



**T.C.
AKSARAY ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**İNŞAAT MÜHENDİSLİĞİ
ANABİLİM DALI**

**YAPAY ARI KOLONİ ALGORİTMASI KULLANILARAK 3 BOYUTLU
ÇELİK YAPILARIN OPTİMUM TASARIMI**

YÜKSEK LİSANS TEZİ

Özer SEVİM

DANIŞMAN

Doç. Dr. Mustafa SÖNMEZ

AKSARAY, 2013

T.C.
AKSARAY ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ
KABUL ve ONAY BELGESİ

Özer SEVİM 'in YAPAY ARI KOLONİ ALGORİTMASI KULLANILARAK 3 BOYUTLU ÇELİK YAPILARIN OPTİMUM TASARIMI başlıklı lisansüstü tez çalışması, aşağıdaki jüri tarafından İnşaat Mühendisliği Anabilim Dalında **YÜKSEK LİSANS** tezi olarak kabul edilmiştir.

İmza

Danışman : Doç. Dr. Mustafa Sönmez (Aksaray Üniversitesi)

Üye : Doç. Dr. M. Aydın Kömür (Aksaray Üniversitesi)

Üye : Yrd. Doç. Dr. Necmi Düşünceli (Aksaray Üniversitesi)

Tezin Savunulduğu Tarih: 12.07.2013

Fen Bilimleri Enstitüsü Yönetim Kurulu' nun 19.07.2013 tarih ve 2013/26-01 sayılı kararı ile onaylanmıştır.


Doç. Dr. Selçuk REİS
Enstitü Müdürü

ÖNSÖZ

Çelik yapılar günümüzde sürekli gelişmektedir ve kullanımı her geçen gün artmaktadır. Çelik yapılar betonarme yapılara göre maliyetli daha fazla olduğundan maliyetini azaltmak için ekonomik olarak tasarlanması çok önemlidir.

Son yıllardaki bilgisayar teknolojisindeki gelişmeler, buna paralel olarak geliştirilen programlama dilleri ve hazır paket programları, yapı analizi için kullanılan programların artık tasarım içinde kullanılabileceğini göstermiştir. Bu çalışmada iki ve üç boyutlu kafes ve çerçevelerin optimum tasarımı için C# programlama dili kullanılarak yapay arı koloni algoritması kodlanmıştır.

Özer SEVİM

Aksaray, Temmuz 2013

TEŞEKKÜR

Akademisyenlikte ilk adım olan yüksek lisans eğitiminin sonuna gelmiş bulunuyorum. Mesleğimin ayrıntılarını öğrenmek, ülkeme faydalı olmak için önümde aşmam gereken birçok engelin olduğunu bilerek;

Eğitimim süresince ilminden faydalandığım, insani ve ahlaki değerleri ile örnek edindiğim, beraber çalışmaktan onur duyduğum değerli danışman hocam Sayın Doç. Dr. Mustafa SÖNMEZ (Aksaray Üniversitesi)' e bölümde desteğini esirgemeyen ve bana yol gösteren başta bölüm başkanım Sayın Doç. Dr. Mehmet Emin KARA (Aksaray Üniversitesi) olmak üzere Aksaray Üniversitesi İnşaat Mühendisliği Bölümü Öğretim Üyelerine, birlikte çalışmaktan zevk aldığım iş arkadaşlarıma ve bugünlere gelmemde büyük pay sahibi olan maddi ve manevi her türlü desteği gösteren aileme teşekkürlerimi sunarım. Bu tezi anne ve babama ithaf ediyorum.

Özer SEVİM

Aksaray, Temmuz 2013

İÇİNDEKİLER DİZİNİ

	Sayfa No
ÖNSÖZ.....	i
TEŞEKKÜR.....	ii
ÖZET.....	v
ABSTRACT.....	vi
ŞEKİLLER DİZİNİ.....	vii
ÇİZELGELER DİZİNİ.....	ix
SİMGELER KISALTMALAR DİZİNİ.....	x
KISALTMALAR DİZİNİ.....	xiii
1. GİRİŞ.....	1
1.1. Temel Kavramlar.....	1
1.2. Bu Çalışmanın Amacı.....	2
1.3. Optimizasyonun Temel Kavramları.....	2
1.3.1. Optimizasyon probleminin tanımı.....	3
1.3.2. Tasarım değişkenleri.....	4
1.3.3. Tasarım kısıtları.....	4
1.3.4. Kısıt yüzeyler.....	4
1.3.5. Amaç fonksiyonu.....	4
1.3.6. Optimizasyon problemlerinin sınıflandırılması.....	5
1.4. Yapısal Optimizasyon.....	5
1.4.1. Boyut optimizasyonu.....	6
1.4.2. Şekil optimizasyonu.....	6
1.4.3. Topoloji optimizasyonu.....	6
1.5. Yapıların Optimizasyonunda Kullanılan Yöntemler.....	7
1.5.1. Gradyan temelli yöntemler.....	8
1.5.2. Direkt (Modern Sezgisel) arama metotları.....	9
1.6. Literatür Çalışmaları.....	10
2. YAPAY ARI KOLONİ (ABC) ALGORİTMASI.....	12
2.1. Sürü Zekası.....	12
2.1.1. Arıların sürü zekası.....	12
2.1.2. Yiyecek arama.....	13
2.2. Doğadaki Arıların Davranışı	13
2.3. Arı Koloni Algoritması.....	15
2.4. Sınırlı Optimizasyonların Sınırsız Optimizasyon Olarak Çözülmesi.....	17
2.5. Örnek 1.....	19
2.6. Örnek 2.....	22

3. SAYISAL ÖRNEKLER.....	
3.1. On Elemanlı Çelik Kafes Sistemi.....	27
3.2 Tek Açıklıklı On Katlı Düzlem Çelik Çerçeve.....	28
3.3. Üç Açıklıklı Yirmi Dört Katlı Düzlem Çelik Çerçeve.....	32
3.4. Tek Açıklıklı Dört Katlı Uzay Çelik Çerçeve.....	43
4. TARTIŞMA VE SONUÇLAR.....	50
KAYNAKLAR.....	62
EKLER	64
ÖZGEÇMİŞ	

ÖZET

Yüksek Lisans Tezi

YAPAY ARI KOLONİ ALGORİTMASI KULLANILARAK 3 BOYUTLU ÇELİK YAPILARIN OPTİMUM TASARIMI

Özer SEVİM

T.C.

Aksaray Üniversitesi Fen Bilimleri Enstitüsü
İnşaat Mühendisliği Anabilim Dalı

Danışman: Doç. Dr. Mustafa SÖNMEZ

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

Bu çalışmada düzlem çelik çerçeveler ve uzay çelik çerçevelerin ayrık optimizasyonu için bir Yapay Arı Koloni Algoritması (ABC) sunulmuştur. Bu çalışmanın etkinliğinin gösterilmesi için 168 elemana ve 20 tasarım değişkenine çıkabilen ve uzay çelik çerçeve içeren dört farklı yapısal örnek çözülmüştür. Sonuçlar literatürdeki bu örneklerin diğer sezgisel arama teknikleri ile elde edilen sonuçları ile kıyaslanmıştır. Sonuçların düzlemsel ve uzay çelik çerçeve yapılarının tasarımı için ABC algoritmasının çok etkili ve güçlü bir teknik olduğu gösterilmiştir.

2013, 88 Sayfa

Anahtar Kelimeler: Yapay Arı Koloni Algoritması, Ayrık Optimizasyon, Çelik Çerçeve Tasarımı, ABC, Sürü Zekâsı, C# Programlama, SAP2000

ABSTRACT

Master of Science Thesis

OPTIMUM DESIGN OF THREE-DIMENSIONAL STEEL STRUCTURES USING ARTIFICIAL BEE COLONY ALGORITHM

Ozer SEVIM

T.R.

Aksaray University Graduate School of Natural and Applied Sciences
Department of Civil Engineering

Supervisor: Assoc. Prof. Dr. Mustafa SONMEZ

Over the past few years, swarm intelligence based optimization techniques such as ant colony optimization and particle swarm optimization have received considerable attention from researchers. These algorithms have been used in the solution of various engineering problems. This study presents an Artificial Bee Colony Algorithm (ABC) technique for discrete optimization of planer steel frames and space steel frames.

In order to demonstrate the effectiveness of algorithm, four problems including space frame problem with up to 168 members and 20 design variables were solved and results were compared with those obtained using other well-known meta-heuristic search techniques. The results demonstrate that the artificial bee colony algorithm (ABC) is very effective and robust for planer steel frames and space steel frame design.

2013, 88 Pages

Keywords: Artificial Bee Colony Algorithm, Discrete Optimization, Steel Frame Design, ABC, Swarm Intelligence, C# Programming, SAP2000

ŞEKİLLER DİZİNİ

	Sayfa No
Şekil 1.1. $f(x)$ 'in minimumu – $f(x)$ 'in maksimumu	3
Şekil 1.2. Boyut optimizasyonu	6
Şekil 1.3. Şekil optimizasyonu	6
Şekil 1.4. Topoloji optimizasyonu	7
Şekil 1.5. Optimizasyon yöntemleri	8
Şekil 2.1. Arılarda görev paylaşımı	13
Şekil 2.2. Bal arılarının sallanma dansı (Waggle Dance).....	14
Şekil 2.3. $f(x, y) = x^2 + y^2$ fonksiyonunun grafiksel çözümü.....	19
Şekil 2.4. İki elemanlı kafes	23
Şekil 3.1. On elemanlı kafes sistemin geometrisi ve yükleme şekli.....	28
Şekil 3.2. On elemanlı kafes sisteminin geometrisi ve yüklemenin Sap2000 görünümü.....	29
Şekil 3.3. On elemanlı kafes sistemi için ağırlık ve döngü sayısı ilişkisi	29
Şekil 3.4. On elemanlı kafes sistemi için optimum tasarım şekli	30
Şekil 3.5. On elemanlı kafes sistemi için x ve y doğrultusundaki yer değiştirmeler	31
Şekil 3.6. On elemanlı kafes sistemi için eleman gerilmeleri	31
Şekil 3.7. Tek açıklıklı on katlı çerçeve sisteminin boyutlandırılması, yükleme şekli ve elemanların gruplandırılması	33
Şekil 3.8. Tek açıklıklı on katlı çerçeve sisteminin Sap2000 ile yüklenmesi	34
Şekil 3.9. Etkili uzunluk faktörü	37
Şekil 3.10. Tek açıklıklı on katlı çerçeve sistemi için maksimum tepe yer değiştirmesi ve maksimum katlar arası yer değiştirmenin yeri	39
Şekil 3.11. Optimum tasarım sonucu oluşan kesitler ve düzlemsel görünümleri	40
Şekil 3.12. Tek açıklıklı on katlı çerçeve sistemi için ağırlık ve döngü sayısı ilişkisi	41
Şekil 3.13. Tek açıklıklı on katlı çerçeve sistemi x doğrultusundaki yer değiştirmeler	41
Şekil 3.14. Tek açıklıklı on katlı çerçeve sistemi (a) eksenel kuvvet, (b) moment diyagramı	42

