

En Hafif Kafes Yapı Tasarımı için Bakteri Yiyecek Arama Optimizasyon Algoritmasının Parametre Analizi

Burak KAYMAK *¹

¹Kütahya Dumlupınar Üniversitesi, Mühendislik Fakültesi, İnşaat Mühendisliği Bölümü, 43100, Kütahya, Türkiye
(ORCID: <https://orcid.org/0000-0002-1318-0456>)

(Alınış / Received: 03.04.2019, Kabul / Accepted: 03.07.2019, Online Yayınlanma / Published Online: 30.08.2019)

Anahtar Kelimeler

Bakteri yiyecek arama optimizasyonu, Parametre analizi, Optimum tasarım, Kafes yapı

Özet: Topolojisi belirli kafes yapıların en hafif tasarımının elde edilmesi problemi kesit alanlarının belirlenmesine yönelik bir optimizasyon problemidir. Optimizasyon probleminin çözümünde sürü tabanlı yöntemlerden olan bakteri yiyecek arama optimizasyon algoritması tercih edilmiştir. Bu algoritmanın en hafif kafes yapı tasarımı problemlerinin çözümünde başarı ile kullanması için seçilmesi gereken parametrelerin neler olması gerektiği üzerine çalışılmıştır. Algoritmanın parametreleri ikili gruplar halinde değiştirilerek sonuca etkileri araştırılmıştır. Ek olarak algoritmadan alınacak sonuca büyük oranda etki eden adım uzunluğu parametresinin seçiminde baştan sona sabit bir değer kullanılması yerine üreme sayılarına bağlı olarak değiştirilmesi önerilmektedir. Elde edilen bulgular sonunda en hafif kafes tasarımı problemleri için uygun parametreler belirlenmiştir. Sürü tabanlı optimizasyon yöntemleri rastgele noktalardan başladıklarından her çalıştırma sonunda elde edilen sonuçlar da farklılık göstermektedir. Elde edilecek sonuçların birbirine olan yakınlığı algoritmanın kararlılığının bir göstergesidir. Çalışma sonunda ortaya çıkan parametreler kullanılarak üç örnek problem üzerinde yapılan analiz sonuçlarının varyasyon katsayılarının %0.7'nin altında olduğu görülmüştür. Bu çalışmada elde edilen bakteri yiyecek arama optimizasyon algoritması parametrelerinin en hafif kafes yapı tasarımı problemlerinde kullanılabilir olduğunu göstermektedir.

Parameter Analysis of Bacterial Foraging Optimization Algorithm for Least Weight Design of Truss Structures

Keywords

Bacterial foraging optimization, Parameter analysis, Optimum design, Truss structure

Abstract: The problem of obtaining the least weight design of truss structures with a fixed topology is an optimization where the cross-sectional areas are determined. In order to solve the optimization problem, bacterial foraging optimization algorithm which is one of the swarm-based methods is chosen. In this work it is studied that which parameters should be chosen for successful application of the algorithm in the solution of the least weight design of truss structures. The parameters of the algorithm are collected in dual groups and their effects are investigated. Additionally, it is recommended to change the step length parameter, which greatly influences the results depending on the reproduction numbers rather than using a constant value from the beginning to end. At the end, the appropriate parameters are determined for least weight truss design problems. Since swarm-based optimization methods start at random points, the results obtained at the end of each run differs. The proximity of the results to be obtained is an indicator of the stability of the algorithm. It is seen that the coefficient of variation of the analysis results using the parameters obtained at the end of the study was below 0.7%. The bacterial foraging optimization algorithm parameters obtained in this study show that these parameters can be used for least weight truss structure design.

1. Giriş

Kafes yapılar endüstriyel yapılardan uçak-uzay sanayine kadar birçok alanda sıklıkla kullanılan yapılardır. Son elli yılda bu yapıların tasarımını yapmak üzere optimizasyon yöntemleri kullanılmaktadır. Optimizasyon yöntemlerinin

ortaya çıktığı erken dönemlerde lineer programlama, kesme düzlemi gibi matematik metodlar kullanılmıştır. Ancak kafes optimizasyon problemleri nonlineer özellik gösterdiklerinden dolayı sonuçların elde edilmesi uzun zaman alabilmektedir. Son yirmi, özellikle on yılda ise

evrimsel veya sürü tabanlı sezgisel yöntemler problemin lineer veya nonlineer olma özelliklerinden çok da fazla etkilenmediklerinden dolayı oldukça popüler hale gelmişlerdir.

Kafes yapıların sezgisel yöntemlerle sürekli değişkenli optimizasyonu ilk olarak 1964'de Dorn ve diğerleri[1] tarafından ortaya atılmıştır. Rajeev ve Krishnamoorthy 1997 yılında kafes yapıların genetik algoritma(GA) ile optimum tasarımlarını gerçekleştirmişlerdir[2]. Dorigo ve Caro[3] karınca koloni optimizasyon algoritmasını(ACO) kesikli değişkenli problemlerin çözümünü elde etmek üzere 1999 yılında ortaya koymuşlardır. Yapay karınca optimizasyon algoritmasının kafes yapılar üzerindeki uygulaması Kaveh ve Talatahari[4] tarafından 2009 yılında yapılmıştır. Bir diğer sezgisel yöntem olan armony arama(HS) algoritması Geem ve diğ.[5] tarafından 2001 yılında kafes yapıların optimum tasarımını elde etmek üzere kullanılmıştır. Tabu arama(TS) algoritması Glover[6, 7] tarafından 1990 yılında önerilmiş ve Bannage ve Dhingra[8] bu sezgisel yöntemi kafes yapıların optimum topolojisini elde etmekte kullanmışlardır. Oldukça yaygın kullanım alanı bulunan parçacık sürü optimizasyonu(PSO) algoritması Kennedy ve Eberhart[9] tarafından 1995 yılında önerilerek sezgisel yöntemler arasında yerini almıştır. PSO algoritmasının kafes yapılarda uygulamasını Schutte ve Groenwold[10] gerçekleştirmişlerdir. Karaboğa ise 2005 yılında arıların davranışlarını model alan yapay arı optimizasyonu algoritmasını ortaya koymuştur[11]. Karaboğa'nın[11] 2005 yılında önerdiği yapay arı optimizasyonu yöntemi Sönmez 2011'de kafes yapıların optimum tasarımını elde etmek üzere kullanmıştır[12, 13]. Sürü tabanlı sezgisel optimizasyon yöntemlerinden bir diğeri olan bakteri yiyecek arama optimizasyonu(BFO) Kevin Passino tarafından 2002 yılında önerilmiştir[14].

Bakteri yiyecek arama optimizasyonu algoritması birçok araştırmacı tarafından çeşitli problemlerin optimum çözümlerini elde etmek amacıyla kullanılmıştır[15–19]. Bunların bir kısmı algoritmanın performansını artırmak amacı taşıyan iyileştirmeler olmuştur[20–22]. Bakteri yiyecek arama optimizasyonu algoritması kısıtları bulunmayan problemlerin optimumlarını bulmak üzere geliştirilmiş olup daha önce çeşitli araştırmacılar[12, 23–26] tarafından Deb'in[27] kuralı uygulanarak kısıtlı optimizasyon problemlerinin çözümü gerçekleştirilmiştir. Bu çalışmada da BFO üzerinde Deb'in kuralı uygulanarak kısıtlı optimizasyon problemlerinden olan topoloji belirli en hafif kafes yapı tasarımı problemi ele alınmıştır.

Parpinelli ve çalışma arkadaşları[28] BFO algoritması kullanılarak yapısal mühendislik problemlerinin optimum çözümlerini elde etmeyi amaçlayan bir çalışma gerçekleştirmişlerdir. Bahsi geçen çalışmada bir kafes yapının en hafif tasarımını ele alan bir örnek problem yer almasına karşın BFO algoritmasının parametrelerinin seçimi konusunda nasıl karara varıldığı ile ilgili bilgi bulunmamaktadır. Oysa ki bu tür algoritmalarda kullanılan parametreler ulaşılan hedef noktanın ne oranda güvenilir ve tekrarlanabilir bir sonuç olduğuna etki etmektedir. Bu çalışmada kafes yapıların en hafif tasarımını elde etmek üzere BFO algoritması kullanılmış olup daha güvenilir, yaklaşım hızı daha az olan ve tekrarlanabilir sonuçların

elde edilmesine hizmet edecek parametrelerin seçimi için iki örnek problem üzerinde bir seri optimizasyon analizleri gerçekleştirilmiştir. Elde edilen bulgular değerlendirilerek kafes yapıların en hafif tasarımını güvenilir biçimde elde etmeyi sağlayacak parametreler belirlenmiş olup örnek kafes yapılar üzerinde elde edilen parametreler kullanılarak optimizasyon algoritması çalıştırılmış elde edilen sonuçlar değerlendirilmiştir. BFO algoritmasının iki örnek kafes yapı üzerinde kullanılacak parametrelerin belirlenmesinin ardından üçüncü bir örnek problem üzerinde de test edilmiştir. Sonuçlar göstermektedir ki bu çalışmada ortaya konulan parametreler kullanılarak gerçekleştirilen en hafif kafes yapı tasarımları için güvenilirlik, yaklaşım hızı ve tekrarlanabilirlik oldukça uygun seviyelerdedir. Ek olarak BFO algoritmasından elde edilecek sonucu önemli derecede etkileyen adım uzunluğu parametresiyle ilgili olarak çalışma kapsamında bir yaklaşım önerilmektedir. Önerilen yaklaşım ile daha az yapı analizi gerçekleştirilerek optimum sonuçların elde edilmesi mümkün olmaktadır.

Makale 6 bölümden oluşmakta olup 2. bölümde kafes yapı optimizasyonu problemi tanımlanmıştır. 3. bölümde bakteri yiyecek arama optimizasyonu algoritmasına ve algoritmanın yalancı koduna yer verilmiştir. BFO algoritmasının parametre sayısının fazla olması sebebiyle her seferde iki parametrenin sonuçlara etkisi incelenmiştir. Öncelikle adım uzunluğu ve yüzme sayılarının ardından önerilen değişken adım uzunluğu stratejisinin, takiben bakteri sayısı ve kemotaksis sayısının etkileri incelenmiştir. Ardından üreme ve ortadan kaldırma-yeniden dağılma sayılarının etkisi ve son olarak da sürü etkileşiminin etkisi ele alınmıştır. Bakteri yiyecek optimizasyonu algoritması kullanılarak gerçekleştirilen analizler neticesinde elde edilen bulgular bölüm 4'de yer almaktadır. Tartışma ve sonuç bölüm 5'de verilmektedir.

2. Kafes Tasarım Problemi

Ele alınan kafes tasarımı problemi topolojisi belirli olan kafes yapının düğüm noktalarında izin verilen deplasman limitleri ve çubuklarında oluşacak gerilmelerin kısıtlanması ile ortaya çıkan optimizasyon problemidir. Kafes yapının denge denklemleri Eş.(1)'de tarif edilmektedir. Burada B_{ij} , denge deklemleri katsayılarını, F_{jl} , l yüklemesi altındaki kafes yapının j çubuğunda oluşan kuvveti, P_{il} ise l yüklemesi altında i serbestliği doğrultusundaki düğüm yükünü gösterir. Eş.(2)'de tümeşik olarak kafes yapının bünye ve uygunluk denklemleri verilmiştir. Burada L_j , A_j ve E_j sırasıyla j çubuğunun uzunluğunu, kesit alanını, elastisite modülünü göstermektedir. x_{jl} ise l yüklemesi altında i serbestliğinde meydana gelen deplasmandır. B_{ji} uygunluk denklemi katsayılarıdır. Eş.(4), Eş.(5) ve Eş.(6) sırasıyla kafes optimizasyonu probleminin gerilme, deplasman ve kesit alanı kısıtlarıdır. σ_{jl} , l yüklemesi altındaki kafes yapının j çubuğunda oluşan gerilmeyi, σ_{jl}^a , gerilme alt limitini, σ_{jl}^u , gerilme üst limitini göstermektedir. x_{il}^a , deplasman alt limitini, x_{il}^u ise deplasman üst limitini göstermektedir. Kesit alanları alttan A_j^a ve üstten A_j^u ile sınırlandırılmıştır. Kafes yapının serbestlik derecesi sd , çubuk sayısı m ve yüklenme sayısı k ile gösterilmiştir.

