,68,69,71,72,74,75	26	158,159,161,162,164,165,167,168,179,180,182,183,185,186,188,189
12	77,78,79,80	27	191,192,193,194
13	81,84,87,90,93	28	195,197,198,200
14	95,96,97,98,99,100	29	196,199
15	102,105,108,111,114		

Tablo 2 200 elemanlı kafes yapı için optimum tasarım sonuçları

Tasarım değişkeni (cm ²)	HS [12]	CMLPSA [13]	HPSACO [14]	ABC-AP [9]	Bu çalışma	
					MABC _{enkötü}	MABC _{eniği}
1	0.8084	0.9471	0.6664	0.6703	0.8923	1.0000
2	6.5529	6.0645	5.9251	6.1051	6.5013	6.0710
3	0.6897	0.64516	0.7755	0.6690	0.64516	0.6452
4	0.7071	0.64516	0.6509	0.7265	0.64516	0.6452
5	12.4961	12.5161	12.4127	12.5935	13.0290	12.5419
6	1.7329	1.9109	1.8232	1.8903	1.8690	1.9290
7	0.6722	0.64516	0.64516	0.6865	0.64516	0.6774
8	19.1813	20.0271	19.1503	20.1606	20.1761	20.0580
9	0.8445	0.64516	0.64516	0.6948	0.8523	0.64516
10	26.9877	26.4787	25.4554	26.6361	26.4909	26.5225
11	2.5593	2.60257	2.4142	2.7419	2.7310	2.6193
12	2.8490	1.2335	2.9039	0.6748	0.64516	1.2452
13	33.4663	35.0219	32.0018	35.3567	34.8715	35.1548
14	1.2335	0.64516	6.9277	0.6839	2.7045	0.64516
15	40.2644	41.4735	38.5709	41.8406	41.1225	41.5419
16	4.5122	3.6993	5.07283	3.6129	4.4852	3.7355
17	0.7471	0.8561	4.7576	1.1774	0.6606	0.8645
18	50.0921	51.43022	47.6186	51.8999	52.1799	51.5676
19	0.64516	0.64516	4.3058	0.6619	0.64516	0.64516
20	56.9541	57.8818	53.5483	58.2799	58.4038	57.9547
21	4.5071	4.54773	7.7207	5.0606	4.9271	4.5677
22	10.0406	2.7071	6.4516	4.8426	4.6619	2.7161
23	70.8424	70.0876	69.8463	72.9399	72.8618	70.1353
24	0.8497	0.4516	0.64516	1.4245	0.6535	0.64516
25	78.3818	76.5199	75.4682	79.1805	78.7727	76.6644
26	10.5632	6.67031	8.9548	9.0677	8.2180	6.6903
27	32.2786	43.1083	31.9503	33.2903	36.1832	43.1548
28	60.3515	69.7502	56.7741	64.4708	65.2386	69.8386
29	97.3669	89.2927	94.6095	114.2026	93.3443	89.3353
Ağırlık (kg)	11542.80	11542.14	11410.99	11582.13	11632.73	11565.80
Ortalama ağırlık (kg)	N/A	N/A	N/A	N/A	11594.07	
Standard sapma (kg)	N/A	N/A	N/A	N/A	29.01	
Sınırlayıcı ihlal yüzdesi (%)	3.69	0.071	9.97	Yok	Yok	Yok
Analiz sayısı	48000	9650	9875	1450×10 ³	30560	33280

Tablo 1’den elde edilen sonuçlar incelendiğinde HS [12], CMLPSA [13] ve HPSACO [14] ile elde edilen tasarımlar MABC’ye kıyasla daha hafif olmakla birlikte tasarım sınırlayıcısını

sağlamadıkları tespit edilmiştir. Bu durumun göz önüne alınmasıyla MABC ile elde edilen tasarımın en iyi tasarım olduğu görülmektedir. MABC; 11565.80 kg ağırlığındaki tasarımı 33280 kafes yapı analizi sonunda elde etmiş, buna karşın standart ABC [9] 11582.13 kg ağırlığındaki tasarımı 1450×10^3 analiz sonunda elde etmiştir. Bu durumda MABC, standart ABC'ye [9] kıyasla sadece daha hafif tasarım elde etmekle kalmamış aynı zamanda hesaplama süresinde de çok büyük oranda bir tasarruf sağlamıştır.