Şekil 3.15. Üç açıklıklı 24 katlı çerçeve sisteminin boyutlandırılması, yükleme şekli ve elemanların gruplandırılması	44
Şekil 3.16. Üç açıklıklı 24 katlı çerçeve sistemi için maksimum yer değiştirme	47
Şekil 3.17. Üç açıklıklı 24 katlı çerçeve sistemi için ağırlık ve döngü sayısı ilişkisi	
Şekil 3.18. Üç açıklıklı 24 katlı çerçeve için katlar arası yer değiştirmeler ..	48
Şekil 3.19. Üç açıklıklı 24 katlı çerçeve sistemi için (a) eksenel kuvvet, (b) moment diyagramı	48
Şekil 3.20. Dört katlı otuz iki elemanlı uzay çerçeve: (a) plan, (b) yandan görünüş	49
Şekil 3.21. Dört katlı otuz iki elemanlı uzay çerçeve (D) yüklemesi	51
Şekil 3.22. Dört katlı otuz iki elemanlı uzay çerçeve (L) yüklemesi	53
Şekil 3.23. Dört katlı otuz iki elemanlı uzay çerçeve (L_r) yüklemesi	54
Şekil 3.24. Dört katlı otuz iki elemanlı uzay çerçeve W yüklemesi	55
Şekil 3.25. Dört katlı otuz iki elemanlı uzay çerçeve optimum kesitlerle Sap2000 görünümü.....	56
Şekil 3.26. Dört katlı otuz iki elemanlı uzay çerçeve maksimum yer değiştirmesi	58
Şekil 3.27. Dört katlı otuz iki elemanlı uzay çerçevenin ağırlık ve döngü sayısı ilişkisi	59
Şekil 3.28. Dört katlı otuz iki elemanlı uzay çerçeve için katlar arası yer değiştirmeler	60
Şekil 3.29. Dört katlı otuz iki elemanlı uzay çerçeve için 4. yükleme (a) ksenel kuvvet, (b) moment diyagramı	60
	61

ÇİZELGELER DİZİNİ

	Sayfa No
Çizelge 2.1. Tasarım değişkenine göre I değerleri tablosu	16
Çizelge 2.2. $x^2 + y^2$ için ilk tasarım değişkenlerinin değeri ve bu tasarım değişkenleri kullanılarak amaç fonksiyonunun değeri.....	20
Çizelge 2.3. $x^2 + y^2$ için istihdam edilen en iyi ilk beş arı	20
Çizelge 2.4. $x^2 + y^2$ için istihdam edilen arılar ve varsa izleyici arılar için amaç fonksiyonu	21
Çizelge 2.5. $x^2 + y^2$ yeni bir yenileme için istihdam edilen arılar ve varsa izci arılar için amaç fonksiyonu	22
Çizelge 2.6. $x^2 + y^2$ için 50. yenileme sonrası en son amaç fonksiyonu	22
Çizelge 2.7. Kafes elemanın ilk tasarım değişkenlerinin değeri ve bu tasarım değişkenleri kullanılarak amaç fonksiyonunun değerleri	24
Çizelge 2.8. Kafes eleman için istihdam edilen en iyi ilk beş arı	24
Çizelge 2.9. Kafes eleman için istihdam edilen arılar ve varsa izleyici arılar için amaç fonksiyonu	26
Çizelge 2.10. Kafes eleman için yeni bir yenileme için istihdam edilen arılar ve varsa izci arılar için amaç fonksiyonu	26
Çizelge 2.11. Kafes eleman için 50. yenileme sonrası en son amaç fonksiyonu	26
Çizelge 3.1. On elemanlı kafes sisteminin optimum sonuçları	30
Çizelge 3.2. Tek açıklıklı on katlı çelik çerçeve sisteminin optimum sonuçları	38
Çizelge 3.3. Üç açıklıklı yirmi dört katlı çerçeve sisteminin optimum sonuçları	46
Çizelge 3.4. Bina yüksekliğine bağlı rüzgar faktörü C_e (UBC97 tablo 16-G)	52
Çizelge 3.5. Dört katlı otuz iki elemanlı uzay çerçevenin optimum sonuçları	57

SİMGELER DİZİNİ

A_g	Elemanın en kesit alanı
A_{st}	Donatı kesit alanı
b	Kiriş genişliği
b_{fbk}	Kiriş kenar genişliği
b_{fck}	Kolon kenar genişliği
C_e	Yüksekliğe bağlı rüzgar faktörü
c_j^{μ}	Dinamik Ceza katsayısı
C_m	Yanal ötelenme durumuna göre kullanılan katsayı
C_q	Rüzgar basınç katsayısı
d_{bn}	Alt kat kolon derinliği
d_{un}	Üst kat kolon derinliği
E	Elastisite modülü
F	Kuvvet
F_y	Çeliğin akma gerilmesi
$f(\vec{x})$	Amaç fonksiyonu
h	Kiriş yüksekliği
I	Atalet momenti
I_c	Kolon atalet momenti
I_g	Kiriş atalet momenti
$max I_j$	Maksimum tasarım indisi
$min I_j$	Minimum tasarım indisi
$^0 I_j$	Tasarım değişkeni yerinin indis olması durumu
I_w	Rüzgar önem katsayısı
K	Etkili uzunluk faktörü
L	Uzunluk
L_c	Kolon uzunluğu
L_g	Kiriş uzunluğu
M_{lt}	Sadece yanal kuvvetlerden meydana gelen çerçevelerdeki momentler
M_{nt}	Yanal deplasmanların önlendiği çerçevelerdeki momentler
M_{nx}	Kuvvetli eksenindeki taşıyabileceği maksimum eğilme momenti
M_{ny}	Zayıf eksenindeki taşıyabileceği maksimum eğilme momenti

M_{ux}	İkinci dereceden etkileri içeren kuvvetli eksendeki eğilme momenti
M_{uy}	İkinci dereceden etkileri içeren zayıf eksendeki eğilme momenti
n_c	Toplam kısıt sayısı
q_s	Rüzgar durgunluk katsayısı
p	Rüzgar yükü
P	Tekil Yük
p_i	Arıların besin kaynağı seçme olasılığı
P_n	Çekmede veya basınçta taşınabilecek maksimum kuvvet
P_u	Çekmede veya basınçta eksenel kuvvet
r	Atalet yarıçapı
r_i^λ	Dinamik Ceza katsayısı
t	Kesit et kalınlığı
$U_{k,l}$	Yer değiştirme kısıtı
U_1	X doğrultusundaki yer değiştirme
U_2	Y doğrultusundaki yer değiştirme
U_3	Z yönünde dönme
iX_j	i. arının j. tasarım değişkeni
V	Hacim
$\beta_{ij}(I)$	Maksimum yer değiştirme kısıtı
$\beta_{ij}(Z)$	Katlar arası yer değiştirme kısıtı
β_1	Kolondan dolayı meydana gelen moment artırma katsayısı
β_2	Katlardan dolayı meydana gelen moment artırma katsayısı
δ_{jl}	Maksimum yer değiştirme
δ_{ju}	Maksimum yer değiştirme sınırı
Δ_{jil}	Katlar arası yer değiştirme
Δ_{ju}	Katlar arası yer değiştirme sınırı
Δ_{oh}	Belirtilen yükler altında kattaki yer değiştirme
σ_{cr}	Burkulma gerilmesi
σ_y	Akma gerilmesi
$\sigma_{m,l}$	Gerilme kısıtı
$\emptyset$	Dayanım azaltma katsayısı
μ	Dinamik Ceza katsayısı
ρ	Malzeme yoğunluğu

σ	Gerilme
ϕ	-1 ile 1 arasında rastgele sayı
λ	0 ile 1 arasında rastgele sayı
ΣH	Kata maruz kalan bütün yatay yüklerin toplamı

KISALTMALAR DİZİNİ

ABC	Artificial Bee Colony (Yapay Arı Kolonisi)
ACO	Ant Colony Optimization (Karıncı Kolonisi Optimizasyonu)
AISC	Amerikan Çelik Yapılar Enstitüsü
ANN	Artificial Neural Network (Yapay Sinir Ağları Yöntemleri)
CPU	Central Processing Unit (Merkezi İşlem Birimi)
FLO	Fuzzy Logic Optimization (Bulanık Mantık Optimizasyonu)
GA	Genetic Algorithm (Genetik Algoritmalar)
HPSO	Hiyerarşi Parçacık Sürü Optimizasyon Algoritması
HS	Harmony Search (Harmoni Arama Metodu)
IACO	Geliştirilmiş Ant Colony Optimization
LRFD	Yük ve Dayanım Faktörü Tasarımı
MNC	Maksimum Döngü Sayısı
PSO	Particle Swarm Optimization (Parçacık Sürü Optimizasyonu)
SA	Simulated Annealing (Isıl İşlem Optimizasyonu)
TLBO	Öğretme-Öğrenme Tabanlı Optimizasyon Yöntemi
UBC	Uniform Building Code (Uniform Binaların Şartnamesi)

1. GİRİŞ

1.1. Temel Kavramlar

İnsan nüfusunun hızla artması, yaşanılacak yerlerin sınırlı olması insanları farklı arayışlara yönlendirmiştir. Bu arayışlar ile birlikte teknolojinin de gelişmesiyle en yüksek, en dayanıklı ve en ekonomik fonksiyonel yapıların tasarlanması mühendislerin en önemli amaçlarından birisi olmuştur.

En sağlam, en yüksek, en ekonomik fonksiyonel yapılar bilgisayar programları yardımı ve optimizasyon teknikleri kullanılarak tasarlanması mühendislerin en büyük amacıdır. Optimizasyon teknikleri klasik metotlar ve modern sezgisel teknikler olarak iki ana başlıkta incelenmektedir. Araştırmacılar çeşitli yapısal optimizasyon problemlerini klasik optimizasyon yöntemleri kullanarak çözmek için çaba göstermişlerdir. Ancak gerçekte mühendislik problemleri belli bir çözüm yöntemine uyacak şekilde modellemek genellikle pek kolay değildir. Bu nedenle klasik optimizasyon tekniklerinin bu yetersizliklerini aşabilmek için problemten ve modelden bağımsız olan, doğadan esinlenen sezgisel optimizasyon algoritmaları önerilmektedir. Bu teknikler hem etkin hem de daha esnek olduğundan her tür problem gereksinimlerine göre uyarlanabilmektedir. Bu nedenlerden dolayı, doğadan esinlenilerek sezgisel optimizasyon algoritmaları geliştirilmiştir.

İnşaat mühendisliğinde ön tasarımın amacı ilk gerilmeleri bularak ön tasarımdaki kabullerin doğruluğunu kontrol etmektir. Ön tasarımdaki ilk kabullerin tam doğru ve en ekonomik olma olasılığı yok denecek kadar azdır. Ön tasarımda seçilen kesitler küçük ise gerilme ve yer değiştirme şartları sağlanmadığı için kesitler artırılır, eğer gerilme ve yer değiştirme değerleri kesitin taşıyabileceğinden çok daha küçük ise sistemin daha ekonomik olması için kesitlerin azaltılması gereklidir. Anlaşılacağı gibi bu işlemler yinelemeli (iteratif) işlemlerdir. Bu yinelemeli işlemlerin bir yöntem (metodoloji) kullanılarak yapılmasına optimizasyon denir.