$$\sum_{j=1}^m B_{ij}F_{jl} = P_{il} \quad (1)$$

$$F_{jl} = \frac{A_j E_j}{L_j} \sum_{i=1}^{sd} B_{ji} x_{il} \quad (2)$$

$$F_{jl} = A_j \sigma_{jl} \quad (3)$$

$$\sigma_{jl}^a \leq \sigma_{jl} \leq \sigma_{jl}^u \quad (4)$$

$$x_{il}^a \leq x_{il} \leq x_{il}^u \quad (5)$$

$$A_j^a \leq A_j \leq A_j^u \quad (6)$$

$$i = 1, 2, \dots, sd, \quad j = 1, 2, \dots, m, \quad l = 1, 2, \dots, k$$

Eş.(2)'de deplasmanlar cinsinden tarif edilen çubuk kuvvetleri Eş.(1)'de yerine koyulursa kafes yapı denge denklemleri deplasmanlar cinsinden tarif edilmiş olur.

$$\underbrace{\left[\sum_{j=1}^m B_{ij} \left(\frac{A_j E_j}{L_j} \right) \sum_{i=1}^{sd} B_{ji} \right]}_{\text{Kafes yapının rijitlik matrisi}} x_{il} = P_{il} \quad (7)$$

Ortaya çıkan lineer denklem takımının çözümü kafes yapının deplasmanlarını verir. Eş.(3) ve Eş.(2) düzenlenirse kafes yapının çubuklarında oluşan gerilmelerin deplasmanlar cinsinden tarifi aşağıdaki gibi elde edilebilir.

$$\sigma_{jl} = \frac{E_j}{L_j} \sum_{i=1}^{sd} B_{ji} x_{il} \quad (8)$$

Kafes optimizasyon probleminin amaç fonksiyonu Eş.(9)'daki gibi tarif edilmektedir. Burada ρ kullanılan malzemenin birim hacim ağırlığını göstermektedir. Eş.(9)'da çubukların kesit alanı(A_j) tasarım değişkeni olarak alınmıştır.

$$\text{Min } W = \sum_{j=1}^m A_j L_j \rho \quad (9)$$

Eş.(7)'nin çözümünden kafes yapının deplasmanları elde edilir. Deplasmanların Eş.(8)'de yerine koyulmasıyla da kafes yapının çubuklarında oluşacak gerilmeler elde edilmiş olur. Elde edilen deplasmanlar ve gerilmeler aşağıdaki gibi kafes optimizasyon probleminin normalize edilmiş kısıtları haline dönüştürülür.

$$\frac{\sigma_{jl}}{\sigma_{jl}^{\text{limit}}} - 1 \leq 0 \quad (10)$$

$$\frac{x_{il}}{x_{il}^{\text{limit}}} - 1 \leq 0 \quad (11)$$

Eş.(10) kafes optimizasyon probleminin normalize edilmiş gerilme kısıtı olup $\sigma_{jl}^{\text{limit}}$, l yüklemesi altındaki kafes yapının j çubuğunda izin verilen gerilme limitini tarif eder. Eş.(11) kafes optimizasyon probleminin normalize edilmiş deplasman kısıtı olup x_{il}^{limit} , l yüklemesi altındaki kafes yapının i serbestliğinde izin verilen deplasman miktarıdır. Eş.(10) ve Eş.(11) tasarım değişkenleri olan çubuk kesit alanlarına doğrusal olmayan şekilde bağlı olduğundan amaç fonksiyonu doğrusal ancak kısıtları doğrusal olmayan optimizasyon problemi olarak sınıflandırılmaktadır.

3. Bakteri Yiyecek Arama Optimizasyonu

Bakteri yiyecek arama algoritması bakterilerin yiyecek arama davranışı temel alınarak Passino[14] tarafından türetilmiştir. Bu algoritma diğer sürü tabanlı algoritmalar gibi eğimin analitik tanımına ihtiyaç duymaz. Bu nedenle doğrusal olmayan optimizasyon problemlerinde rahatlıkla kullanılabilir.

Bir bakterinin içinde bulunduğu ortam θ ile tanımlı olmak üzere $J(\theta)$ ortamın kalitesi ile ilgili bilgiyi verir. Eğer $J(\theta) < 0$ ise besin açısından zengin olan bir ortam, $J(\theta) = 0$ ise doğal ve $J(\theta) > 0$ ise zararlı ortam olduğu anlaşılır[14]. Amaç yiyecek bakımından zengin ortama ulaşmaktır. BFO algoritmasının kemotaksis, üreme ve ortadan kaldırma-yeniden dağılma olmak üzere üç temel operasyonu bulunur.

Kemotaksis operasyonu; bakterinin ortam içindeki hareketlerini tarif eder. Bu hareketler yuvarlanma ve yüzme olmak üzere iki çeşittir. Yuvarlanma hareketi Eş.(12)'de belirtildiği gibi gerçekleşir.

$$\theta^{i+1}(j+1, k, l) = \theta^i(j, k, l) + C(i) \frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}} \quad (12)$$

Burada θ^i , i . Bakterinin j . kemotaksis adımında k . üreme, ve l . ortadan kaldırma-yeniden dağılma adımındaki konumunu belirtir. θ^{i+1} ise aynı bakterinin kemotaksiste bir adım ilerlemesi durumundaki pozisyonudur. $C(i)$, i . bakterinin adım uzunluğudur. $\Delta(i)$ optimizasyon probleminin boyutu kadar elemanı bulunan ve $[-1, 1]$ değerleri arasında rastgele reel sayılardan oluşan bir vektördür.

Eş.(12) kullanılarak herhangi bir bakteri için daha iyi bir ortama geçiş yapılabilmişse bu durumda aynı doğrultuda hareket edilmeye devam edilir. Bu olay yüzme olarak adlandırılmaktadır. Yüzme hareketinin sonlanmasına ya belirlenen yüzme uzunluğunun sonuna gelmesi ya da yeni pozisyonadaki yiyecek miktarının bir öncekinden daha az olması yani daha kötü bir ortama geçilmiş olmasına göre karar verilir.

Kemotaksis operasyonları sırasında her bir bakterinin sağlık durumu aşağıdaki gibi kayıt altına alınır.

$$J_{\text{health}}^i = \sum_{j=1}^{Nc+1} J(i, j, k, l) \quad (13)$$

Eş.(13)'de elde edilen değer i . Bakterinin kemotaksis operasyonu sonundaki sağlık durumunu gösterir. Minimizasyon problemleri için düşük değer daha sağlıklı bir bakteriyi tarif eder.

Kemotaksis operasyonunun sonlanmasının ardından bakteriler en sağlıklıdan daha az sağlıklı olana doğru sıralanır. Popülasyondaki bakterilerin sağlık durumu daha kötü olan yarısı ölür. Geri kalan bakteriler ise bölünerek ürerler. Diğer bir deyişle mevcut(sağ kalan) bakterilerin bir kopyası meydana gelir. Bu adım üreme operasyonu olarak isimlendirilir[14].

Üreme operasyonları daha önceden tespit edilen belirli bir sayıya eriştiğinde P_{ed} olarak tarif edilen olasılık değerine bağlı olarak bakterilerin bir kısmı ölür ve çözüm uzayında

rastgele bir noktada yeni bir bakteri meydana gelir. Bu işlem de ortadan kaldırma-yeniden dağılıma operasyonu olarak isimlendirilir[14]. Bu çalışmada olasılık değeri klasik BFO için Passino'ya bağlı kalınarak $P_{ed} = 0.25$ olarak alınmıştır. Ortadan kaldırma-dağılıma operasyonu lokal optimuma takılma ihtimalini azaltmak amacı ile gerçekleştirilmektedir.

Bakteri yiyecek arama algoritmasında hücreden hücreye bilgi aktarımı bir diğer deyişle sürü etkileşimi Eş.(14) ve Eş.(15)'nin Eş.(13)'e eklenmesiyle mümkün olur.

$$\sum_{i=1}^s \left[-d_{attract} \exp \left(-w_{attract} \sum_{m=1}^p (\theta_m - \theta_m^i)^2 \right) \right] \quad (14)$$

$$\sum_{i=1}^s \left[h_{repellant} \exp \left(-w_{repellant} \sum_{m=1}^p (\theta_m - \theta_m^i)^2 \right) \right] \quad (15)$$

Burada s , bakteri sayısı, p , tasarım değişkenleri sayısı, θ_m^i , i . bakterinin m . tasarım değişkeni değeri, θ_m , m . tasarım değişkeni için ilgili değişkenin alt ve üst sınırları arasında rastgele bir değerdir. $d_{attract}$, $w_{attract}$, $h_{repellant}$ ve $w_{repellant}$ yöntemine ait sabitler olup Passino[14] tarafında aşağıdaki gibi verilmektedir.

$$\begin{aligned} d_{attract} &= 0.1 \\ w_{attract} &= 0.2 \\ h_{repellant} &= 0.1 \\ w_{repellant} &= 10.0 \end{aligned} \quad (16)$$

Yapı analizi ve optimizasyon algoritması C++ programlama dili kullanılarak bir bütün olarak kodlanmıştır. Bakteri yiyecek arama optimizasyon algoritması için yalancı kodlar aşağıdaki gibi tarif edilebilir.

Bakteri yiyecek arama optimizasyon algoritması için yalancı kod

```

· Çözüm uzayında rastgele noktalarda bakterileri oluştur
for l = 1 to  $N_{ed}$  do
  for k = 1 to  $N_{re}$  do
    for j = 1 to  $N_c$  do
      for i = 1 to  $S_b$  do
        m = 0
        while m <  $N_s$  do
          m=m+1
          if  $J(i,j+1,k,l) < J_{son}$  then
            ·  $J_{son} = J(i, j + 1, k, l)$ 
            · Eş.(12)'e göre yeni  $\theta^{i+1}$  hesapla
          · Eş.(13)'e göre  $J_{health}^i$  değerlerini hesapla
          ·  $J_{health}^i$  değerlerini küçükten büyüğe sırala
          · Bakterilerin yarısını öldür
            ( $J_{health}^i$  değeri büyük olanlar)
          · Sağlıklı bakterileri oldukları yerde
            ikiye bölünerek çoğalt
        ·  $P_{ed}$  değerine göre bakterilerin bir kısmını ele ve
          rastgele bir noktada yeniden oluştur

```

Burada N_{ed} ortadan kaldırma-yeniden dağılıma operasyonları sayısını, N_{re} üreme operasyonları

sayısını, N_c kemotaksis operasyonları sayısını, N_s yüzme operasyonları sayısını, S_b ise bakteri sayısını göstermektedir. $J(i, j + 1, k, l)$ $j + 1$. kemotaksis adımında, k . üreme adımında, l . ortadan kaldırma-yeniden dağılıma adımındaki i . bakterinin amaç fonksiyonu değerini gösterir. θ^{i+1} ise yukarıda tarif edilen durumdaki bakterinin çözüm uzayındaki pozisyonudur. Tasarım değişken sayısı kadar elemanı olan bir vektördür.

3.1. BFO algoritmasının parametrelerinin incelenmesinde uygulanan yöntem

Bir önceki bölümde belirtildiği üzere bakteri yiyecek arama algoritmasının kemotaksis, üreme ve ortadan kaldırma-yeniden dağılıma olmak üzere üç aşaması bulunmaktadır. Kemotaksis aşaması kendi içinde yüzme ve yuvarlanma olmak üzere ikiye ayrılır. Bu üç temel aşamada yüzme de dahil olmak üzere dört farklı parametre değerinin belirlenmesi ulaşılabilecek sonucun kalitesini etkiler. Bunların yanında her yeni pozisyon belirlemede Eş. (12) gereğince adım uzunluğunun ($C(i)$) büyüklüğü elde edilecek sonucu etkileyen parametrelerden bir diğeridir. Bir başka parametre ise ortadan kaldırma-yeniden dağılıma aşamasında rastgele olarak seçilen bakterilerin bir kısmının ölmesi ve çözüm uzayında rastgele noktalarda konumlandırılması işleminde kullanılan olasılık değeri P_{ed} 'dir. Bu çalışmada Passino'nun çalışmasına bağlı kalınarak olasılık değeri $P_{ed} = 0.25$ olarak alınmıştır. Orijinal BFO algoritmasının iterasyon, adım uzunluğu ve olasılık değeri dışında bir grup parametresi daha bulunmaktadır. Sürü etkileşimi olarak adlandırılan etkinin dikkate alınması durumunda kullanılacak parametreler Passino[14] tarafından Eş. (16)'da verildiği gibi ele alınmıştır. Bu çalışma kapsamında sürü etkileşimi parametreleri üzerine çalışılmamıştır. Ancak sürü etkileşiminin etkisinin olup olmadığı çalışma kapsamında değerlendirilmiştir.