SONUÇLAR

Yapılan kıyaslamalar; öne sürülen yapay arı koloni algoritması ile diğer optimizasyon yöntemlerine ve standart yapay arı koloni algoritmasına göre daha iyi kafes yapı tasarımı elde edildiğini göstermektedir. Önemli bir sonuç olarak; modifiye edilmiş arı koloni algoritmasının optimum kafes yapıyı, standart arı koloni algoritmasından çok daha az kafes yapı analizi sonucunda elde ettiği dolayısıyla çok daha az hesaplama süresi gerektirdiği belirlenmiştir. Ayrıca, modifiye edilmiş arı koloni algoritması kullanılarak elde edilen 20 farklı kafes yapı için hesaplanan standart sapma değerinin, ortalama ağırlığa kıyasla oldukça küçük olduğu (% 0.25 mertebesinde) tespit edilmiştir. Bu durum modifiye edilmiş arı koloni algoritmasının farklı başlangıç tasarımları için bile yaklaşık olarak global optimum veya global optimuma yakın sonuçlara yakınsayabildiğinin göstergesidir. Tüm bu sonuçlar öne sürülen yöntemin kafes yapı optimizasyonunda kullanılabilir bir sezgisel optimizasyon yöntemi olduğunu kanıtlamaktadır.

KAYNAKLAR

- [1] Holland, J.H., "Adaption in Natural and Artificial Systems" The Universtiy of Michigan Press, Ann Arbor,1975.
- [2] Glover, F., "Tabu Search-Part I" ORSA Journal on Computing 1(3) 190-206, 1989.
- [3] Glover, F, "Tabu Search-Part II" ORSA Journal on Computing 2(1) 4-32, 1990.
- [4] Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P., "Optimization by simulated annealing" Science 220 671-680,1983.
- [5] Geem, Z.W., Kim, J.H., Loganathan, G.V., "A new heuristic optimization algorithm: harmony search" Simulation 76(2) 60-68, 2001.
- [6] Kennedy, J., Eberhart, R., "Particle swarm optimization" IEEE international conference on neural networks 4 1942-1948, 1995.
- [7] Karaboga, D., "An idea based on honey bee swarm for numerical optimization" Technical report-TR06, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.
- [8] Karaboga, D., Akay, B., "A modified artificial bee colony (ABC) algorithm for constrained optimization problems" Applied Soft Computing 11(3) 3021-3031, 2011.
- [9] Sonmez, M., "Artificial bee colony algorithm for optimization of truss optimization" Applied Soft Computing 11(2) 2406-2418, 2011.
- [10] Degertekin, S.O., "Optimum design of geometrically non-linear steel frames using artificial bee colony algorithm" Steel and Composite Structures 12(6) 505-522, 2012.
- [11] Kaveh, A., Talathari, S., "A particle swarm ant colony optimization for truss structures with discrete variables" Journal of Constructional Steel Research 65 1558-1568 2009.

- [12] Lee, K.S., Geem, Z.W., “A new structural optimization method based on the harmony search algorithm” *Computers and Structures* 82 781–798 2004.
- [13] Lamberti, L., “An efficient simulated annealing algorithm for design optimization of truss structures” *Computers and Structures*, 86 1936-1953 2008.
- [14] Kaveh, A., Talatahari, S., “Particle swarm optimizer, ant colony strategy and harmony search scheme hybridized for optimization of truss structures” *Computers and Structures*, 87 267-283 2009.