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

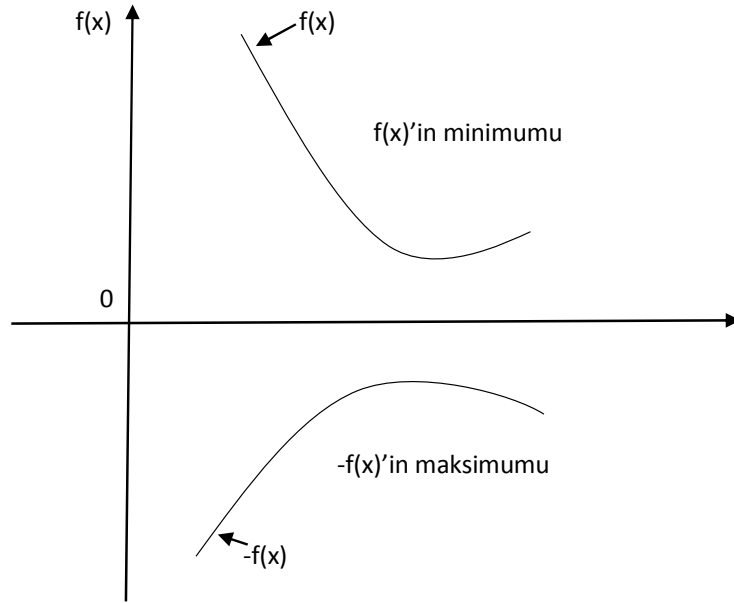
1.2. Bu Çalışmanın Amacı

Çelik taşıyıcı sistemlerin tasarımında emniyet gerilmeleri yöntemi yaygın olarak kullanılmakla beraber, son yıllarda taşıma gücü yöntemini esas alan yönetmeliklerin hazırlanması, taşıma gücü yönteminin kullanımını hızla yaygınlaştırmaktadır. Artan hammadde ve işçilik maliyetleri, hammadde bulunmasında yaşanabilen problemler ve çelik fiyatlarındaki ani yükselişler çelik taşıyıcı sistemlerin ekonomik olarak boyutlandırılması zorunluluğunu ortaya çıkarmıştır. Bu amaçla çelik yapıların optimum tasarımı konusunda farklı optimizasyon yöntemlerini kullanan önemli araştırma çalışmaları yapılmıştır.

Yapay Arı Kolonisi (Artificial Bee Colony) optimizasyon algoritması kullanılarak iki ve üç boyutlu kafes ile çerçeve sistemlerin optimum tasarımlarını yapacak bilgisayar programı geliştirilmiştir. Yapay arı koloni algoritması (ABC-Artificial Bee Colony) daha önceden uzay kafes (Sonmez 2011a; Sonmez 2011b; Sonmez vd., 2013) ve düzlem çerçevelerin (Değertekin 2012; Saka vd., 2013) optimizasyonunda kullanılmıştır. Düzlem ve uzay çelik çerçeve sistemlerin tasarımında tasarım değişkenleri olarak sadece eleman kesitleri ele alınmıştır. Bu çalışmada ticari hazır paket programlarından Sap2000 yapı analiz programı hazırlanan bilgisayar programına entegre edilmiştir. Programlama dili olarak nesne tabanlı C# programı seçilmiştir. C# programlama dilinin seçilmesindeki en önemli etken, .Netframework çevresinde hazır paket programlar ile uyumlu çalışmasıdır. Sonuç olarak, çubuk kuvvetleri ve deplasmanlar Sap2000 programı yardımı ile hesaplanmış olup optimizasyon işlemi ise C# program dili kullanılarak geliştirilmiştir.

1.3. Optimizasyonun Temel Kavramları

Optimizasyon bir problemin çözümü için elde edilen sonuçlardan en iyisini seçme işidir. En iyisini seçme işi bir karar vermedir. Bu kararlar en karlı yatırım, en hafif tasarım veya en ucuz üretim olabilir. Bu kararlar verilirken aynı zamanda bazı kısıtlarlarda göz önünde bulundurulmalıdır. Mesela en ucuz üretim yapılacak ise üretim miktarının günlük bir sınırı vardır. İnşaat mühendisi, en hafif yapıyı yapacaksa bu yapı hem az maliyetli ve hem de aynı zamanda verilen servis yükleri ve maksimum yükler altında istenilen performansı göstermelidir. Sonuç olarak optimizasyon minimum veya maksimum bulma işi olarak tanımlanabilir. Şekil 1.1.' de görüleceği gibi minimum yapma işi ile maksimum yapma işi aslında aynı şeydir.



Şekil 1.1. $f(x)$ 'in minimumu – $f(x)$ 'in maksimumu

Sonsuz sayıdaki optimizasyon problemleri için etkili ve tek bir optimizasyon yöntemi yoktur. Çok sayıda optimizasyon tekniği farklı tipteki optimizasyon problemlerini çözmek için geliştirilmiştir. Optimizasyon işine “Matematiksel programlama teknikleri” de denir ve genellikle *yöneylem* konusu altında incelenir.

1.3.1. Optimizasyon probleminin tanımı

En genel hali ile optimizasyon problemi aşağıdaki gibi tanımlanabilir;

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

$$g_j(\vec{x}) \leq 0 \quad j = 1, 2, 3, \dots, m \quad (1.1)$$

$$l_k(\vec{x}) = 0 \quad k = 1, 2, 3, \dots, p \quad (1.2)$$

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

Buradaki $\vec{x}$, n boyutlu bir vektördür ve tasarım değişkenleri vektörü (design variable vector) denir. Tasarım değişkenleri vektöründeki her bir eleman tek bir tasarım değişkenini temsil eder; $f(\vec{x})$ amaç fonksiyonu (objective function); $g_j(\vec{x})$ eşitsizlik kısıtları (inequality constraints) ve $l_k(\vec{x})$ eşitlik kısıtları (equality constraints) olarak tanımlanır. n , m ve p ise birbiri ile ilişkisi olmayan tamsayılarıdır; sırası ile tasarım değişkenlerinin, eşitsizlik kısıtlarının ve eşitlik kısıtlarının sayısını göstermek için kullanılmıştır. Denklem 1.1 ve 1.2’de verilen kısıtlar sınırlı optimizasyon probleminde vardır. Bazı problemlerde sınırlar yoktur ve bu tür problemler de sınırsız optimizasyon denir. Sınırsız optimizasyon aşağıdaki gibi gösterilebilir.

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

$$\vec{x} = \{x_1, \dots, x_i, \dots, x_n\} \quad (1.3)$$

Denklem 1.3'deki gibi problemler sınırsız optimizasyon problemi olarak adlandırılır. Burada adı geçen terimler daha ayrıntılı olarak aşağıda tanımlanmıştır.

1.3.2. Tasarım değişkenleri

Herhangi bir mühendislik yapısı tasarlanırken bazı büyüklükler tasarım aşamasında değişken olarak bazıları da sabit olarak alınır. Örnek olarak kiriş tasarlanıyorsa kiriş açıklığı, moment, beton ve çelik sınıfı sabit alınırken kirişin yüksekliği (h), genişliği (b), kullanılacak donatı alanı (A_{st}) değişken olarak alınabilir. Değişken olarak alınacak büyüklükler eğer bir vektör ile gösterilirse $\vec{x} = \{b, h, A_{st}\}$ veya $\vec{x} = \{x_1, x_2, x_3\}$ bu vektörün elemanlarına tasarım değişkenleri, vektöre ise “tasarım vektörü” denir. Kartezyen koordinat sisteminin her bir eksenini bir değişkene karşı gelecek şekilde oluşturulan uzaya tasarım değişkenleri uzayı veya tasarım uzayı denir.

Bu uzaydaki her bir noktaya “tasarım noktası” (design point) da denir. Bu nokta tasarım probleminin çözümü olabilir veya olmayabilir. Kiriş örneğinde olduğu gibi b , h , A_{st} değerlerinin hesaplar sonucunda negatif olma durumu, imkânsız bir çözümü göstermektedir.

1.3.3. Tasarım kısıtları

Optimizasyon problemlerinde tasarım değişkenleri için istenilen her değer verilmez. Kiriş problemi örneğinde olduğu gibi b değerleri negatif olamaz ve ayrıca bu değerler TS-500 de verildiği gibi her zaman 20 cm den büyük veya eşit olmalıdır. Diğer taraftan toplam donatı alanı minimum sınırın üstünde ve maksimum sınırında altında olmalıdır. Bu tür kısıtlara geometrik veya yan kısıtlar da denir.

1.3.4. Kısıt yüzeyler

Kısıt fonksiyonu $g_j(\vec{x}) \leq 0$ ve $l_k(\vec{x}) = 0$ ile sınırlanan alan kısıt yüzeyidir. Bu yüzey tasarım uzayını farklı bölgelere ayırır. Bunlara örnek olarak müsaade edilen/uygun (feasible) veya uygun olmayan / müsaade edilmeyen (unfeasible) bölgeler verilebilir. Örnek olarak tasarlanan kirişin göstereceği maksimum yer değiştirme standartlardakinden az olmalı veya kirişin taşıyacağı moment miktarı ise kirişe uygulanan momentten fazla olmalıdır. Bu tür kısıtlara bazen performans bazen de davranış kısıtları da denilmektedir.

1.3.5. Amaç fonksiyonu

Klasik tasarım işleminin amacı kabul edilebilir veya sınır şartlarına uygun (genellikle bunlar standartlarda verilmiştir) bir çözüm elde etmektir. Genellikle bu tür problemler

için birden çok çözüm vardır. Optimizasyonun amacı var olan bu çözümlerden en iyisini bulmaktır. Buradaki en iyi çözüm deyimi çözülecek probleme göre değişir. Eğer problem bir uçağın tasarlanması ise amaç en hafif uçağı imal etmek, eğer bir inşaat mühendisliği problemi çözülecek ise amaç en ekonomik yapıyı elde etmek olabilir.

Bazı problemlerde ise çözülecek problemin birden çok kriteri olabilir. Bu tür problemlere çok amaçlı optimizasyon (multi-objective optimization) problemleri denir. Bu tür problemler ise

$$f(\vec{x}) = \alpha_1 f_1(\vec{x}) + \alpha_2 f_2(\vec{x}) + \alpha_3 f_3(\vec{x}) + \dots \quad (1.4)$$

şeklinde formüle edilerek tek amaç fonksiyonlu hale dönüştürülebilir. Ancak bu yapılırken dikkat edilmesi gerekli nokta, α 'ların alabileceği değerlerin bir sistem dahilinde seçilmesidir.

1.3.6. Optimizasyon problemlerinin sınıflandırılması

Optimizasyon problemleri farklı şekillerde sınıflandırılabilir. Bu bölümde bunlardan bazıları üzerinde durulacaktır. Rao' ya göre optimizasyon problemleri aşağıdaki gibi sınırlandırılabilir (Rao, 2009).

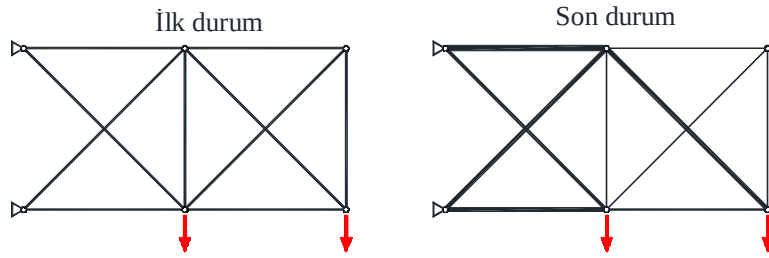
- Kısıtların var olup olmadığına göre sınırsız/sınırlı optimizasyon problemleri.
- Tasarım değişkenlerinin optimizasyon işlemleri için her noktada sabit kaldığı (statik optimizasyon) veya tasarım değişkeninin belli noktalar arası değişmesine müsaade edildiği (dinamik optimizasyon problem) durumlarına göre ayrılır. Örnek olarak sabit bir kesitli kirişteki boyut optimizasyonu statik optimizasyon olarak adlandırılırken, değişken kesitli bir kirişin boyut optimizasyonu dinamik optimizasyon olur.
- Tasarım değişkenlerinin alabileceği değerlere göre sürekli (continuous) / ayrık (discrete) değişkenli optimizasyon problemleri.
- Amaç fonksiyonunun yapısına göre doğrusal (linear), doğrusal olmayan (non-linear) ve geometrik (geometric) optimizasyon problemleri.
- Tasarım değişkenlerinin deterministik veya stokastik olarak elde edilmesine göre sınıflandırılabilir.
- Amaç fonksiyonun sayısı ile ilgi olarak tek amaç fonksiyonlu veya çok amaç fonksiyonlu optimizasyon problemleri.