BFO algoritmasının çok sayıda parametresi olması sebebiyle her bir parametrenin farklı değer alması durumunda ortaya çıkan kombinasyon sayısı çok fazla olacağından bu çalışmada parametreler ikili gruplar haline getirilerek kombinasyonlar oluşturulmuş ve bunun elde edilen sonuçlara etkisinin incelenmesi yöntemi tercih edilmiştir. Her aşamada etkileri incelenen iki parametre dışındaki parametreler sabit olarak tutulmuştur. Sabit tutulan parametre değerleri ilgili alt bölümlerde belirtilmektedir.

3.2. Adım uzunluğu ve yüzme sayısının incelenmesi

BFO algoritmasında optimum çözüme ulaşmada adım uzunluğu yöntemin önemli parametrelerden biridir. Adım uzunluğunun büyük seçilmesi durumunda yaklaşım hızı istenen seviyelerde olurken optimum nokta üzerinden atlanarak geçilmesi durumları ile karşılaşılabilir. Adım uzunluğunun küçük seçilmesi halinde ise optimum noktanın yakalanma ihtimali artmasına rağmen yaklaşım hızı çok yavaş kalabilir. Bazı durumlarda optimum noktaya erişebilmek için çok fazla sayıda iterasyon yapılması gerekebilir. Bu nedenle adım uzunluğunun seçimi önemlidir.

Yüzme sayısı(N_s), bir bakterinin doğrultusu belirlenmiş bir yönde en fazla kaç adım atabileceğini tanımlar. Eğer bakteri bir doğrultuda ilerlerken daha iyi bir ortama geçiyorsa(besin olarak zengin veya zararlı etkilerin bulunmadığı) belirlenen yüzme sayısına ulaşana kadar o ortamda yüzmeye devam eder(yüzme sayısı sınırına ulaşıldığında yön değiştirme işlemi gerçekleştirilir), daha kötü bir ortama geçmesi durumunda yüzme işlemi sonlandırılıp yön değiştirme işlemi gerçekleştirilir. Zengin ortamda bulunan bakterinin yüzme sayısı kısıtlanırsa optimuma doğru ilerlemesi durumunda yüzmenin hemen ardından yön değiştirme yapılacağı için muhtemel bir optimum noktanın kaçırılması gündeme gelebilir. Bu nedenle yüzme sayısının büyüklüğü önemlidir.

Yüzme sayısı ve adım uzunluğu parametrelerinin etkilerini gözleyebilmek amacıyla farklı adım uzunlukları ve farklı yüzme sayısı değerleri için BFO algoritması her iki örnek problem için çalıştırılmıştır. Burada kullanılan yüzme sayısı değerleri $N_s \in \{10, 100, 1000\}$ olarak ve adım uzunlukları da $C(i) \in \{20, 2, 0.5, 0.1, 0.05, 0.01\}$ olarak kullanılmıştır. Adım uzunluğu ve yüzme sayılarının etkilerini belirlemek amacıyla bu aşamada algoritmada kullanılan diğer parametreler şöyledir: $S_b = 40, N_c = 20, N_{re} = 20, N_{ed} = 5$. Bunlara ek olarak sürü etkileşimi ihmal edilmiştir.

3.3. Önerilen adım uzunluğu stratejisi

Adım uzunluğunun BFO algoritmasından elde edilecek optimum değer üzerindeki etkisinin son derece önemli olduğu Şekil 4 ve Şekil 5'de açıkça görülmektedir. Adım uzunluğunun problemin başında sabit olarak belirlenmesi bu haliyle bir deneme-yanılma süreci gerektirmektedir. Bir problem için uygun sonuç veren adım uzunluğu bir başka problem için çok uygun olmayan sonuçlar verebilir. Bu gibi durumların önüne geçebilmek amacıyla bu çalışmada adım uzunluğunun her üreme iterasyonu bitiminde bakterilerin adım uzunluğunun yarıya düşürülmesi yoluyla değiştirilmesi önerilmektedir.

$$C(i)_j = \frac{C(i)_{j-1}}{2} \quad j = 1, 2, \dots, N_{re} \quad (17)$$

Burada $C(i)_j$ i bakterisinin j . üreme adımındaki adım uzunluğunu belirtir. Her üreme iterasyonu sonunda bakterilerin tamamının adım uzunluğu yarıya düşürülür. Böylece başlangıçta daha büyük adımlar atan bakteriler yeni jenerasyonlarda adım sayılarını küçültmüş olacaklardır. Burada amaç optimum noktaya yaklaştıkça adım uzunluklarını küçültmek suretiyle daha hassas arama yapmaktır. Önerilen adım uzunluğu stratejisinin etkilerini incelemek üzere kafes yapı örneklerinin ikisi için de $S_b = 40, N_s = 100, N_c = 50, N_{re} = 20$ ve $N_{ed} = 5$ parametreleriyle BFO algoritması çalıştırılmıştır. Başlangıç adım uzunluğu $C(i)_0 = 20.0$ olarak alınmıştır. Sürü etkileşimi ihmal edilmiştir.

3.4. Bakteri ve kemotaksis sayılarının incelenmesi

Adım uzunluğu ve yüzme sayılarının algoritmanın yaklaşım hızına olan etkilerinin incelenmesinin ardından adım uzunluğunun değişken olarak kullanılmasının

algoritmadan elde edilen sonuçları iyileştirdiği görülmüştür(Şekil 6 ve Şekil 7). Bu nedenle çalışmanın sonraki aşamalarında önerilen adım uzunluğu stratejisi tercih edilerek algoritmada kullanılacak bakteri sayısının(S_b) ve kemotaksis sayısının(N_c) ulaşılan optimum noktaya etkisi incelenmiştir. Bunun yanında algoritmanın başlangıç noktalarını rastgele belirlemesi nedeniyle farklı zamanlarda çalıştırılması durumunda ulaşılan optimum noktaların birbirinden ne kadar dağınık, başka bir deyişle ne kadar yakın elde edilmesine etki ettikleri tespit edilmeye çalışılacaktır. Bu amaçla bakteri sayısı 20, 40, 60, 80 ve 100 olması durumunda ve kemotaksis sayısı 10'dan 100'e kadar(100 dahil) 10'ar adım artışlarla elde edilen kemotaksis sayıları kullanılarak algoritmanın 30 defa bağımsız çalıştırılması durumunda elde edilen ortalama yapı ağırlıkları ile varyasyon katsayısının (V_k) değişimi incelenmiştir. Bakteri sayısı ve kemotaksis sayısı dışındaki parametreler şöyle kullanılmıştır: $N_s = 100, N_{re} = 20, N_{ed} = 5$ ve başlangıç adım uzunluğu $C(i)_0 = 20.0$. Sürü etkileşimi ihmal edilmiştir.

3.5. Üreme ve ortadan kaldırma-yeniden dağılma sayılarının incelenmesi

Üreme sayısı(N_{re}) ve ortadan kaldırma-yeniden dağılma sayısının(N_{ed}) algoritmadan elde edilecek sonuca ne oranda etki ettiğini belirlemek amacıyla farklı üreme sayıları($N_{re} \in \{5, 10, 20\}$) ve farklı ortadan kaldırma-yeniden dağılma sayıları($N_{ed} \in \{2, 5, 10\}$) kullanılmıştır. Üreme ve ortadan kaldırma-yeniden dağılma sayılarının etkileri incelenirken bakteri adım uzunluğu parametresi -bu çalışmada önerilen- her üreme adımında adım uzunluğunun yarıya düşürülmesi stratejisi kullanılarak değişken olarak alınmıştır. Bunun için başlangıç adım uzunluğu $C(i)_0 = 20.0$ olarak seçilmiştir. Üreme sayısı ve ortadan kaldırma-yeniden dağılma sayılarının etkileri incelenirken Şekil 8, Şekil 9, Şekil 10 ve Şekil 11 dikkate alınarak bakteri sayısının $S_b = 40$, kemotaksis sayısı $N_c = 50$ ve yüzme sayısı $N_s = 100$ ve başlangıç adım uzunluğu $C(i)_0 = 20.0$ olacak şekilde algoritma çalıştırılmıştır. Sürü etkileşimi ihmal edilmiştir.

3.6. Sürü etkileşiminin incelenmesi

Son olarak sürü etkileşiminin etkilerini incelemek üzere sürü etkileşiminin dahil edilmesi ve ihmal edilmesi durumları için bakteri sayısı $S_b = 40$, kemotaksis sayısı $N_c = 50$, yüzme sayısı $N_s = 100$, üreme sayısı $N_{re} = 20$ ve ortadan kaldırma-yeniden dağılma sayısı $N_{ed} = 5$, başlangıç adım uzunluğu $C(i)_0 = 20.0$ olarak seçilmek suretiyle algoritma çalıştırılmıştır. Bu çalışmalar sonunda elde edilen bulgular bir sonraki bölümde verilmektedir.

4. Bulgular

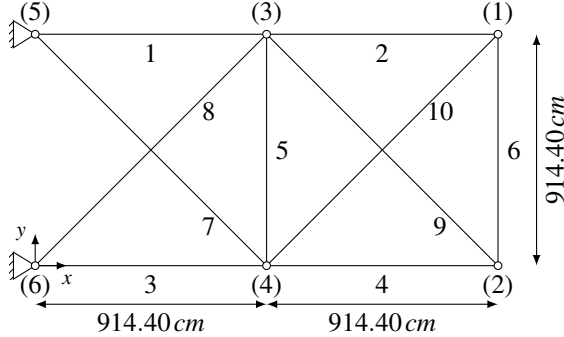
Bakteri yiyecek arama optimizasyon algoritmasının parametre sayısının fazla olması nedeniyle her bir parametre için farklı değerlerin kombinasyonunun tamamı bir seferde değerlendirmesinin zorluğundan dolayı ikili etkilerin incelenmesi yolu tercih edilmiştir. Önceki bölümde parametrelerin hangi aralıklarda kullanıldığı ile

İlgili bilgiler paylaşılmıştır. Bu parametreler kullanılarak elde edilen bulgular alt bölümlerde verilmektedir.

Çalışmada yapı ağırlığının iterasyon sayısına göre değişiminin sunulduğu grafikler bulunmaktadır. Bu grafiklerde yer alan iterasyon sayısı kemotaksis sayısı, üreme sayısı ve ortadan kaldırma-yeniden dağılıma sayıları çarpımı sonucu elde edilen sayı (iterasyon sayısı = $N_c \times N_{re} \times N_{ed}$) olarak tarif edilmiştir. Yüzme sayısı iterasyon sayısının hesabı dışında tutulmuştur çünkü bazı durumlarda bakteri için belirlenen yönde besin olarak zayıf bir ortama geçilmesi halinde yüzme sayısı sıfır olurken bazı durumlarda bakterinin her adımında besin yönünden zengin ortama geçilmesi sebebiyle verilen yüzme sayısının sonuna kadar gidilebilir. Bu nedenle yüzme sayıları her döngüde farklılık gösterebilir.

4.1. Kafes yapı örnekleri

Bu çalışmada üç farklı kafes yapı ele alınmıştır. Birinci kafes yapı örneği birçok araştırmacı tarafından [12, 29–33] en hafif kafes yapı tasarımı optimizasyonu çalışmalarında test problemi olarak ele alınan on çubuklu düzlem kafes yapısıdır. Kafes yapının geometrisi Şekil 1’de verilmektedir. Düzlem kafes yapının çubukları $\pm 172.37 MPa$ gerilme ile düğüm noktası deplasmanları ise $\pm 5.08 cm$ ile sınırlandırılmıştır. Elastisite modülü $E = 68947.57 MPa$ ve malzeme birim hacim ağırlığı $\rho = 2767.99 kg/m^3$ ’tür. Kafes yapı üzerinde 2 ve 4 numaralı düğümlerde düşey yönde $667.23 kN$, 1 ve 3 numaralı düğümlerde ise yine düşey yönde $222.41 kN$ kuvvet uygulanmıştır.



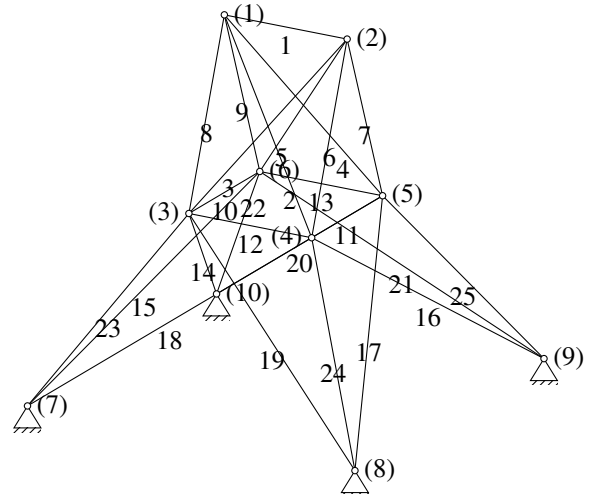
Şekil 1. 10 çubuklu düzlem kafes yapı

İkinci örnek problem de bir çok araştırmacı [12, 29–34] tarafından optimizasyon problemlerinde test örneği olarak kullanılmıştır. Yirmi beş çubuklu uzay kafes yapının geometrisi Şekil 2’de verilmektedir. Kafes yapıda kullanılan malzemenin elastisite modülü $E = 68947.57 MPa$ ve birim hacim ağırlığı $\rho = 2767.99 kg/m^3$ ’tür. Çubuk elemanlarda kullanılan gerilme limitleri ve çubukların bulunduğu grup bilgileri Tablo 1’de verilmektedir. Deplasman limitleri tüm düğümler için $\mp 0.89 cm$ ’dir. Kesit alanları alt limiti $0.06 cm^2$ olarak verilmiştir. Kafes yapı iki yükleme durumunda zorlanmakta olup etki eden kuvvetler Tablo 2’de verilmektedir.

Kafes yapı optimizasyonunda kullanılan bir başka test problemi olan 72 çubuklu uzay kafes yapının geometrisi Şekil 3’de verildiği gibidir. Kafes yapının çubukları üzerinde $\pm 172.37 MPa$ gerilme limiti ve

Tablo 1. 25 çubuklu uzay kafes yapı çubuklarının bağlı olduğu grup bilgileri ve gerilme limitleri

Grup No.	Grupta bulunan çubuklar	Basınç gerilme limiti (MPa)	Çekme gerilme limiti (MPa)
1	A ₁	241.95	275.79
2	A ₂ ~ A ₅	79.91	275.79
3	A ₆ ~ A ₉	119.31	275.79
4	A ₁₀ ~ A ₁₁	241.95	275.79
5	A ₁₂ ~ A ₁₃	241.95	275.79
6	A ₁₄ ~ A ₁₇	46.60	275.79
7	A ₁₈ ~ A ₂₁	46.60	275.79
8	A ₂₂ ~ A ₂₅	76.41	275.79



Şekil 2. 25 çubuklu uzay kafes yapı

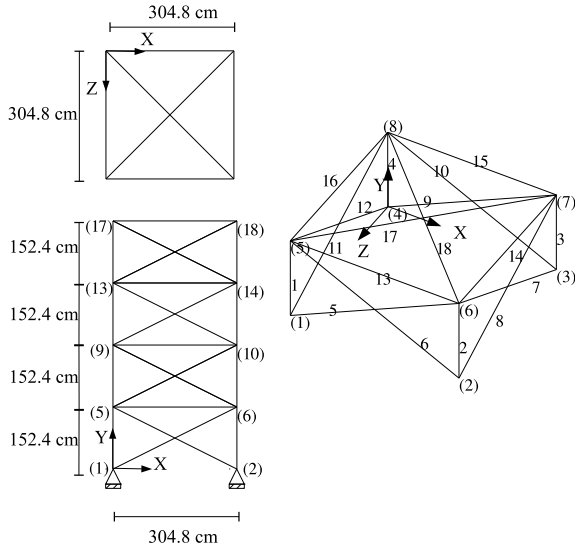
Tablo 2. 25 çubuklu uzay kafes yapıya etki eden yükler

	Düğüm No.	P_x (kN)	P_y (kN)	P_z (kN)
Yükleme durumu I	1	0.00	88.96	-22.24
	2	0.00	-88.96	-22.24
Yükleme durumu II	1	0.00	44.48	-22.24
	2	0.00	44.48	-22.24
	3	2.22	0.00	0.00
	6	2.22	-88.96	0.00

düğüm noktalarında deplasman limiti $\pm 0.64 cm$ olarak uygulanmıştır. Kullanılan malzemenin elastisite modülü $E = 68947.57 MPa$ ve birim hacim ağırlığı $\rho = 2767.99 kg/m^3$ ’tür. Kafes yapı üzerinde iki yükleme durumu etki etmektedir. Birinci yükleme durumunda 17, 18, 19 ve 20 numaralı düğüm noktalarında $22.24 kN$ kuvvet düşey yönde etki etmektedir. İkinci yükleme durumunda ise 17 numaralı düğüm noktasında pozitif x ve y yönlerinde $22.24 kN$ ve negatif z yönünde $22.24 kN$ büyüklüğünde kuvvet etki etmektedir. Kafes yapının elemanları 16 grupta toplanmıştır. Çubukların bağlı olduğu grup bilgileri Tablo 3’de verilmiştir.

Tablo 3. 72 çubuklu uzay kafes yapının çubukların bağlı olduğu grup bilgileri

Grup No.	Grupta bulunan çubuklar	Grup No.	Grupta bulunan çubuklar
1	$A_1 \sim A_4$	9	$A_{37} \sim A_{40}$
2	$A_5 \sim A_{12}$	10	$A_{41} \sim A_{48}$
3	$A_{13} \sim A_{16}$	11	$A_{49} \sim A_{52}$
4	$A_{17} \sim A_{18}$	12	$A_{53} \sim A_{54}$
5	$A_{19} \sim A_{22}$	13	$A_{55} \sim A_{58}$
6	$A_{23} \sim A_{30}$	14	$A_{59} \sim A_{66}$
7	$A_{31} \sim A_{34}$	15	$A_{67} \sim A_{70}$
8	$A_{35} \sim A_{36}$	16	$A_{71} \sim A_{72}$

**Şekil 3.** 72 çubuklu uzay kafes yapı

4.2. Adım uzunluğunun ve yüzme sayısının etkileri

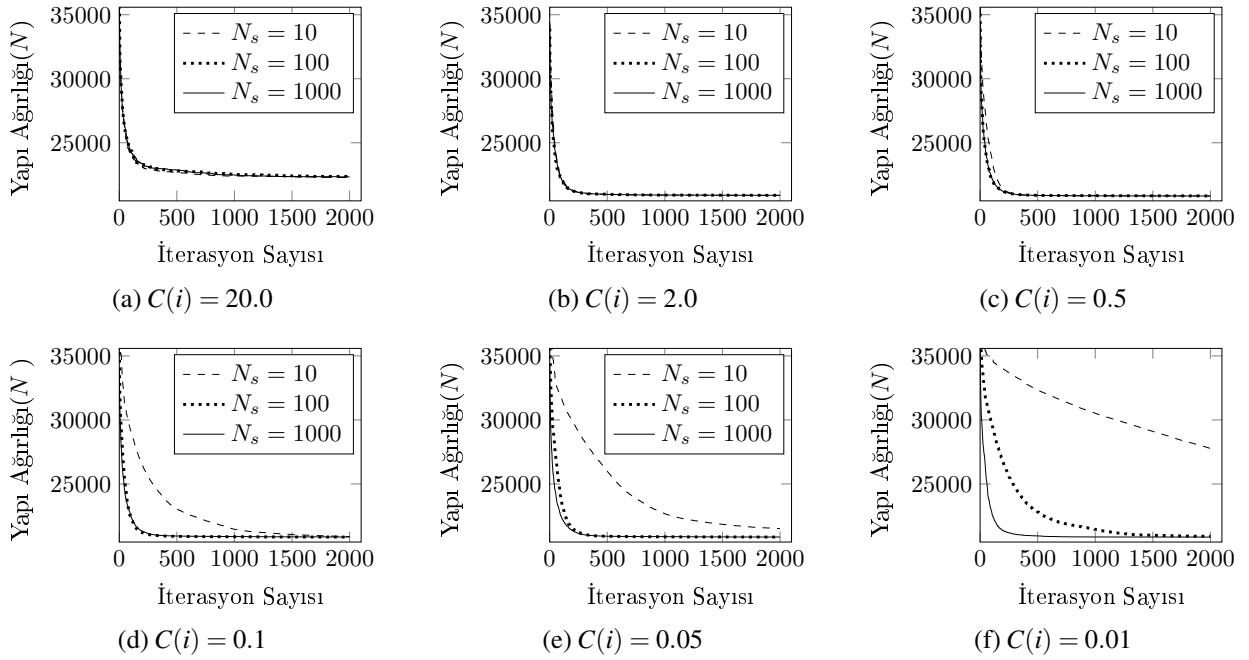
Yüzme sayısı ve adım uzunluğu parametrelerinin etkilerini gözleyebilmek için farklı adım uzunlukları için yüzme sayısı 10, 100 ve 1000 seçilmesi durumunda iterasyon sayısına karşılık yapı ağırlığının değişimi on çubuklu düzlem kafes problemi için Şekil 4'da yirmi beş çubuklu uzay kafes problemi için ise Şekil 5'de verilmiştir. Burada uygulanan adım uzunlukları büyükten küçüğe şöyle sıralanmıştır: 20, 2, 0.5, 0.1, 0.05, 0.01.

On çubuklu düzlem kafes yapı problemi için Şekil 4 incelendiğinde adım uzunluğunun 20.0 ve 2.0 olması durumunda yüzme sayısının etkili olmadığı, $C(i) = 0.5$ için yüzme sayısı $N_s = 10$ olması durumunda çok az bir farkın olduğu gözlenmektedir. Adım uzunluğunun $C(i) = 0.1$ olması halinde $N_s = 10$ için yaklaşım hızının oldukça değiştiği gözlenmektedir. Adım uzunluğunun küçülmesiyle birlikte yüzme sayılarının önemli hale geldiği anlaşılmaktadır. Bu durum adım uzunluğu 0.01 olduğu durumda Şekil 4.f'de açıkça görülmektedir (f durumu için çizgi bilgilendirmesi diğer beş durum ile aynıdır). Yüzme sayısının 10 olması halinde yaklaşım hızının oldukça düştüğü 100 ve 1000 olması durumunda adım uzunluğu 0.01 dışında çok büyük değişimlerin

olmadığı Şekil 4'den anlaşılmaktadır. Yüzme sayısı 100 ve 1000 için adım uzunluğunun büyük olması durumunda elde edilen yapı ağırlıkları görece olarak daha ağır kalmaktadır. Dikkat çeken diğer bir durum adım uzunluğunun büyük olması durumunda elde edilen yapı ağırlıklarının daha fazla olduğudur. Adım uzunluğunun küçülmesiyle birlikte daha hafif yapı ağırlıkları elde edilebilmektedir.

Yirmi beş çubuklu uzay kafes problemi için adım uzunluğu ve yüzme sayılarının farklı değerler alması durumunda iterasyon sayısına bağlı olarak yapı ağırlığının değişimleri Şekil 5'de verilmektedir. On çubuklu düzlem kafes yapı problemine benzer olarak adım uzunluğunun büyük olması durumunda algoritmadan elde edilen yapı ağırlıkları fazla olmaktadır. Özellikle adım uzunluğu $C(i) = 20.0$ için elde edilen sonuçların optimum olarak değerlendirilmesi kabul edilebilir sınırların dışında kalmaktadır. Bu örnek problemde adım uzunluğunun $C(i) = 0.01$ değeri dışında yüzme sayılarının elde edilen sonuçlara etki etmediği görülmektedir. Adım uzunluğunu 0.01 olması durumunda yüzme sayısı 10 olarak kullanılmak suretiyle algoritma çalıştırıldığında yaklaşım hızının oldukça yavaş ve elde edilen yapı ağırlığının da fazla olduğu gözlenmektedir.