1.4. Yapısal Optimizasyon

Yapısal optimizasyonda ise problemler genellikle boyut, geometri ve topoloji optimizasyonu olarak 3 ana başlık altında toplanabilir.

1.4.1. Boyut optimizasyonu

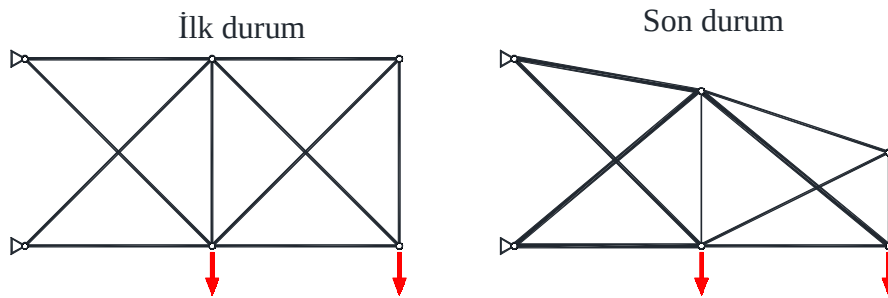
Yapıyı oluşturan elemanların en kesitleri tasarım değişkeni olarak kullanılan optimizasyon yöntemidir. Şekil 1.2.'de verilen bir kafes için elemanların sadece kesit alanları tasarım değişkeni olarak kabul edilen optimizasyon problemlerine boyut optimizasyonu denir. Sistemin geometrisi gelişmiş güzel eleman en kesiti seçilerek oluşturulur. Daha sonra sınırlara bağlı olarak bir metodoloji kullanılarak en kesitlerin değerleri değiştirilir. Boyut optimizasyonunda eleman sayısı ve eleman koordinatları değişmez.



Şekil 1.2. Boyut optimizasyonu

1.4.2. Şekil optimizasyonu

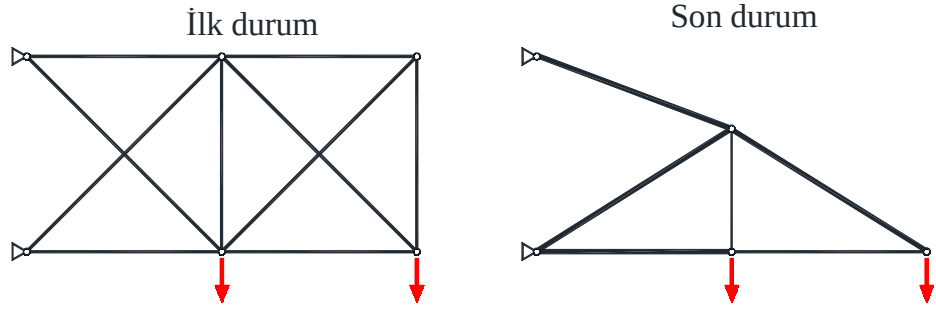
Tasarım değişkenleri, yapıyı oluşturan elemanların sayısını değiştirmeden sadece düğüm noktalarının koordinatlarını değiştirerek yapılmış ise buna şekil (Shape) optimizasyonu denir. Bu tür optimizasyonda tasarım değişkeni olarak düğüm noktaları koordinatları ve eleman en kesitleri kullanılır. Dolayısıyla şekil optimizasyonu ile boyut optimizasyonu birlikte yapılır.



Şekil 1.3. Şekil optimizasyonu

1.4.3. Topoloji optimizasyonu

Optimizasyonun en genel şeklidir. Taşıyıcı sistemi oluşturan elemanların sayısı, kesitleri, düğüm noktalarının yeri değişiyor ise bu tür optimizasyon problemleri topoloji optimizasyonu olarak isimlendirilir. Yalnız mesnetin ve yüklerin uygulanacağı noktaların koordinatları tasarım değişkeni olarak kabul edilmez.



Şekil 1.4. Topoloji optimizasyonu

1.5. Yapıların Optimizasyonunda Kullanılan Yöntemler

Yıllar içerisinde bir çok farklı optimizasyon yöntemleri geliştirilmiş ve kullanılmıştır. Bu yöntemler en genel hali ile analitik yöntemler, grafik yöntemler, deneysel yöntemler ve sayısal yöntemlerdir. Sayısal yöntemlerde kendi aralarında gradyan tabanlı yöntemler ve direkt arama yöntemleri olarak ayrılabilir.

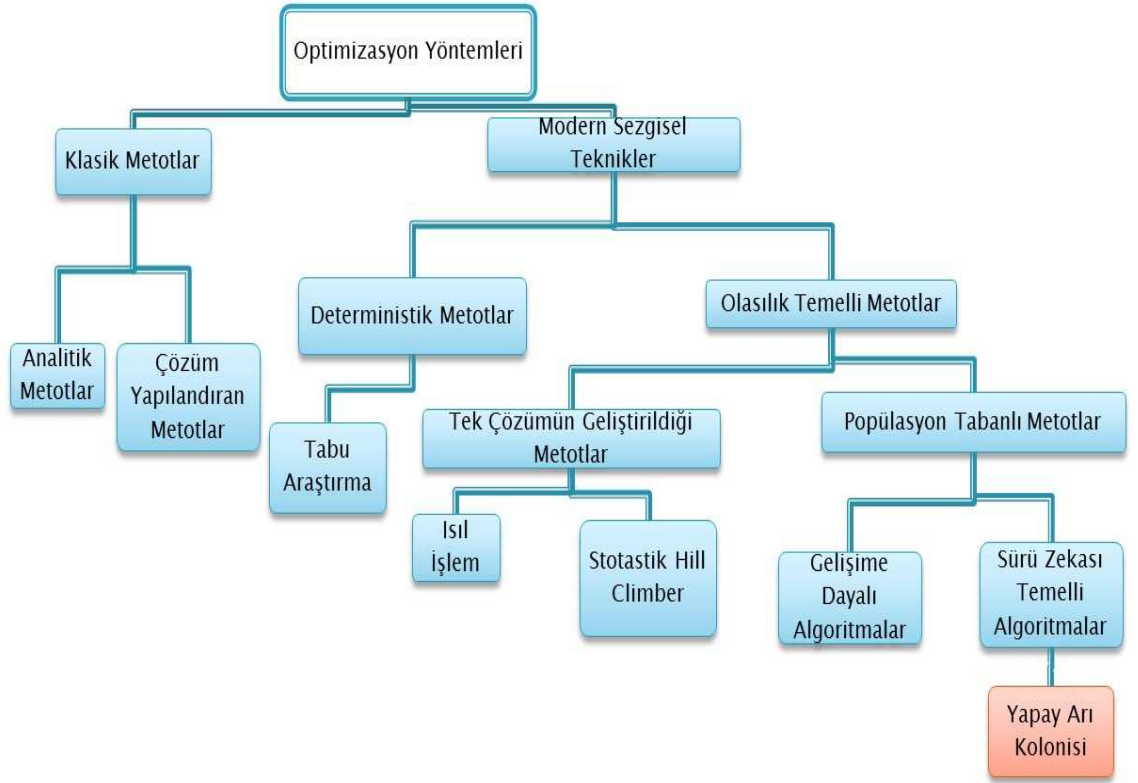
Analitik metotlar, diferansiyel matematik kullanılarak yapılan optimizasyona denir. Amaç fonksiyonunun tasarım değişkenlerine göre türevi alınarak sifıra eşitlenir. Bu şekilde problemi maksimum ve minimum yapan tasarım değişkenleri analitik olarak bulunmuş olur. Bu yöntemin bilgisayara ihtiyacı yoktur. Fakat bu yöntem doğrusallık derecesi çok fazla olan problemlere ve değişken sayısı üç veya daha fazla olan problemlere uygulanamaz.

Grafik yönteminde ise amaç fonksiyonu istenilen değerler arasında çizilerek fonksiyonu minimum veya maksimum yapan değerler kolaylıkla bulunabilir. Eğer amaç fonksiyonu tek değişkene bağlı ise sonucu bulmak çok kolay olacaktır. Eğer iki değişkenli ise üç boyutlu bir grafik çizilerek optimum noktalar hesap edilebilir. Değişken sayısı üç ve daha fazla ise bu durumlarda bu yöntemin kullanılması hemen hemen mümkün olmamaktadır. Bazı durumlarda bir sistemin optimum performansı deneme-yanılma yolu kullanılarak da bulunabilir. Bunun için önce tasarım değişkenlerinden birisi alınır ve belirli değerler verilerek amaç fonksiyonunun performansı hesaplanır. Amaç fonksiyonunu en iyi yapan tasarım değişkeni en yakın çözüm olarak kabul edilebilir. Ancak çok değişkenli problemlerde deneme-yanılma sayısı çok artacağı için kullanışlı bir yöntem olarak kabul edilmez.

Sayısal (nümerik) yöntemler optimizasyon için en çok kullanılan yaklaşımdır. Bu yaklaşımda, tasarım değişkenleri için bir başlangıç tahmini ile başlanır ve yinelemeli olarak her adımda tasarım değişkenleri bir önceki adımdaki bilgiler kullanılarak çözümler biraz daha optimum sonuca yaklaştırır. Anlaşılacağı gibi bu çözüm bir yinelemeli veya iteratif bir çözümdür. Bu çözümlerde tam (exact) çözüme ulaşamaz fakat iterasyon belli

bir sayıya gelindiğinde veya belli bir yakınlık elde edilince sonlandırılır. Onun için elde edilen sonuçlara optimale yakın (near optimal) çözümler denir.

Nümerik çözümlerde bir önceki adımdaki bilgiler kullanılarak bir sonraki adım tahmin edilirken kullanılan yöntemler birbirinden farklıdır. Bu yöntemler kendi aralarında iki kısma ayrılabilir. Birincisi gradyan temelli yöntemler (klasik yöntemler) ve ikincisi direkt yöntemlerdir (modern sezgisel yöntemler). Şekil 1.5.'de optimizasyon yöntemlerin basit bir sınıflandırılması gösterilmiştir.



Şekil 1.5. Optimizasyon yöntemleri (Akay, 2009)

1.5.1. Gradyan temelli yöntemler

Gradyan temelli optimizasyon yöntemleri sürekli ve türevi alınabilir fonksiyonların optimum çözümü için çok kullanışlıdır. Bu metotlarda analitik ve optimum noktayı belirlerken diferansiyel teknikler kullanılır. Çoğu zaman gerçek mühendislik problemleri, sürekli ve/veya türevi alınabilir olmadığından bu metotların kullanılabilirliği de çok sınırlıdır. Bunun yanında direk arama metotları olarak geliştirilen yöntemler gradyan tabanlı yöntemler kullanılarak geliştirildiğinden bu yöntemlerin bilinmesi işlemlerin nasıl yapıldığını anlamak için çok önemlidir. Gradyan temelli yöntemlerin bazıları ardışık kuadratik programlama ve artırılmış Lagrange yöntemleridir. Bu yöntemlerle ilgili birçok çalışma literatürde mevcuttur (Arora, 2007) .