Tablo 4'de on çubuklu düzlem kafes yapı problemi ve Tablo 5'de yirmi beş çubuklu uzay kafes yapı problemi için adım uzunluğu $C = 20, 2, 1, 0.5, 0.1, 0.05, 0.01$, $S_b = 40$, $N_s = 10, 100, 1000$, $N_c = 20$, $N_{re} = 20$ ve $N_{ed} = 5$ parametre değerleri kullanılarak 30 farklı analiz sonunda elde edilen en iyi, en kötü ve ortalama yapı ağırlığı değerleri ve bu değerleri elde etmek için gerçekleştirilen yapı analizi sayıları verilmiştir. Bunlara ek olarak aynı tablolarda 30 farklı analizin standart sapması ve varyasyon katsayısı % olarak verilmiştir. Bu tablolar incelendiğinde adım uzunluğunun büyük olduğu durumlarda elde edilen yapı ağırlıklarının görece fazla olduğu ancak yapı analizi sayılarının daha az olduğu görülmektedir. Tablo 4'de ortalama en hafif yapı ağırlığı 20849.0N olarak adım uzunluğu 0.5, yüzme sayısı 10 olduğu durumda elde edilmiştir. $C(i) = 0.5$, $N_s = 10$ için gerçekleştirilen yapı analizi sayılarının da nispeten az olduğu görülmektedir. Ancak aynı tablo incelendiğinde farklı adım uzunlukları için yüzme sayısı 10 olması durumunda standart sapma ve varyasyon katsayısı değerlerinin bazılarının oldukça büyük olduğu göze çarpmaktadır. Tablonun geneline bakıldığında farklı adım uzunluklarında varyasyon katsayısı değeri değişmekle beraber yüzme sayısı 10 olması durumunda en büyük varyasyon katsayısı değerleri ile karşılaşılmaktadır. Bu durumun her iki örnek için de geçerli olduğu görülmektedir. Varyasyon katsayısının büyüklüğü algoritmanın her bağımsız çalışması sonunda elde edilen değerlerin birbirinden uzaklığının ölçütü olduğuna göre yüzme sayısı 10 için algoritmanın diğer yüzme sayısı değerlerine nazaran daha az kararlı davrandığı sonucu ortaya çıkmaktadır. Bunun yanında adım uzunlukları 20.0, 2.0, 0.5, 0.1, 0.05 ve 0.01 değerlerinin herbirinde varyasyon katsayısı açısından oldukça farklı değerlere rastlanmaktadır. Bu bulgu adım uzunluğunun elde edilecek sonuca etki ettiğini açıkça göstermektedir.



Şekil 4. On çubuklu düzlem kafes probleminde farklı N_s ve $C(i)$ değerleri için BFO algoritmasının yaklaşım hızı

Tablo 4. 10 çubuklu düzlem kafes yapı optimum analiz istatistik değerleri

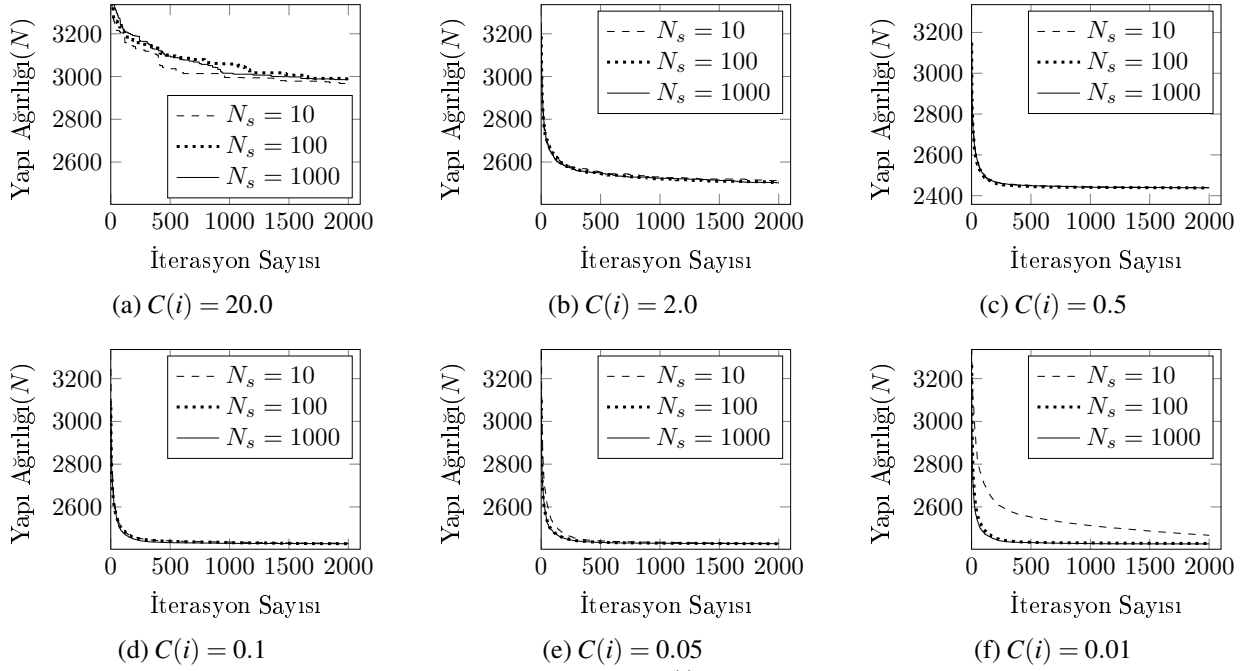
$C(i)$	N_s	Yapı Ağırlığı(N)			Yapı Analizi Sayısı			Std Sapma	V_k (%)
		En iyi	Ortalama	En kötü	En iyi	Ortalama	En kötü		
20.0	10	21567.8	22287.0	22772.8	27478	28631	30763	59.75	1.19
	100	21854.1	22364.0	23072.7	28456	28668	28859	68.45	1.36
	1000	21834.7	22337.2	22808.1	26575	28734	27960	54.60	1.09
2.0	10	20844.1	20889.0	20967.7	32424	32071	31302	6.99	0.15
	100	20823.4	20894.5	20976.1	32986	32497	32908	7.14	0.15
	1000	20836.0	20894.0	21022.7	32281	32431	33016	7.22	0.15
0.5	10	20813.4	20849.0	20919.1	39428	38874	38701	5.92	0.13
	100	20811.6	20849.2	20920.0	44829	45168	46673	5.37	0.11
	1000	20813.1	20842.5	20928.0	44597	45297	44283	4.56	0.10
0.1	10	20806.5	20891.6	21076.5	62751	60750	56319	16.71	0.36
	100	20811.0	20851.2	20943.8	90425	95393	95490	7.34	0.16
	1000	20811.1	20854.7	20971.5	114793	115022	117951	7.49	0.16
0.05	10	20811.8	21511.4	23865.4	70742	86139	88340	218.56	4.52
	100	20806.6	20851.5	20904.8	133292	136386	141294	5.86	0.12
	1000	20806.5	20854.3	20932.7	194559	199919	206766	7.09	0.15
0.01	10	23140.9	27783.7	33227.7	198044	186190	162397	537.29	8.60
	100	20811.1	20914.1	21396.2	389440	348896	403056	27.17	0.58
	1000	20810.0	20849.7	21017.4	653866	699463	703577	8.92	0.19
değişken	10	20831.4	20946.4	21233.3	43759	43270	39618	21.93	0.47
	100	20812.7	20904.2	21056.3	58378	53460	44733	14.06	0.30
	1000	20808.3	20865.8	20981.3	47322	79695	49389	8.74	0.19

4.3. Adım uzunluğu stratejisinin etkileri

Sabit adım uzunluğu kullanılması ve değişken adım uzunluğu kullanılması durumunda yapı ağırlığının iterasyon sayısı ile değişimi on çubuklu düzlem kafes yapı problemi için Şekil 6'de, yirmi beş çubuklu uzay kafes yapı problemi için ise Şekil 7'de verilmiştir.

On çubuklu düzlem kafes yapı örneğinde adım uzunluğu $C(i) \in \{20.0, 2.0, 0.5, 0.1, 0.05, 0.01\}$ değerlerinde sabit tutulmak üzere 6 farklı adım uzunluğu için BFO algoritması çalıştırılmıştır. Her adım uzunluğu için 30 bağımsız çalışma gerçekleştirilmiş olup Şekil 6'da elde

edilen 30 değerlerin ortalamasının değişimleri yer almaktadır. Adım uzunluğu $C(i) =$ değişken olarak tarif edilen durumda ise başlangıç adım uzunluğu $C(i)_0 = 20.0$ olan ancak her üreme iterasyonunda adım uzunluğunun yarıya düşürülmesi ile elde edilen yapı ağırlığı değişimleri yer almaktadır. 5000 iterasyon sonunda elde edilen ortalama en hafif yapı adım uzunluğu 0.01 olması durumunda $20871.0N$ ve ortalama yapı analizi sayısı 519125 olarak elde edilmiştir. Adım uzunluğunun değişken olması halinde ise elde edilen ortalama yapı ağırlığı $20843.6N$ ve yapı analizi sayısı 79215 olmuştur. Adım uzunluğunun değişken olması durumu ile 0.01 olarak sabit tutulması



Şekil 5. Yirmi beş çubuklu uzaykafes probleminde farklı N_s ve $C(i)$ değerleri için BFO algoritmasının yaklaşım hızı

Tablo 5. 25 çubuklu uzay kafes yapı optimum analiz istatistik değerleri

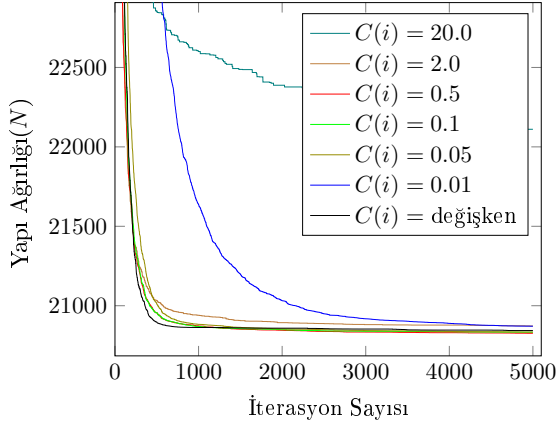
$C(i)$	N_s	Yapı Ağırlığı(N)			Yapı Analizi Sayısı			Std Sapma	V_k (%)
		En iyi	Ortalama	En kötü	En iyi	Ortalama	En kötü		
20.0	10	2621.9	2967.5	3212.0	44250	54995	59791	29.97	4.49
	100	2805.0	2989.5	3161.4	55561	56152	59918	20.27	3.02
	1000	2764.7	2985.6	3159.7	52035	55988	59424	21.58	3.22
2.0	10	2440.7	2513.2	2602.8	38591	40822	40021	7.79	1.38
	100	2475.7	2505.1	2561.4	40072	40684	40337	4.24	0.75
	1000	2451.5	2502.2	2558.5	40434	40728	42216	5.40	0.96
0.5	10	2431.5	2440.0	2459.6	39520	39495	39869	1.18	0.22
	100	2428.8	2437.9	2446.1	39555	39548	39388	0.83	0.15
	1000	2433.7	2438.7	2446.1	39181	39624	39765	0.78	0.14
0.1	10	2425.8	2429.4	2467.1	42981	43441	43811	1.81	0.33
	100	2425.9	2428.0	2443.6	45259	45988	46738	0.71	0.13
	1000	2425.9	2427.9	2440.2	46405	46312	45928	0.57	0.10
0.05	10	2425.5	2429.3	2493.5	47420	46444	46260	2.74	0.50
	100	2425.7	2427.8	2448.0	52336	53794	54755	1.18	0.22
	1000	2425.5	2428.0	2455.9	57010	55397	58128	1.41	0.26
0.01	10	2425.3	2466.9	2596.9	63503	60599	61810	8.15	1.47
	100	2425.1	2428.9	2462.4	98683	95134	92040	1.92	0.35
	1000	2425.2	2426.9	2450.5	109848	119289	121009	1.01	0.19
değişken	10	2425.2	2429.8	2485.7	42308	43388	44655	2.56	0.47
	100	2425.0	2426.7	2431.9	44561	44978	43558	0.41	0.07
	1000	2425.0	2427.4	2444.9	47229	47350	51418	0.82	0.15

halinde iki çözüm arasında ortalama yapı ağırlığı farkının %0.13 kadar olmaktadır. Adım uzunluğu değişken olduğu durumda yapı ağırlığı daha hafif olmaktadır. Bunun yanında gerekli ortalama yapı analizi sayısı sabit adım uzunluğu kullanılması durumunda değişken olması durumuna göre 6.5 katı kadar olmaktadır.

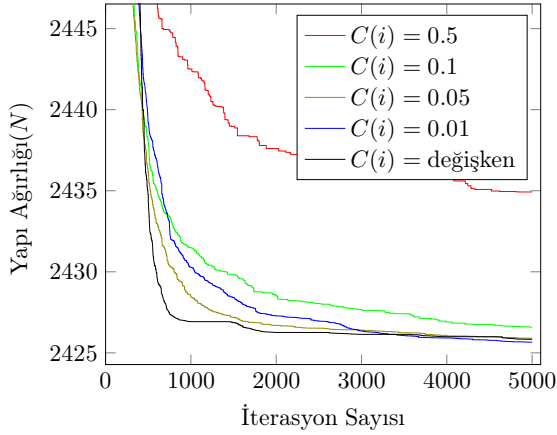
Yirmi beş çubuklu uzay kafes örneğinde de adım uzunlukları on çubuklu düzlem kafes yapı örneğinde olduğu gibi kullanılmıştır. Ancak adım uzunluğunun 20.0 ve 2.0 olması durumunda elde edilen ortalama yapı ağırlıkları 2446.5N'nin üzerinde olduğundan Şekil 7'de

bu büyüklükler kullanılarak elde edilen sonuçlara yer verilmemiştir. Yirmi beş çubuklu uzay kafes yapı örneğinde adım uzunluğunun 0.01 olarak sabit tutulması durumunda ortalama yapı ağırlığı 2425.7N ve gerçekleştirilen yapı analizi sayısı 174090 kadar olmuştur. Adım uzunluğunun değişken tutulması durumunda elde edilen yapı ağırlığı 2425.8N ve gerçekleştirilen yapı analizi sayısı da 101182 olmuştur. İki çözüm arasındaki ortalama yapı ağırlığı farkının %0.007 kadar olduğu görülmektedir. Adım uzunluğu 0.01 için ortalama yapı ağırlığı çok az da olsa daha hafif olmasına rağmen ortalama

yapı analizi sayısı yaklaşık 1.7 katı kadar olmaktadır.



Şekil 6. On çubuklu düzlem kafes yapının farklı $C(i)$ değerleri için BFO algoritmasından elde edilen ortalama yapı ağırlığı değişimleri

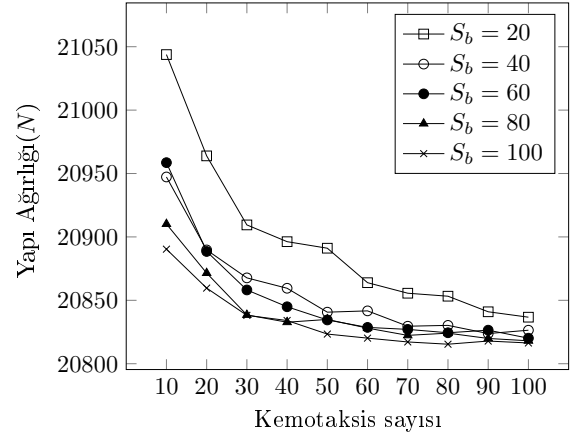


Şekil 7. Yirmi beş çubuklu uzay kafes yapının farklı $C(i)$ değerleri için BFO algoritmasından elde edilen ortalama yapı ağırlığı değişimleri

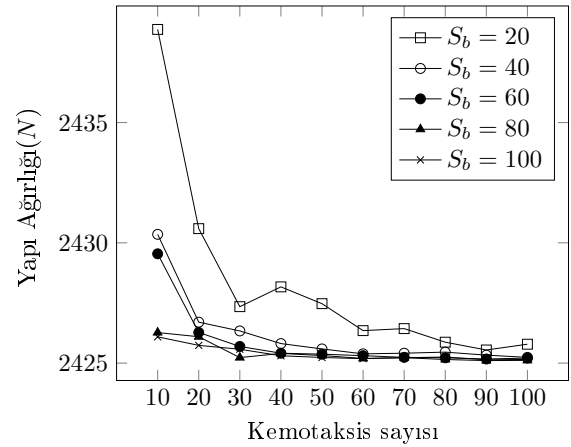
4.4. Bakteri sayısı ve kemotaksis sayısının etkisi

Şekil 8’de on çubuklu düzlem kafes yapının ve Şekil 9’da yirmi beş çubuklu uzay kafes yapının, farklı bakteri sayısı ve farklı kemotaksis sayıları kullanılması durumunda elde edilen ortalama yapı ağırlığı değişimleri verilmektedir. Bakteri sayısının 20 olması durumunda belirgin biçimde her iki örnek problemde de yapı ağırlığının daha ağır elde edildiği görülmektedir. Bakteri sayısının 40, 60, 80 ve 100 olması durumlarında yapı ağırlıklarının birbirine yakın olduğu anlaşılmakla birlikte kemotaksis sayısının artmasıyla beraber elde edilen yapı ağırlıklarının da birbirine daha fazla yaklaştığı gözlenmektedir. On çubuklu düzlem kafes yapı probleminde kemotaksis sayısının 50 ve daha büyük olması halinde gerçekleştirilen optimizasyon sonucu elde edilen yapı ağırlıkları pek de farklı değildir. Yirmi beş çubuklu uzay kafes yapı probleminde ise kemotaksis sayısı 40 ve daha büyük olması durumunda benzer durum ile karşılaşılmaktadır.

Şekil 10 ve Şekil 11’de sırasıyla on çubuklu düzlem kafes yapının ve yirmi beş çubuklu uzay kafes yapının farklı bakteri sayısı ve kemotaksis sayısı değerleri kullanılarak



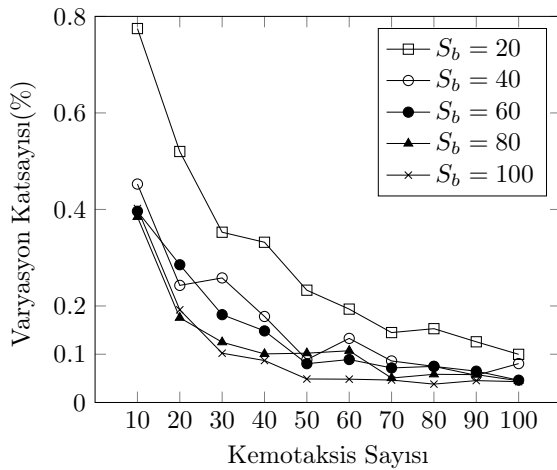
Şekil 8. On çubuklu düzlem kafes yapının farklı S_b ve N_c değerleri için BFO algoritmasından elde edilen ortalama yapı ağırlığı değişimleri



Şekil 9. Yirmi beş çubuklu uzay kafes yapının farklı S_b ve N_c değerleri için BFO algoritmasından elde edilen ortalama yapı ağırlığı değişimleri

elde edilen varyasyon katsayısı değişimleri verilmektedir. Her iki örnek problem için de elde edilen sonuçların varyasyon katsayılarının tamamının %0.8’in altında olduğu gözlenmektedir. Bu durum algoritmanın oldukça kararlı sonuçlar ürettiğinin bir göstergesidir. Bakteri sayısının 20 olması durumunda her iki örnek problemde yapı ağırlığı karşılaştırmasında olduğu gibi varyasyon katsayısı karşılaştırmasında da diğer koloni sayısı değerlerinden açıkça daha ağır ve saçılımın daha fazla olduğu değerler elde edilmektedir. On çubuklu düzlem kafes yapı probleminde kemotaksis sayısı 50 ve daha fazla olması durumunda bakteri sayısı 40, 60, 80 ve 100 için varyasyon katsayısının çok farklı olmadığı bahsi geçen koloni sayılarında %0.1 veya daha altında olduğu gözlenmektedir. Yirmi beş çubuklu uzay kafes probleminde ise kemotaksis sayısı 40 ve daha fazla olması durumunda bakteri sayısı 20 dışındaki durumlarda %0.05 civarında olduğu görülmektedir. Varyasyon katsayısının bu derece küçük olması algoritmada kullanılan parametrelerin oldukça kararlı sonuçlar üretilmesini sağladığını göstermektedir. Yapılan karşılaştırmalar sonunda bakteri sayısının 20 olması durumunda varyasyon katsayısı değerleri tatmin edici olsa da elde edilen yapı ağırlığı değerlerinin memnuniyet verici olmadığı söylenebilir. Ancak

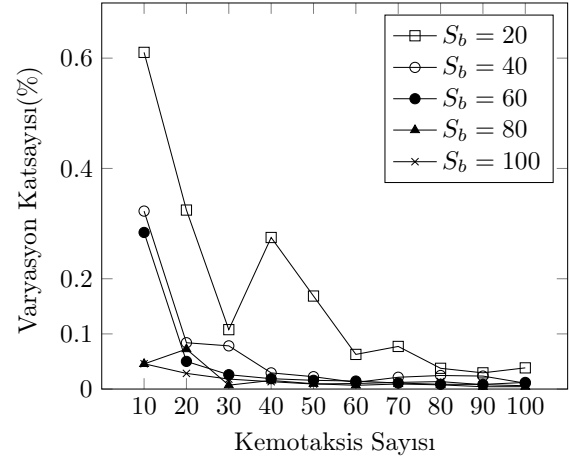
mühendislik açısından bakıldığında on çubuklu düzlem kafes yapı örneğinde bakteri sayısı 20 için elde edilen yapı ağırlıkları diğer koloni sayıları ile karşılaştırıldığında oldukça az sayıda yapı analizi gerçekleştirilerek elde edilmiş sonuçlardır. Kemotaksis sayısı 50 için elde edilen değerler baz alınarak bakteri sayısı 20 ve diğer koloni sayıları kullanılarak elde edilen sonuçlar karşılaştırılırsa yaklaşık olarak %0.34 kadar daha ağır olduğu anlaşılmaktadır. Aynı şartlarda yirmi beş çubuklu uzay kafes yapı probleminde bu fark %0.11 civarındadır. Yapı ağırlıkları arasındaki bu fark ele alınan problemin hassasiyetine göre değişiklik gösterebilir (daha hızlı sonuç alınmak istenirse) bazı durumlarda önemli görülmeyebilir. Daha hassas sonuçların elde edilmesi istenmesi halinde ise bakteri sayısının 40 civarında seçilmesi yeterli olabilecektir. Koloni sayısının 40'ın üstünde tercih edilmesi durumunda her iki örnek problemde hassasiyette bir artışın elde edilebileceğini göstermekle birlikte gerekli yapı analizi sayılarında da büyük artışlar meydana gelecektir. Ancak elde edilecek sonuçların daha az bakteri kullanılarak (40 civarında) elde edilen sonuçlardan çok da farklı olmayacağı görülmektedir.



Şekil 10. On çubuklu düzlem kafes yapının farklı S_b ve N_c değerleri için BFO algoritmasından elde edilen varyasyon katsayı değişimleri

4.5. Üreme sayısı ve ortadan kaldırma-yeniden dağılma sayısının etkisi

Farklı üreme sayıları ($N_{re} \in \{5, 10, 20\}$) ve farklı ortadan kaldırma-yeniden dağılma sayıları ($N_{ed} \in \{2, 5, 10\}$) kullanılarak on çubuklu düzlem kafes yapı ve yirmi beş çubuklu uzay kafes yapının BFO algoritmasından elde edilen sonuçları sırasıyla Tablo 6 ve Tablo 7'de verilmiştir. Bahsi geçen tablolarda ilk kolon üreme sayılarını, ikinci kolon ortadan kaldırma-yeniden dağılma sayılarını göstermektedir. Algoritma 30 defa bağımsız olarak çalıştırılmış olup bu 30 farklı çalıştırma sonunda elde edilen yapı ağırlıklarının ortalaması üçüncü kolonda, ortalama yapı analizi sayısı ise dördüncü kolonda verilmektedir. Otuz farklı çalıştırmanın standart sapması ve varyasyon katsayısı da beşinci ve altıncı kolonlarda verilmiştir.