1.5.2. Direkt (Modern Sezgisel) arama metotları

Modern Sezgisel arama metotları doğrudan çözüm uzayında rasgele sayı üreterek çözüm ararlar. Bu yöntemler modern ya da geleneksel olmayan optimizasyon yöntemleri olarak da adlandırılabilir. Bu yöntemlerin çoğu biyolojik davranış ve karakteristiği, moleküler böcek sürüsü ve nörobiyolojik sistemler üzerinde temellendirilir. Aşağıda bu yöntemlerden bazılarının adları verilmiştir.

- Genetik algoritmalar (Genetic Algorithm, GA),
- Isıl işlem optimizasyonu (Simulated Annealing, SA),
- Parçacık sürü optimizasyonu (Particle Swarm Optimization, PSO),
- Karınca kolonisi optimizasyonu (Ant Colony Optimization, ACO),
- Harmoni arama metodu (Harmony Search, HS)
- Bulanık mantık optimizasyonu (Fuzzy Logic Optimization, FLO),
- Yapay sinir ağları yöntemleri (Artificial Neural Network, ANN),
- Yapay arı kolonisi algoritması (Artificial Bee Colony, ABC)
- Öğretme-Öğrenme tabanlı optimizasyon yöntemi (Teaching-Learning Based Optimization, TLBO)

Bu yöntemlerin çoğu son yıllarda geliştirilmiş olmasına rağmen karmaşık mühendislik problem çözümleri için heyecan verici ilerlemeler göstermişlerdir. Bu yöntemlerde hem amaç fonksiyonun hem de sınır şartlarının türevlerine ihtiyaç duyulmaz, sadece fonksiyon değerlerine ihtiyaç vardır. Genetik algoritmalar doğal seleksiyon ve doğal genetik ilkelerini temel alır. Isıl işlem optimizasyonu (bu yönteme benzetimli tavlama denir) hassas olarak ısıtılmış termal tavlama simülasyonunu temel alır. Genetik algoritma ve ısı işlem optimizasyonu, ayrık optimizasyon problemlerin çözümü için doğal olarak uygulanabilir ve yüksek bir olasılıkla global minimumu bulabilen yöntemlerdir. Parçacık sürü optimizasyonu böcek sürüsü, kuş sürüsü ya da balık sürüsü gibi yaşam kolonilerinin davranışını temel alır. Karınca koloni optimizasyonu, yuvalarından yiyecek kaynağına en kısa yolu bulabilen gerçek karınca kolonilerinin kooperatif davranışlarını temel alır. Birçok pratik sistemde, objektif fonksiyon, sınırlamalar ve tasarım verileri sadece dil dönemleri ve belirsizlik olarak bilinirler. Bulanık mantık yöntemi bunun gibi problemleri çözmek için geliştirilmiştir. Sinir-ağ temelli yöntemlerde, problem birkaç nöron içeren bir ağ olarak modellenir ve ağ optimizasyon problemlerini etkili bir şekilde çözmek için uygun bir şekilde eğitilir. Direkt optimizasyon yöntemleri genellikle fiziksel veya biyolojik olaylar taklit edilerek geliştirilmiştir. Örnek metal tavlama fiziksel olayının metaforu olarak ısı işlem (Simulated Annealing) ve doğal genetik evrimin metaforu olan genetik algoritma (Genetic Algorithm) gibi iki popüler direkt algoritmalar geliştirilmiştir.

Bunun sonucu olarak amaç ve sınırlama fonksiyonları, gradyanlar değerlendirilmesine gerek kalmadan sonuç optimizasyon problemleri çözümlenebilir.

Arı algoritması kısa bir zaman önce geliştirilmiştir. Günümüzde yapı mühendisleri tarafından sık kullanılan bir algoritma değildir. Arı algoritması yiyecek arayan bal arılarının en iyi çiçek nektarını bulmak için izledikleri yoldan esinlenerek geliştirilen bir yöntemdir.

1.6. Literatür Çalışmaları

Son elli yılda en çok araştırma yapılan konulardan biriside yapısal optimizasyon problemlerinin çözümüdür. Optimizasyon çalışmaları genellikle sürekli değişkenli optimizasyon problemleri ve ayrık değişkenli optimizasyon problemleri olarak iki ana kolda devam etmiştir. Bunun ilk örneklerinden birisi Toakley 1968 çalışmasıdır. Toakley (1968) çalışmasında ayrık değişkenli yapısal bir optimizasyona gereksinim duyulduğuna dikkat çekmiştir. Reinschmidt (1971), çalışmasında çerçeve binaların plastik tasarım problemlerini dal ve sınır (branch and bound) yöntemi kullanarak çözmüştür.

Hua (1983), çalışmasında yer değiştirme ve gerilme ile sınırlandırılmış kafes sistemlerin ayrık değişkenli optimizasyon problemlerini çözmüştür. Shin vd., (1990), çalışmasında ayrık değişkenli olmayan değerler için bir ceza fonksiyonu tanımlanmıştır. Fu vd., (1991), çalışmasında beş küçük ölçekli mühendislik optimizasyon problemlerini çözmek için bir ceza fonksiyonu yaklaşımı kullanmıştır.

Ayrık değişkenli uzay kafes sistemlerin optimum tasarımı için farklı araştırmacılar tarafından bir çok direkt arama tabanlı optimizasyon yöntemi/algoritması önerilmiştir. Kafes sistemlerin tasarımı için önerilen yöntemlerden bazıları: genetik algoritma (Rajeev ve Krishnamoorthy, 1992; Deb ve Gulati, 2001; Marler ve Arora, 2004), ısıtım işlem algoritması (Kripka, 2004; Hasançebi vd., 2011), sezgisel parçacık sürü algoritması (Pereza ve Behdinan, 2007, Li vd., 2009), karınca koloni algoritması (Camp ve Bichon, 2004), harmoni arama (Harmony Search) (Lee vd., 2005), yapay arı koloni optimizasyon algoritmasıdır (Sonmez, 2011a; Sonmez vd., 2013, Hadidi vd, 2010). Bazı araştırmacılar ise yukarıda sözü edilen direk arama metotlarını geliştirerek veya iki farklı yöntemi birleştirerek yeni yöntemler elde etmişlerdir. Bu tür algoritmalara örnek olarak parçacık sürüsü karınca koloni optimizasyonu verilebilir (Kaveh ve Talatahari, 2009a, 2009b).

Düzlemsel çelik çerçevelerin optimum tasarımı için genellikle direkt arama optimizasyon teknikleri geliştirilmiştir. Bunun en büyük nedeni, çelik çerçevelerde kullanılan profillerin özel üretim olmayıp hazır kullanılan profillerden seçilmesidir. Çelik çerçeveler için kullanılan tekniklerden bazıları şunlardır: Genetik algoritma (Kameshki ve Saka,

2001; Gero vd., 2006; Safari vd., 2011), Tabu arama metodu (Değertekin vd., 2008), Harmani arama (Saka, 2009; Değertekin 2012), karınca kolonileri (Kaveh vd., 2010), Parçacık sürü algoritması (Doğan ve Saka, 2012), Büyük patlama-büyük büzülme (Hasancebi ve Azad, 2012, 2013), Öğretme-Öğrenme tabanlı optimizasyon (Togan, 2012), yapay arı kolonisi (Değertekin, 2012; Aydogdu vd., 2012).

2. YAPAY ARI KOLONİ (ABC) ALGORİTMASI

2.1.Sürü Zekası

Sürü terimi birlikte davranış sergileyen karınca, arı kolonileri, balık, kuş ve böcek toplulukları için kullanılmaktadır. Bir sürüdeki bireyler herhangi bir danışma işlemi olmadan hareket ederler. Herhangi bir ilişkileri olmadan kendi alanlarındaki kurallar vasıtasıyla kendi kendine organize olabilen bireyler arasındaki etkileşim sürü zekâsı adı verilen birlikte davranış zekâsının ortaya çıkmasına sebep olmaktadır. Kendi kendine organize olabilme sürü sisteminin temel özelliğidir (Akay, 2009). Bonabeau (1999) kendi kendine organize olabilmeyi dört özellik ile karakterize etmiştir.

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

2.1.1. Arıların sürü zekası

Doğadaki sürülerden biriside her türlü işbirliğinde topluluk zekâlarıyla cevap veren bal arısı sürüsüdür. Bal arıları fotoğrafik hafızaya, uzay çağı sensörlerine, yön bulma sistemine, sezgisel kavrama yeteneğine, herhangi bir iş birliğinde grup olarak karar verme özelliğine sahiptirler. Bu özellikler araştırmacıları arıların davranışlarını modellemeleri yönünde teşvik edici olmuştur.

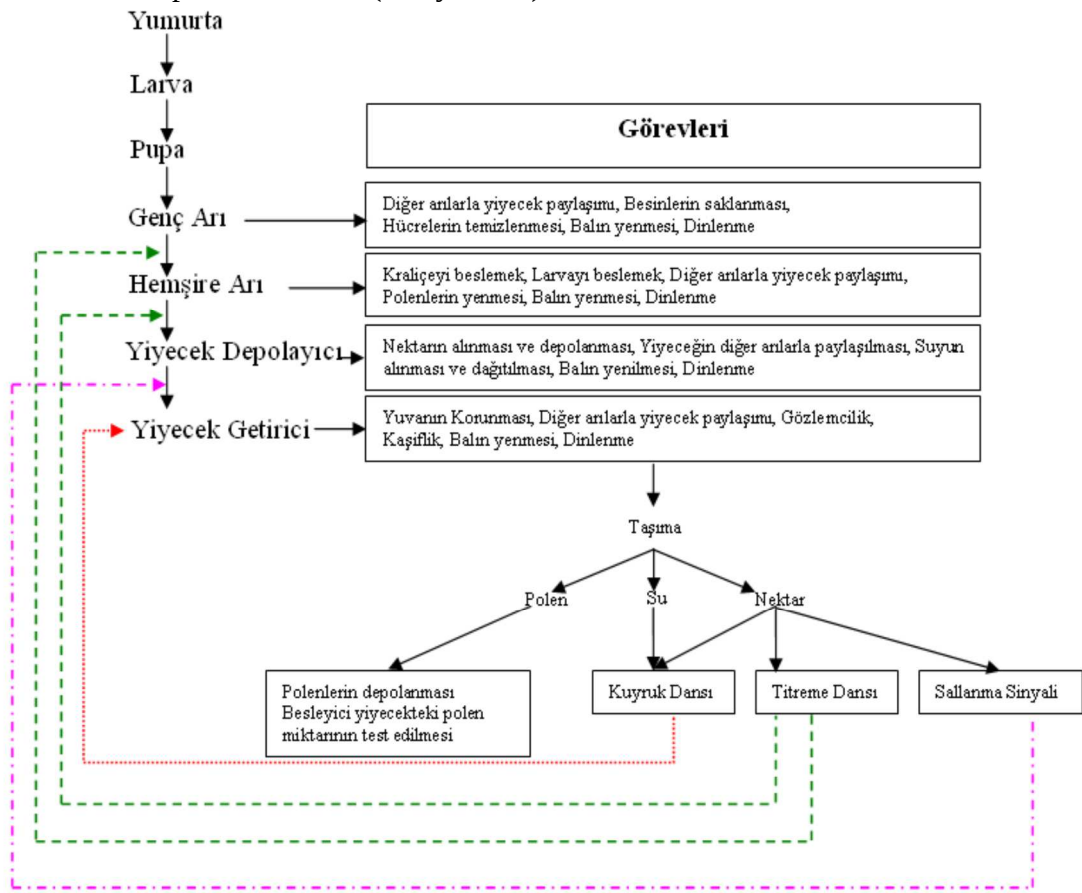
Arılar koloniler halinde yaşayan böceklerdir. Bir kolonide üç çeşit arı bulunmaktadır. Kraliçe arı, erkek arı ve dişi olan işçi arıdır (Akay, 2009).