Şekil 11. Yirmi beş çubuklu uzay kafes yapının farklı S_b ve N_c değerleri için BFO algoritmasından elde edilen varyasyon katsayı değişimleri

Tablo 6. 10 çubuklu düzlem kafes yapı için farklı N_{re} ve N_{ed} değerleri için elde edilen istatistik değerleri

N_{re}	N_{ed}	Ortalama Yapı Ağırlığı(N)	Ortalama Analiz Sayısı	Std Sapma	V_k (%)
5	2	21052.8	9101	21.77	0.46
10	2	20859.5	16908	10.17	0.22
20	2	20854.9	37865	6.89	0.15
5	5	20938.7	19698	13.65	0.29
10	5	20858.2	37471	8.33	0.18
20	5	20843.4	79444	5.69	0.12
5	10	20890.6	37137	5.20	0.11
10	10	20846.8	71590	7.66	0.16
20	10	20835.4	148122	4.08	0.09

Tablo 7. 25 çubuklu uzay kafes yapı için farklı N_{re} ve N_{ed} değerleri için elde edilen istatistik değerleri

N_{re}	N_{ed}	Ortalama Yapı Ağırlığı(N)	Ortalama Analiz Sayısı	Std Sapma	V_k (%)
5	2	2533.7	13257	6.85	1.20
10	2	2429.5	22796	1.10	0.20
20	2	2426.1	44575	0.57	0.10
5	5	2509.6	29008	4.32	0.77
10	5	2427.8	51741	0.77	0.14
20	5	2425.7	101569	0.20	0.04
5	10	2486.7	54883	4.76	0.85
10	10	2426.5	99573	0.21	0.04
20	10	2425.5	195265	0.10	0.02

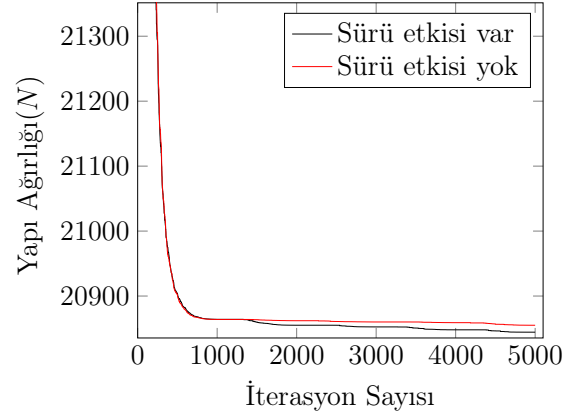
Üreme sayısı değerleri 5, 10 ve 20 için elde edilen sonuçlar incelendiğinde ortalama yapı ağırlığı açısından en ağır yapıların üreme sayısı 5 olması durumunda ortaya çıktığı görülmektedir. Yapı analizi sayıları karşılaştırıldığında en az analiz sayılarının da üreme sayısı 5 olduğu durumda elde edildiği görülmektedir. Ancak üreme sayısının 5 olarak kullanılması halinde varyasyon katsayılarının 10 ve 20 olması durumlarına göre daha fazla olduğu görülmektedir. Üreme sayısının 10 olarak kullanılması durumunda yapı ağırlığı ve varyasyon katsayılarında belirgin bir azalma meydana gelmekle birlikte yapı

analizi sayılarının da yaklaşık olarak iki katına çıktığı gözlenmektedir. Üreme sayısı 20 için elde edilen en hafif ortalama yapı ağırlıklarının ve en düşük değerli varyasyon katsayılarının elde edildiği görülmektedir. Bunun yanında en fazla ortalama yapı analizi sayıları üreme sayısı 20 olması durumunda ortaya çıktığı görülmektedir. Üreme sayısının arttıkça elde edilecek yapıların daha hafif olması ve gerekli yapı analizi sayılarının da fazla olması beklenen bir durumdur. Üreme sayısı 5 ile 20 olması durumunda gerekli yapı analizi sayıları yaklaşık olarak dört kat artarken yapı ağırlıklarındaki azalma on çubuklu düzlem kafes yapı probleminde $N_{re} = 2$ için %0.9, $N_{re} = 5$ için %0.5, $N_{re} = 10$ için %0.3, yirmi beş çubuklu uzay kafes yapı probleminde ise $N_{re} = 2$ için %4.2, $N_{re} = 5$ için %3.3, $N_{re} = 10$ için %2.5 olmaktadır. Her iki örnek problem için de üreme sayısı 20 ve ortadan kaldırma-yeniden dağılma sayısı 10 olması durumunda en hafif yapıların elde edildiği görülmektedir. Bunun yanında bahsi geçen değerler kullanılması durumunda varyasyon katsayısının da en az olduğu görülmektedir. Ancak en fazla yapı analizi sayıları da bu parametrelerle ortaya çıkmaktadır. Ancak üreme sayısı 20 ve ortadan kaldırma-yeniden dağılma sayısının 5 olarak kullanılması durumunda on çubuklu düzlem kafes yapı probleminde yapı analizi sayısı yaklaşık olarak yarıya düşerken ortalama yapı ağırlığındaki artış %0.038 kadar olmaktadır. Yirmi beş çubuklu uzay kafes probleminde ise aynı parametreler için ortalama yapı analizi sayısı benzer şekilde yaklaşık olarak yarıya düşerken ortalama yapı ağırlığındaki artış %0.008 kadar olmaktadır. Her iki problem için de yapı ağırlığındaki artış miktarı mühendislik açısından önemsenmeyebileyecek düzeyde olurken yapı analizi sayılarında yarıya yakın azalmalar meydana gelmektedir. Bu sebeple çok hassas çözüm elde edilmesi gerekmeyen durumlarda üreme sayısı 10 veya 20 seçilmesinin yeterli olacağı değerlendirilmektedir. Ortadan kaldırma-yeniden dağılma sayısının hızlı çözüm istenmesi halinde 2 olarak seçilmesi yeterli olabilecektir. Ancak hem hassasiyetin artması hem de nispeten hızlı çözüm elde edilmesi istenmesi halinde $N_{re} = 5$ olarak kullanılması uygun olacaktır.

4.6. Sürü etkileşiminin etkisi

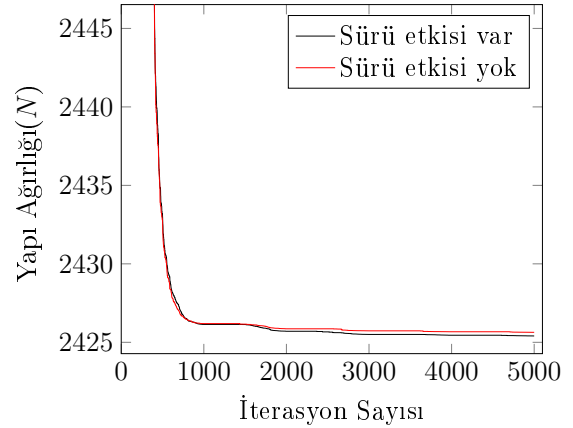
Sürü etkileşiminin ihmal edilmesi ve dahil edilmesi durumunda iterasyon sayısına bağlı olarak yapı ağırlığı değişimleri on çubuklu düzlem kafes yapı için Şekil 12'de ve yirmi beş çubuklu uzay kafes için Şekil 13'de verilmiştir. Bahsi geçen grafikler BFO parametreleri $S_b = 40$, $N_c = 50$, $N_s = 100$, $N_{re} = 20$, $N_{ed} = 5$ ve adım uzunluğu değişken, başlangıcı $C(i)_0 = 20.0$ olması durumunda elde edilmiştir. BFO algoritması diğer etkilerin incelenmesinde olduğu gibi sürü etkileşiminin etkisi incelenirken de 30 bağımsız rastgele çalıştırma gerçekleştirilerek uygulanmıştır.

On çubuklu düzlem kafes yapı için sürü etkileşimi ihmal edildiğinde 30 analizden elde edilen ortalama yapı ağırlığı 20855.0N olmuştur. Aynı problem için sürü etkileşimi dahil edildiğinde ortalama yapı ağırlığı 20844.4N olarak elde edilmiştir. İki çözüm arasındaki fark %0.051 olmaktadır. On çubuklu düzlem kafes yapı probleminde sürü etkileşimi ihmal edilmesi durumunda ortalama 79127 yapı analizi gerçekleştirilirken, sürü



Şekil 12. 10 çubuklu kafes yapı için sürü etkileşimi etkisi

etkileşiminin dahil edilmesi halinde ortalama 80554 yapı analizi gerçekleştirilmiştir. Yapı analizleri arasında fark olmadığı söylenebilir.



Şekil 13. 25 çubuklu kafes yapı için sürü etkileşimi etkisi

Yirmi beş çubuklu uzay kafes probleminde ortalama yapı ağırlıkları sürü etkileşimi ihmal edilmesi ve dahil edilmesi durumlarında sırasıyla 2425.7N ve 2425.4N olarak elde edilmiştir. İki çözüm arasındaki fark %0.011 olmaktadır. Bunun yanında sürü etkileşimi ihmal edilmesi durumunda ortalama 100343 yapı analizi gerçekleştirilirken, sürü etkileşiminin dahil edilmesi halinde ortalama 100913 yapı analizi gerçekleştirilmiştir. İki durum arasındaki ortalama yapı analizi farkı yok denilebilecek seviyededir.

4.7. Kafes yapı örnek sonuçları

Bakteri sayısı $S_b = 40$, yüzme sayısı $N_s = 100$, kemotaksis sayısı $N_c = 50$, üreme sayısı $N_{re} = 20$, ortadan kaldırma-yeniden dağılma sayısı $N_{ed} = 5$ ve adım uzunluğu değişken olarak kullanılması durumunda on çubuklu düzlem kafes için BFO algoritmasından elde edilen sonuçlar Tablo 8'de verilmiştir. Aynı tabloda farklı çözüm yöntemleri ile aynı problemi ele alan çeşitli araştırmacıların elde ettikleri sonuçlar da yer almaktadır. Yirmi beş çubuklu düzlem kafes yapı problemi için de aynı parametreler kullanılarak elde edilen sonuçlar Tablo 9'da yer almaktadır. Yetmiş iki çubuklu uzay kafes yapının optimum çözümünden elde edilen değerler Tablo 10'da yer almaktadır. BFO

Tablo 8. On çubuklu düzlem kafes yapı sonuçları

Grup No.	Kesit Alanları (cm^2)					
	Lee ve Geem [29]	Li ve diğ. [30]	Farshi ve Alinia-Ziazi [31]	Sönmez [12]	Aslani ve diğ. [32]	Bu çalışma
1	150.000	150.664	151.787	151.413	150.703	151.322
2	0.658	0.645	0.645	0.652	0.645	0.645
3	166.000	164.529	163.187	162.832	162.038	159.813
4	93.613	91.935	92.748	92.606	92.580	92.903
5	0.645	0.645	0.645	0.645	0.645	0.645
6	12.755	12.723	12.710	12.710	12.710	12.710
7	78.774	79.761	80.026	80.084	80.116	80.826
8	81.355	83.187	82.742	83.180	82.716	84.103
9	131.355	131.329	131.161	131.187	131.658	131.567
10	0.645	0.652	0.645	0.645	0.645	0.645
W(N)	20767.9	20805.6	20807.9	20804.7	20804.3	20806.9
Aşılan kısıt	3.561×10^{-3}	25.000×10^{-6}	-	-	-	-

Tablo 9. Yirmi beş çubuklu uzay kafes yapı sonuçları

Grup No.	Kesit Alanları (cm^2)			
	Li ve diğ. [30]	Sönmez [12]	Farshi ve Alinia-Ziazi [31]	Bu çalışma
1	0.065	0.071	0.065	0.065
2	12.710	12.768	12.890	12.884
3	19.458	19.374	19.245	19.213
4	0.065	0.065	0.065	0.065
5	0.065	0.065	0.065	0.065
6	4.477	4.452	4.413	4.413
7	10.845	10.832	10.806	10.806
8	17.052	17.110	17.206	17.206
W(N)	2425.1	2425.1	2425.9	2425.0

algoritmasında her üç problem için de başlangıç adım uzunluğu $C(i)_0 = 20.0$ olarak alınmıştır. Sürü etkileşimi ihmal edilmiştir.

Tavsiye edilen parametrelerin kullanılması ile üç örnek problem için elde edilen en hafif kafes yapı ağırlıkları Tablo 8, Tablo 9 ve Tablo 10'da verilmektedir. On çubuklu düzlem kafes yapı örneği için elde edilen en hafif yapı ağırlığının referans alınan beş değerden dördünden çok az da olsa daha ağır olduğu görülmektedir. Ancak literatürde verilen sonuçlardan ikisinin optimizasyon probleminin kısıtlarını bir miktar aştığı göz ardı edilmemelidir. Bu çalışmada elde edilen sonuçların tamamında üç örnek problem için de aşılacak kısıt bulunmamaktadır. Bu açıdan bakıldığında elde edilen sonucun literatür ile uyumlu olduğu değerlendirilmektedir. Yirmi beş çubuklu uzay kafes probleminde ise bu çalışmada elde edilen en hafif yapı ağırlığı referans alınan çalışmalara göre çok az daha hafif olarak elde edilmiştir. Bu örnek problemin sonuçlarının da literatürdeki sonuçlarla uyumlu olduğu görülmektedir. Benzer şekilde yetmiş iki çubuklu uzay kafes yapı örneğinde de literatür sonuçları ile yakın değerlerin elde edildiği görülmektedir.

On çubuklu düzlem kafes yapı ve yirmi beş çubuklu uzay kafes yapı örneklerinin 30 farklı çözümü sonunda elde edilen varyasyon katsayısı değerleri gerçekleştirilen

analizler ile beraber önceki bölümlerde paylaşılmıştı. Yetmiş iki çubuklu uzay kafes yapının 30 bağımsız analizi sonunda ortaya çıkan en hafif yapı ağırlığı 1693.7N olmaktadır. Bunun yanında ortalama yapı ağırlıkları 1705.7N ve elde edilen en kötü yapı ağırlığı değeri ise 1738.9N'dur. Bu örnek için standart sapma 2.61 ve varyasyon katsayısı ise %0.68 olmaktadır. Analizlerden bağımsız olarak çözülen çubuk sayısı diğer iki örneğe göre nispeten fazla olan bu örnek problemde elde edilen bulguların diğer iki örnek problemde elde edilenlerle uyumlu olduğu görülmektedir.

5. Tartışma ve Sonuç

Bakteri yiyecek arama algoritmasının kafes yapıların en hafif tasarımı problemlerinde kullanılması durumunda yüzme sayısının 10 olarak kullanılmasının -bir miktar adım uzunluğu parametresine bağlı olmakla beraber-yetersiz kaldığı görülmüştür. Yüzme sayısının 100 veya 1000 olarak seçilmesi durumunda elde edilecek sonuçlar arasındaki farkın önemsiz seviyelerde kaldığı ancak gerekli yapı analizi sayılarının oldukça fazla olduğu görülmektedir. Bu nedenle yüzme sayısının 100 olarak kullanılması tavsiye edilmektedir. Adım uzunluğu parametresinin yüzme sayılarının farklı değerlerine etki ettiği çalışmanın bulgularından anlaşılabilir bir şekilde her farklı problem için

Tablo 10. 72 çubuklu uzay kafes yapı sonuçları

Grup No.	Kesit Alanları (cm^2)		
	Dede ve diğ. [35]	Camp [33]	Bu çalışma
1	10.98	11.99	11.70
2	3.20	3.26	3.31
3	0.65	0.65	0.65
4	0.65	0.65	0.65
5	8.31	8.05	8.75
6	3.03	3.40	3.37
7	0.65	0.65	0.65
8	0.65	0.65	0.65
9	3.26	3.36	3.23
10	3.55	3.34	3.61
11	0.70	0.65	0.65
12	0.76	0.65	0.65
13	0.99	1.01	1.03
14	3.90	3.55	3.29
15	2.85	2.53	2.74
16	3.90	3.82	3.46
W(N)	1700.8	1689.7	1693.7

adım uzunluğunun hangi değer seçilmesi gerektiği ile ilgili olarak deneme-yanılma süreciyle belirlenmesinin gerektiği anlaşılmaktadır. Bunun yerine bu çalışmada önerilen adım uzunluğu parametresinin üreme iterasyonlarına bağlı olarak değişken olarak kullanılması adım uzunluğunun hangi değer olarak kullanılması problemini ortadan kaldırdığı görülmektedir.

Algoritmada kullanılan bakteri sayısının 40 ve daha fazla olması durumunda kemotaksis sayısının da 50 ve üzeri seçilmesi halinde elde edilen sonuçların dağılımı ve değerlerinin yaklaşık olarak aynı olduğu söylenebilir. Bakteri sayısının artmasıyla birlikte elde edilen yapı ağırlıkları az miktarda iyileşirken gerçekleştirilmesi gereken yapı analizi sayıları büyük oranda artmaktadır. Bu durum kemotaksis sayıları için de geçerlidir. Bakteri sayısının 40, kemotaksis sayısının 50 olarak uygulanması makul sürelerde tatmin edici sonuçların alınmasını sağlayacaktır.

Üreme sayısı lokalde arama iterasyon sayısını, ortadan kaldırma-yeniden dağılma sayısı ise aramanın yalnızca lokalde kalmamasını sağlamak amacıyla kullanılan parametrelerdir. Bu değerlerin mümkün mertebe yüksek seçilmesi daha iyi sonuçların elde edilmesini sağlayabileceği düşünülse de bu parametrelerdeki artışın gerekli kıldığı yapı analizi sayısı oldukça fazla olmaktadır. Buna karşılık yapı ağırlığındaki azalma oranı analiz sayısı ile karşılaştırıldığında aradaki ilişkinin doğrusal olmadığı anlaşılmaktadır. Elde edilen bulgular göz önünde tutulduğunda üreme sayısının 20, ortadan kaldırma-yeniden dağılma sayısının ise 5 olarak kullanılması durumunda makul sürelerde uygun sonuçların alınabildiği, algoritmanın farklı zamanlarda çalıştırıldığında elde edilen sonuçlar arasındaki varyasyon katsayılarının düşük seviyelerde kaldığı görülmektedir.

Sürü etkileşiminin dahil edilmesi ve ihmal edilmesi durumları arasında yapı ağırlığı açısından bakıldığında

iki çözüm arasındaki farkın %0.05 ve altında olduğu görülmektedir. Bu farkın önemli olmadığı, en hafif kafes yapı tasarımı problemlerinde sürü etkileşiminin ihmal edilebileceği düşünülmektedir.

Bakteri yiyecek arama optimizasyon algoritmasının topolojisi belirli olan en hafif kafes yapı tasarımı problemlerinde başarılı olarak uygulanabileceği görülmektedir. Bunun yanında en hafif kafes yapı tasarımı problemleri için algoritmada kullanılacak parametre değerlerinin belirlenmesi ile ilgili olarak yapılan çalışmaların uygun değerlerin seçiminde yol gösterici olacağı düşünülmektedir. Ek olarak algoritmanın önemli parametrelerinden olan adım uzunluğunun tespiti ile ilgili önerilen tekniğin de daha az yapı analizi ile daha hafif yapıların elde edilmesine katkı sağladığı sonucuna varılmaktadır.

Kaynakça

- [1] Dorn, W. S., 1964. Automatic design of optimal structures, *Journal de mecanique* 3 25–52.
- [2] Rajeev, S., Krishnamoorthy, C. S, 1997. Genetic Algorithms-Based Methodologies for Design Optimization of Trusses, *Journal of Structural Engineering*, 123 (3)
- [3] Dorigo, M., Di Caro, G., 1999. Ant colony optimization: a new meta-heuristic, *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99*, 6-9 July, Washington, 1470–1477.
- [4] Kaveh, A., Talatahari, S., 2009. A particle swarm ant colony optimization for truss structures with discrete variables, *Journal of Constructional Steel Research*, 65 (8-9), 1558–1568
- [5] Geem, Z. W., Kim, J. H., Loganathan, G., 2001. A New Heuristic Optimization Algorithm: Harmony Search, *SIMULATION*, 76 (2), 60–68
- [6] Glover, F., 1990. Tabu Search - Part I, *ORSA journal on Computing*, 2 (1)
- [7] Glover, F., 1990. Tabu Search—Part II, *ORSA Journal on Computing*, 2 (1)
- [8] Bennage, W. A., Dhingra, A. K., 1995. Optimization of truss topology using tabu search, *International Journal for Numerical Methods in Engineering*, 38 (23), 4035–4052
- [9] Kennedy, J., Eberhart, R., 1995. Particle swarm optimization, *International Conference on Neural Networks*, 27 Nov.-1 Dec., Perth, 1942–1948
- [10] Schutte, J. F., Groenwold, A. A., 2003. Sizing design of truss structures using particle swarms, *Structural and Multidisciplinary Optimization*, 25 (4), 261–269
- [11] Karaboga, D., 2005. An idea based on Honey Bee Swarm for Numerical Optimization, *Technical Report TR06*, Erciyes University
- [12] Sonmez, M., 2011. Artificial Bee Colony algorithm for optimization of truss structures, *Applied Soft Computing* 11 (2), 2406–2418.

- [13] Sonmez, M., 2011. Discrete optimum design of truss structures using artificial bee colony algorithm, *Structural and Multidisciplinary Optimization* 43 (1), 85–97.
- [14] Passino, K. M., 2002. Biomimicry of bacterial foraging for distributed optimization and control, *Control Systems, IEEE*, 22 (3), 52–67
- [15] Devi, S., Geethanjali, M., 2014. Application of Modified Bacterial Foraging Optimization algorithm for optimal placement and sizing of Distributed Generation, *Expert Systems with Applications* 41 (6), 2772–2781
- [16] Niu, B., Wang, H., Wang, J., Tan, L., 2013. Multi-objective bacterial foraging optimization, *Neurocomputing*, 116, 336–345.
- [17] Sathya, P. D., Kayalvizhi, R., 2011. Modified bacterial foraging algorithm based multilevel thresholding for image segmentation, *Engineering Applications of Artificial Intelligence*, 24 (4), 595–615
- [18] Majhi. R., Panda, G., Majhi, B., Sahoo, G., 2009. Efficient prediction of stock market indices using adaptive bacterial foraging optimization (ABFO) and BFO based techniques, *Expert Systems with Applications*, 36 (6), 10097–10104
- [19] S. Hezer, Y. Kara, 2014, Eşzamanlı dağıtım ve toplamalı araç rotalama problemlerinin çözümü için bakteriyel besin arama optimizasyonu tabanlı bir algoritma, *Gazi Üniversitesi Mühendislik-Mimarlık Fakültesi Dergisi*, 28 (2), 373–382
- [20] Zhao, W., Wang, L., 2016. An effective bacterial foraging optimizer for global optimization, *Information Sciences*, 329, 719–735.
- [21] Biswas, A., Das, S., Abraham, A., Dasgupta, S., 2010. Stability analysis of the reproduction operator in bacterial foraging optimization, *Theoretical Computer Science*, 411 (21), 2127–2139
- [22] Chen, H., Niu, B., Ma, L., Su, W., Zhu, Y., 2014. Bacterial colony foraging optimization, *Neurocomputing*, 137, 268–284.
- [23] Karaboga, D., Basturk, B., 2007. Artificial Bee Colony (ABC) Optimization Algorithm for Solving Constrained Optimization, 12th International Fuzzy Systems Association World Congress, June 18-21, Mexico, 789–798
- [24] Kaveh, A., Bakhshpoori, T., 2013. Optimum Design of Space Trusses Using Cuckoo Search Algorithm With Levy Flights, *IJST, Transactions of Civil Engineering*, 37 (C1), 1–15.
- [25] Cheng, M. Y., Prayogo, D., 2014. Symbiotic Organisms Search: A new metaheuristic optimization algorithm, *Computers and Structures*, 139, 98–112
- [26] Cuevas, E., Cienfuegos, M., 2014. A new algorithm inspired in the behavior of the social-spider for constrained optimization, *Expert Systems with Applications*, 41 (2), 412–425
- [27] Deb, K., 2000. An efficient constraint handling method for genetic algorithms, *Computer Methods in Applied Mechanics and Engineering*, 186 (2-4), 311–338
- [28] Parpinelli, R. S., Teodoro, F. R., Lopes, H. S., 2012. A comparison of swarm intelligence algorithms for structural engineering optimization, *International Journal for Numerical Methods in Engineering*, 91 (6), 666–684
- [29] Lee, K. S., Geem, Z. W., 2004. A new structural optimization method based on the harmony search algorithm, *Computers & Structures*, 82 (9-10), 781–798
- [30] Li, L. J., Huang, Z. B., Liu, F., Wu, Q. H., 2007. A heuristic particle swarm optimizer for optimization of pin connected structures, *Computers and Structures*, 85 (7-8), 340–349
- [31] Farshi, B., Alinia-Ziazi, A., 2010. Sizing optimization of truss structures by method of centers and force formulation, *International Journal of Solids and Structures*, 47 (18-19), 2508–2524
- [32] Aslani, M., Ghasemi, P., Gandomi, A. H., 2018. Constrained mean-variance mapping optimization for truss optimization problems, *Structural Design of Tall and Special Buildings*, 27 (6), 1–17
- [33] Camp, C. V., 2007. Design of Space Trusses Using Big Bang–Big Crunch Optimization, *Journal of Structural Engineering*, 133 (7), 999–1008
- [34] Lamberti, L., 2008. An efficient simulated annealing algorithm for design optimization of truss structures, *Computers & Structures*, 86 (19-20), 1936–1953
- [35] Dede, T., Bekiroglu, S., Ayvaz, Y., 2011. Weight minimization of trusses with genetic algorithm, in: *Applied Soft Computing Journal*, 11, 2565–2575