- Kraliçe Arı: Kraliçe kolonide birkaç yıl yaşayabilmektedir. Kolonideki tüm arıların annesidir. Aynı zamanda kraliçe arı yumurtlama özelliğine sahip tek arıdır.
- Erkek Arılar: Kolonideki arıların babası erkek arılardır. Erkek arılar döllenmemiş yumurtalardan oluşurken kraliçe ve işçi arılar döllenmiş yumurtalardan oluşurlar. Yaşama süreleri 6 ayı geçmemektedir. Erkek arıların asıl görevi kraliçe arının döllenmesidir. Erkek arılar kraliçe arı ile çiftleşmeleri sonucu ölürler.

- **İşçi Arılar:** İşçi arılar kovandaki iş bölümünden sorumludurlar. Yiyecek toplanması, yiyeceklerin saklanması, kovanın temizliği, kovanın bakımı ve kovanın güvenliğinden sorumludurlar. Yaşamlarının belirli bir bölümünde ise yiyecek arayıcı (forager) olarak çalışırlar.

2.1.2. Yiyecek arama

Kovandaki en önemli aktivite yiyecek aramaktır. Bal arılarının yiyecek arama davranışlarını koku, sallanma (waggle) danstan gelen bilgiler ve hafızada tutulan kaynakların etkilediği yapılan çalışmalarla belirtilmiştir. Yiyecek arayan arılar kovan dışından bal, polen ve su ile dönmektedirler. Yiyecek arama süreci (forager) arının kovandan ayrılmasıyla başlamaktadır. Bir kaynak bulunduktan sonra, arı bu çiçeğin nektarını bal deposunda saklar (Akay, 2009).



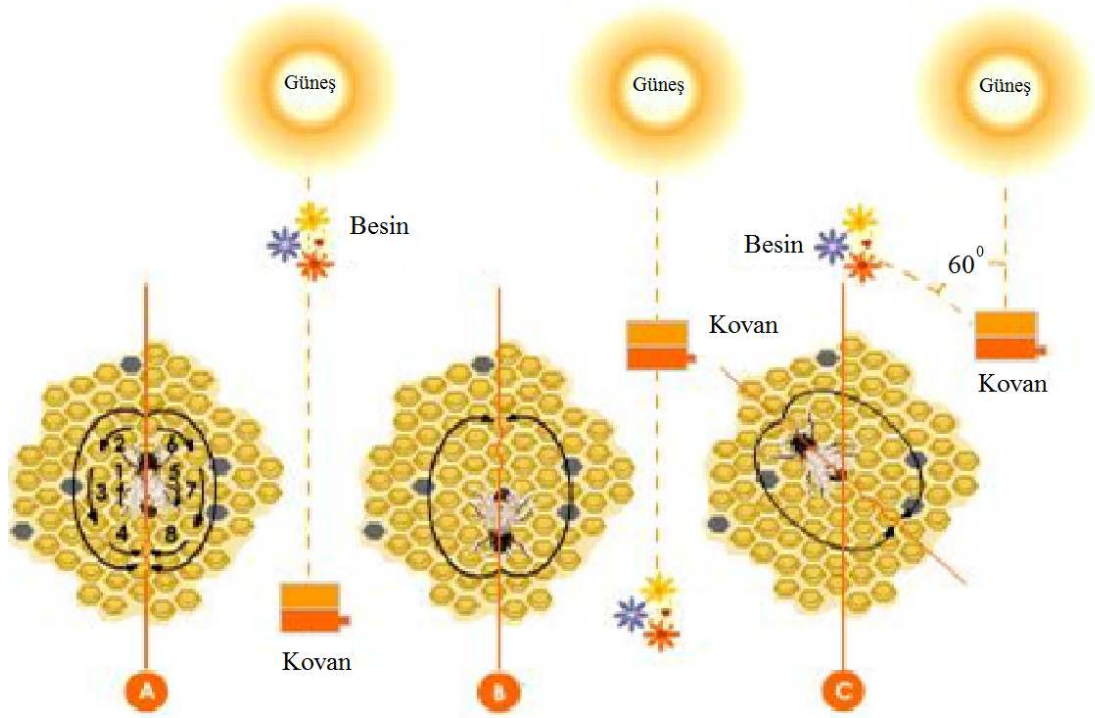
Şekil 2.1. Arılarda görev paylaşımı (Akay, 2009)

2.2. Doğadaki Arıların Davranışı

Bir bal arısı kolonisi tek bir kraliçe arı, binlerce dişi işçi arı ve yüzlerce erkek arıdan oluşur. Bal arıları koloni halinde yaşayan sosyal böceklerdir. Kraliçenin görevi erkek arılar ile çiftleşerek yeni koloniler oluşturmaktır. Bir kolonideki tüm görevler; gıda işleme, bu gıdaların depolanması, hücrelerin temizliği, petek inşa etmek ve kovayı

koruma gibi görevler henüz 3 haftalık olmamış dişi arılar tarafından yapılır. Kovan içerisine çalışan bu işçi arılar yaklaşık üç haftalık olduğunda, kovan içerisindeki tüm işlerini bırakırlar ve çiçeklerden nektar toplayan birer izci arı olurlar. Bal arılarının görev dağılımları Şekil 2.1. 'da verilmiştir.

Bir bal arısı yaklaşık olarak kovandan 12 kilometre uzağa nektar bulmak için gidebilir ancak genellikle bu mesafe 3 km'yi geçmez. Eğer bir arı çiçeklerin birinde nektar keşfederse, bu nektarı alır, kovana götürür ve gerekli yere depo ettikten sonra "sallanma dansı" (Waggle Dance) ile nektarın bulunduğu yeri diğer izci arılara iletir (Karaboğa ve Baştürk 2008, Karaboğa 2005). Şekil 2.2.'de bir kovan, üç gıda kaynağı ve bu gıda kaynağını keşfeden arılar tarafından gerçekleştirilen sallanma dansı şematik olarak gösterilmiştir.



Şekil 2.2. Bal arılarının sallanma dansı (Waggle Dance)

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

1. Sekize benzeyen şeklin ortasındaki düz çizgi besin kaynağının yönünü,
2. Sekize benzeyen şeklin ortasındaki düz çizginin uzunluğu besin kaynağının kovandan ne kadar uzakta olduğunu,

3. Sekize benzeyen şeklin ortasındaki düz çizginin alındığı süre ise nektar kalitesi hakkında bilgi verir.

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

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

Her arı nektarın kaynağındaki gıda düzeyine bağlı olarak yukarıdaki seçeneklerden birini izler. Bir arı belirli bir sınırın üzerinde olan nektar kaynağı bulursa, o arı 3. seçeneği izler. Eğer besin kaynağındaki nektar orta düzeyde ise, arı kovandaki arılara besin kaynağının yerini haber vermeden tekrar aynı besin kaynağına gider. Eğer arı 3. ve 2. yolu izlemeyecek ise bu durumda 1. yolu izleyerek yeni umut verici nektar kaynaklarını aramaya devam eder (Karaboğa ve Baştürk 2008, Teodoroviç 2009, Sonmez 2011a, 2011b).

2.3. Arı Koloni Algoritması

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

$$^0_i x_j = \min x_j + \lambda (\max x_j - \min x_j), \quad \min x_j \leq x_j \leq \max x_j \quad (2.1)$$

Yukarıdaki eşitlikte, λ değeri 0 ile 1 arasında rastgele bir sayıdır. Sağ alt indis her tasarım değişkeninin değişken numarasını, sol üst indis değişkenin alabileceği maksimum, minimum değeri ve sol alt indis ise çözümün hangi besin kaynağında gerçekleştiğini göstermektedir. Denklem 2.1’de verilen $\max x_j$ ve $\min x_j$ değerleri tasarım değişkeninin önceden bilinen tasarım sınırlarıdır. Eğer tasarım değişkenleri sürekli ise Denklem 2.1’de verildiği gibi gösterilir. Eğer tasarım değişkenleri bir listeden seçilecek ise bu durumda Denklem 2.1 aşağıdaki şekilde yazılabilir.

$$^0_i I_j = \min I_j + \lambda (\max I_j - \min I_j), \quad \min I_j \leq I_j \leq \max I_j \quad (2.2)$$

Denklem 2.2’de verilen I değerleri değişkenin değeri değil tasarım değişkeninin indisini gösterir. Örnek olarak eğer tasarım değişkeni bir listeden seçilecek ise I değerleri aşağıdaki tablodan olduğu gibi seçilir.

Çizelge 2.1. Tasarım değişkenine göre I değerleri tablosu

Tasarım Değişkeni	x_1	x_2	x_3	x_4	..	...	x_i	...	x_n
Değişken İndisi, i	1	2	3	4	...	...	i	...	n

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

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

$$p_i = \frac{\frac{1}{i^{f(x)}}}{\sum_{m=1}^n \frac{1}{m^{f(x)}}} \quad (2.3)$$

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

$$x_j^{yeni} = x_j^{isth} + \phi(x_j^{isth} - x_j^{isth_k}) \quad (2.4)$$

Yukarıdaki eşitlikte, ϕ değeri -1 ile 1 arasında rastgele bir reel sayıdır. Sağ alt indis (j) tasarım değişkenin adını gösterirken sol alt indisler (i) besin kaynağının numarasını temsil

eder, (k) ise 1 ile SN arasında i ye eşit olmayan bir tam sayıyı temsil eder. Bu formül diğer gıda kaynağı arılarının arama faaliyetlerini kontrol etmek için kullanılır. $isth_k x_j$ ABC yakınsama davranışlarında önemli bir rol oynar. Bulunduğu yerde besin seviyesi önceki en iyi besin kaynağına göre daha iyi ise, bulunduğu konum iyi besin kaynağı özelliği kazanır yani $isth_k x_j = yeni_k x_j$ olur. Aksi takdirde, eski konumu en iyi besin kaynağı olarak korunur.

Daha önceden de söylendiği gibi ABC nümerik algoritmalarından olduğundan dolayı, çözüm iteratiftir. Önceden tanımlanmış bir yinelemeden (LIMIT) sonra belirli bir besin kaynağının nektar miktarında herhangi bir gelişme yoksa bu besin kaynağı istihdam edilen arı tarafından iptal edilir ve bir daha buraya gidilmez. Sonra bu istihdam edilen arı bir "izci arı" (Teodoroviç, 2009) olur. Bir izci arı herhangi bir rehberlik hizmeti olmadan araştırma yaparken bir yerde iyi bir besin kaynağı bulursa bu izci arı bu besin kaynağı için istihdam edilir.

Bu arama işlemi önceden belirtilen maksimum adım sayısına ulaşıncaya kadar devam eder. Arılar içerisindeki en iyi nektar miktarı amaç fonksiyonunun değeri, besin kaynağının yeri ise tasarım değişkenlerinin değeri olarak kabul edilir. Bulunan tasarım değişkenlerinin değeri optimuma yakın sonuçlar verir. Analitik hesaplar gibi tam hesap sonucunu vermez.

2.4. Sınırlı Optimizasyonların Sınırsız Optimizasyon Olarak Çözülmesi

Yapısal optimizasyonun amacı, en güvenli yapıları en ekonomik olarak tasarlamaktır. Yapının güvenli olması her bir elemandaki gerilmelerin emniyet gerilmelerini aşmamış olmasına, her bir düğüm noktasındaki yer değiştirmelerin müsaade edilen yer değiştirmeden fazla olmamasına ve basınç elemanlarının burkulmadan gelen yükleri emniyetle taşıyabilmesine bağlıdır. Bu tür zorunluluklara optimizasyonda kısıt denir ve hemen hemen tüm gerçek mühendislik problemlerinde bu tür kısıtlar vardır.

Tüm nümerik optimizasyon algoritmalarında kısıtlarla baş etmek için çeşitli yöntemler geliştirilmiştir. Belki de bunların en basiti Deb'in yöntemidir. Deb'in (2000), "seçim mekanizması" kısıtları optimizasyona dahil etmek için kullanılabilir. Deb'in yöntemi iki çözümü karşılaştırmak için üç çok basit kural içerir:

1. Eğer iki çözümde kısıtları sağlıyorsa, uygun çözüm her zaman amaç fonksiyonunun en iyi olanıdır.
2. Eğer iki çözümden biri kabul edilebilir bölgede ve diğeri kabul edilemez bölgede ise bu durumda her zaman kabul edilebilir bölgedeki amaç fonksiyonu seçilir.

3. İki çözümde kabul edilir bölgede değil ise, kısıtları en az ihlal eden çözüm en iyi çözüm olarak kabul edilir.

Deb'in seçim mekanizmasının avantajı, herhangi bir ceza fonksiyonu kullanılmaya ihtiyaç duyulmaz ve iki çözümde sadece biri seçilerek işlem yapılır. Bu yöntem diğer yöntemlerin çoğundan daha kolaydır. Deb'in yöntemi aday çözümlerini kabul edilebilir bölgede olmasını ve onları sınırlarının içerisinde kalmasını zorlar. Bu optimizasyon algoritmasının kabul edilemez bölgede bir çözüm üzerinde yaklaşmasını önler fakat çözümün süresini uzatır.

Nümerik optimizasyon algoritmalarında kısıtlarla baş etmek için kullanılan yöntemlerde biriside ceza fonksiyonu yöntemidir. Ceza yöntemi de farklı şekillerde uygulanmaktadır. Bunlar arasında statik ceza yaklaşımı, dinamik ceza yaklaşımı, tavlama yaklaşımı ceza, ayarlanabilir (adaptive) ceza ve ortak-evrimleşme (co-evolutionary) ceza yaklaşımı sayılabilir. Genel olarak ceza temelli evrimsel hesaplama yöntemlerinde kullanılan ceza fonksiyonu Denklem 2.5'deki gibi verilmektedir (Coello, 2002)

$$F(\vec{x}) = f(\vec{x})[1 \pm \text{Penalty}(\vec{x})] = f(\vec{x})[1 \pm \sum_{i=1}^m r_i * g_i + \sum_{j=1}^p c_j * L_j] \quad (2.5)$$

Eğer normalize edilen kısıtlar sıfırdan büyük ise yani kabul edilemez bölgede bulunduğu durumlarda bir ceza katsayısı (r_i veya c_j) ile çarpılarak amaç fonksiyonun değeri bir belli miktarda artırılır veya azaltılır. Eğer optimizasyon amaç fonksiyonunu en düşük seviyeye indirme (minimizasyon) problemi ise Denklem 2.5 'in ikinci tarafındaki işaret artı (+) ve eğer amaç fonksiyonunu en yüksek seviyeye çıkarma (maksimizasyon) problemi ise işaret eksi (-) olacaktır.

Denklem 2.5' de verilen fonksiyonda kullanılan ceza katsayıları (r_i ve c_j) farklı yaklaşımlar için farklı olarak hesaplanır. Eğer statik ceza fonksiyonu kullanılacak ise bu durumda bu katsayılar analiz sırasında değişmez. Dinamik ceza katsayılarının kullanıldığı yaklaşımlarda, ceza katsayıları r_i^λ ve c_j^μ (λ ve μ sayıları önceden tanımlanmış 1 ile 2 arasında bir reel sayı) şeklinde gösterilir.

Ayarlanabilir (adaptif) ceza fonksiyonu uygulamasında ise ceza katsayısı Denklem 2.6 gibi kullanılır. Bu yaklaşımda ceza katsayısı r yinelemenin bir fonksiyonu olarak kullanılır ve $r(t)$ nin alabileceği değerler Denklem 2.7'de verilmiştir. Denklemden de anlaşılacağı gibi eğer çözüm birçok kez kabul edilemeyecek bölgede ise bu durumda ceza katsayısı her adımda daha da büyüyecek ve bulunan çözüm terkedilecektir. Bunun tersine eğer çözüm kabul edilebilir bölgede olursa bu durumda ceza katsayısı küçülecek ve çözüm daha uygun hale gelecektir.

$$F(\vec{x}) = f(\vec{x})[1 \pm r(t) * (\sum_{i=1}^m g_i + \sum_{j=1}^p L_j)] \quad (2.6)$$

$$r(t) = \begin{cases} \frac{1}{\omega} * r(t-1) & \text{eğer } F(t-1) \text{ kabul edilebilir bölge} \\ \omega * r(t-1) & \text{eğer } F(t-1) \text{ uygun olmayan bölge} \end{cases} \quad (2.7)$$

Denklem 2.7’da verilen ω ’ değeri tavlama yaklaşımında 1.1 olarak kullanılması tavsiye edilirken, Sonmez (2011a) bunun yerine toplam kısıt sayısının bir fonksiyonu olacak şekilde bir denklem önermiştir. Sönmez’in önerdiği fonksiyon Denklem 2.8’de verilmiştir.

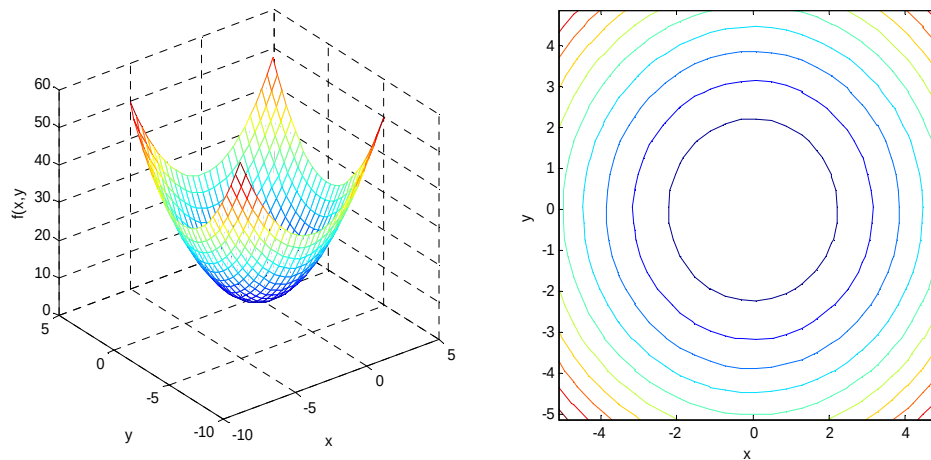
$$\omega = 1 + \frac{1}{n_c} > 1.01 \quad (2.8)$$

Bu denklemde n_c toplam kısıt sayısıdır ve optimizasyon problemi çözülmeye başlamadan önce bilinen bir değerdir. Optimizasyon problemlerin çözümleri genellikle kısıt yüzeylerine yakın yerlerde bulunur bundan dolayı da çözümler kabul edilmeyen bölgede olabilir ama optimum çözüme de çok yakın olabilir. Bu durumlarda çözüm kabul edilemeyen bölgede olduğu için reddetmek optimumdan uzaklaşmak anlamına gelecektir. Bundan dolayı özellikle tasarım değişkenlerinin sürekli olduğu problemler için Denklem 2.8’de verilen adaptif ceza fonksiyonu iyi sonuçlar verdiği gözlenmiştir.

Anlatılan arı algoritmasını daha iyi anlaşılması için iki adet optimizasyon problemi çözülecektir. Bunlardan birincisi bir fonksiyonun optimizasyonu, diğeri ise bir mühendislik probleminin optimizasyondur.

2.5. Örnek 1

Bir $f(x,y) = x^2 + y^2$ fonksiyonunu minimum yapan x ve y değerini hesaplayınız. x ve y nin alabileceği en küçük değer $-5,12$ ve en büyük alabileceği değer ise $5,12$ olsun. Eğer grafiksel olarak çözümlürse aşağıdaki grafik elde edilir. Bu grafiğe göre fonksiyonu minimum yapan değerler $f(x = 0, y = 0) = 0^2 + 0^2 = 0$ olarak bulunur.



Şekil 2.3. $f(x,y) = x^2 + y^2$ fonksiyonunun grafiksel çözümü

Bu soruyu ABC algoritması kullanarak aşağıda belirtilen ABC algoritması adımlarını izleyerek çözelim. Fonksiyonu $f(x,y) = x^2 + y^2$ genellikle $f(x_1, x_2) = x_1^2 + x_2^2$ olarak gösterilebilir.

1. Algoritma arı sayısı (N) ve maksimum iterasyon sayısı (MNC) seçimi ile başlar. Bu problem için $N = 10$ ve $MNC = 50$ olarak seçilmiştir.
2. Denklem 2.1 kullanılarak ilk tasarım değişkenlerinin değeri ve bu tasarım değişkenleri kullanılarak amaç fonksiyonunun değerleri hesaplanır. Denklem 2.1 deki rastgele sayılar x_1 ve x_2 değişkeni için ayrı ayrı hesaplanmıştır.

Çizelge 2.2. $x^2 + y^2$ için ilk tasarım değişkenlerinin değeri ve bu tasarım değişkenleri kullanılarak amaç fonksiyonunun değeri

Arı No	x_1 için λ	$-5.12 < X_1 < 5.12'$	x_2 için λ	$-5.12 < X_2 < 5.12'$	Amaç Fonksiyonu
1	0.289	-2.162	0.256	-2.496	10.905
2	0.888	3.968	0.754	2.605	22.528
3	0.384	-1.192	0.938	4.486	21.542
4	0.905	4.148	0.303	-2.020	21.290
5	0.439	-0.624	0.249	-2.566	6.975
6	0.074	-4.359	0.563	0.644	19.414
7	0.582	0.841	0.858	3.669	14.172
8	0.337	-1.670	0.415	-0.875	3.557
9	0.350	-1.536	0.264	-2.413	8.181
10	0.942	4.529	0.080	-4.299	38.988

En iyi ilk beş arı istihdam edilen arı olarak seçilir. Diğer arılar izleyici arı olur.

Çizelge 2.3. $x^2 + y^2$ için istihdam edilen en iyi ilk beş arı

Arı No	x_1 için λ	$-5.12 < X_1 < 5.12'$	x_2 için λ	$-5.12 < X_2 < 5.12'$	Amaç Fonksiyonu
8	0.337	-1.670	0.415	-0.875	3.557
5	0.439	-0.624	0.249	-2.566	6.975
9	0.350	-1.536	0.264	-2.413	8.181
1	0.289	-2.162	0.256	-2.496	10.905
7	0.582	0.841	0.858	3.669	14.172

3. Her bir besin kaynağına gidecek izleyici arıların sayısı Denklem 2.3 kullanılarak aşağıdaki gibi hesaplanır.

$$p_8 = \frac{\frac{1}{3.557}}{\frac{1}{3.557} + \frac{1}{6.975} + \frac{1}{8.181} + \frac{1}{10.905} + \frac{1}{14.172}} = 0,40 \text{ ise gönderilecek arı sayısı } 2 = 2$$

$$p_5 = \frac{\frac{1}{6.975}}{\frac{1}{3.557} + \frac{1}{6.975} + \frac{1}{8.181} + \frac{1}{10.905} + \frac{1}{14.172}} = 0,202 \text{ ise gönderilecek arı sayısı } 1,01 \approx 1$$

$$p_9 = \frac{\frac{1}{8.181}}{\frac{1}{3.557} + \frac{1}{6.975} + \frac{1}{8.181} + \frac{1}{10.905} + \frac{1}{14.172}} = 0,172 \text{ ise gönderilecek arı sayısı } 0,86 \approx 1$$

$$p_1 = \frac{\frac{1}{10.905}}{\frac{1}{3.557} + \frac{1}{6.975} + \frac{1}{8.181} + \frac{1}{10.905} + \frac{1}{14.172}} = 0,129 \text{ ise gönderilecek arı sayısı } 0,65 \approx 1$$

$$p_7 = \frac{\frac{1}{14.172}}{\frac{1}{3.557} + \frac{1}{6.975} + \frac{1}{8.181} + \frac{1}{10.905} + \frac{1}{14.172}} = 0,0995 \text{ ise gönderilecek arı sayısı } 0,49 \approx 0$$

Buna göre 8 numaralı besin kaynağını 2 izleyici arı 5, 9 ve 1 numaralı besin kaynaklarını ise 1 er izleyici arı ziyaret edecektir. Diğer besin kaynağını ise izleyici arı gitmeyecek sadece istihdam edilen arılar gidecektir.

4. Sırası ile 8, 5, 9, 1 ve 7 numaralı besin kaynaklarına bir istihdam edilen arı ve yukarıda hesaplandığı kadar izleyici arı gönderilir. Örnek olarak 8 numaralı besin kaynağının ilk değerleri bir önceki tabloda verilmiştir. İstihdam edilen arı ve izleyici arı için x_1 ve x_2 değişken değerleri Denklem 2.4 kullanılarak hesaplanmış ve aşağıdaki tabloda gösterilmiştir. Örnek olarak 8 numaralı besin kaynağına giden istihdam arı için

$$yeni_8x_1 = -1,67 + 0,869 * (-1,670 - (-0,624)) = -2,58$$

$$yeni_8x_2 = -0,875 + 0,031 * (-0,875 - (3,669)) = -1,014$$

yeni tasarım değişkenleri hesaplanmış olur. Amaç fonksiyonlu da aşağıdaki gibi hesaplanır.

$$f_{yeni} = -2,58^2 + (-1,014)^2 = 7,687$$

Çizelge 2.4. $x^2 + y^2$ istihdam edilen arılar ve varsa izleyici arılar için amaç fonksiyonu

Arı No		x1			x2			Amaç Fonksiyonu
		k	ϕ	$-5.12 < X1 < 5.12'$	k	ϕ	$-5.12 < X2 < 5.12'$	
8	Önceki Adım			-1.670			-0.875	3.557
	İstihdam	5	0.869	-2.580	7	0.031	-1.014	7.687
	İzleyici 1	9	0.571	-1.747	5	-0.622	-1.928	6.769
	İzleyici 2	7	0.606	-3.193	7	0.636	-3.766	24.375
5	Önceki Adım			-0.624			-2.566	6.975
	İstihdam	9	0.605	-0.072	7	0.934	-5.120	26.220
	İzleyici 1	8	-0.473	-1.119	9	-0.926	-2.424	7.129
9	Önceki Adım			-1.536			-2.413	8.181
	İstihdam	1	-0.124	-1.614	8	-0.503	-1.639	5.290
	İzleyici 1	7	-0.370	-0.656	8	-0.655	-1.406	2.408
1	Önceki Adım			-2.162			-2.496	10.905
	İstihdam	9	0.398	-2.762	8	0.764	-3.734	21.573
	İzleyici 1	5	-0.339	-1.640	5	0.384	-2.469	8.786
7	Önceki Adım			0.841			3.669	14.172
	İstihdam	8	0.016	0.881	8	-0.219	2.673	7.921

Aynı işlemler izleyici arılar içinde yapılarak. Daha sonra 8 numaralı besin kaynağı için en avantajlı yani amaç fonksiyonunu en küçük yapan arı burada artık çözüm olarak alınarak bir sonraki adıma başlanır. Tabloda iyi çözümü bulan arılar boyanarak gösterilmişlerdir. Bu birinci yinelemenin sonudur.

5. İkinci yineleme başlamadan önce her bir besin kaynağına kaç adet arı gönderileceği araştırılır ve daha sonra bir önceki adımların aynısı yapılır.

Çizelge 2.5. $x^2 + y^2$ yeni bir yenileme için istihdam edilen arılar ve varsa izci arılar için amaç fonksiyonu

Arı No		x1			x2			Amaç Fonksiyonu
		k	ϕ	-5.12 < X1 < 5.12'	k	ϕ	-5.12 < X2 < 5.12'	
8	Önceki Adım			-1.670			-0.875	3.557
	İstihdam	9	-0.457	-1.207	1	0.528	-0.033	1.457
	İzleyici 1	5	0.068	-1.742	9	-0.405	-1.090	4.223
5	Önceki Adım			-0.624			-2.566	6.975
	İstihdam	1	0.634	0.020	1	0.795	-2.643	6.988
	İzleyici 1	8	-0.845	-1.116	7	-0.410	-0.418	1.421
9	Önceki Adım			-0.656			-1.406	2.408
	İstihdam	7	-0.836	0.628	8	0.752	-2.439	6.343
	İzleyici 1	5	-0.792	-1.021	7	0.872	-4.963	25.673
	İzleyici 2	8	-0.055	-0.687	7	-0.574	0.935	1.345
1	Önceki Adım			-1.640			-2.469	8.786
	İstihdam	5	-0.667	-1.291	9	-0.032	-2.361	7.241
7	Önceki Adım			0.881			2.673	7.921
	İstihdam	5	-0.564	-0.246	9	-0.664	1.519	2.369

6. Bu işlemler maksimum yineleme sayısına kadar devam ettirilir. Bu problem için 50 kere tekrarlanırsa en son çözüm aşağıdaki gibi bulunur. Bu tabloda çözümü veren satır taranmıştır. Sonuç

ABC algoritması: $f(x_1 = -1,112 \times 10^{-6}, x_2 = -2,287 \times 10^{-5}) = 6,699 \times 10^{-10}$

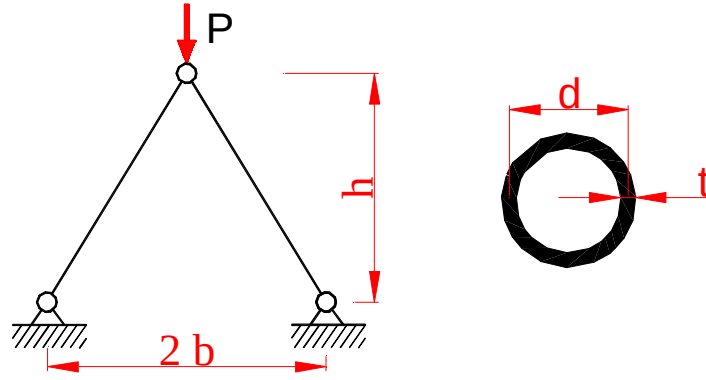
Analitik sonuç : $f(x_1 = 0, x_2 = 0) = 0$

Çizelge 2.6. $x^2 + y^2$ için 50. yenileme sonrası en son amaç fonksiyonu

Arı No	x1 için λ	-5.12 < X1 < 5.12'	x2 için λ	-5.12 < X2 < 5.12'	Amaç Fonksiyonu
1	-0.271	-1.112E-06	0.437	-2.586E-05	6.699E-10
9	-0.286	-8.303E-05	-0.944	4.029E-05	8.518E-09
8	0.161	1.260E-04	-0.428	3.480E-05	1.709E-08
7	-0.766	-1.075E-04	-0.576	2.016E-04	5.220E-08
5	-0.655	-1.800E-03	0.173	6.008E-03	3.934E-05

2.6. Örnek 2

Bu örnekte sınırlı bir yapı optimizasyon problemi çözülecektir. Şekilde verilen iki elemanlı bir kafes yapısında bir adet P yüklemesi etkitiliyor. Kafes yapıyı oluşturan yapı elemanları dairesel kesitli borulardan imal edilmiştir. Her bir elemandaki gerilmelerin akma mukavemetini geçmemelidir. Burkulma şartı göz önüne alınarak minimum malzeme için gerekli olan en kesit çapı ve kafes yüksekliği ne olmalıdır? Kafesin geometrik ve malzeme özellikleri: Elastisite Modülü $E = 29000$ ksi (200GPa), mesnetler arası uzaklığın yarısı $b = 100$ in (254 cm), akma gerilmesi $\sigma_y = 36$ ksi (0,248 GPa) , dairesel kesitli elemanların et kalınlığı $t = 0.25$ in (0,635 cm), uygulanan kuvvet $P = 100$ kips (444,82 kN)'dir.



Şekil 2.4. İki elemanlı kafes

Kafes eleman için lineer analiz yapılırsa her bir elemandaki eleman uzunlukları $L = \sqrt{h^2 + b^2}$ kuvvetleri $F = F_1 = F_2 = \frac{P}{2} * \frac{\sqrt{b^2 + h^2}}{h} = \frac{PL}{2h}$, atalet momenti $I = \frac{\pi t d}{8} (d^2 + t^2)$, gerilmeler $\sigma = \frac{F}{A}$ ve burkulma gerilmesi $\sigma_{cr} = \frac{\pi^2 EI}{L^2} \frac{1}{A}$ olarak bulunur. Amaç fonksiyonu kafes sistemin hacmi olarak aşağıdaki gibi hesaplanabilir.

$$V = 2AL = 2 \pi t d \sqrt{h^2 + b^2} \quad (2.9)$$

$$f(h, d) = 2 \pi t d \sqrt{h^2 + b^2} \quad (2.10)$$

İlk tasarım kısıtı için eleman üzerindeki gerilmeler müsaade edilen gerilmelerden az olması gerekiyor. Matematiksel olarak gösterilirse;

$$\frac{F}{A} - \sigma_y \leq 0 \quad (2.11)$$

$$g_1(h, d) = \frac{P}{2} * \frac{\sqrt{b^2 + h^2}}{h} * \frac{1}{\pi d t} - \sigma_y \leq 0 \quad (2.12)$$

İkinci tasarım kısıtı için eleman üzerindeki gerilme burkulmadan meydana gelecek gerilmenden az olmalıdır.

$$\sigma - \sigma_{cr} \leq 0 \quad (2.13)$$

$$\frac{P}{2} * \frac{\sqrt{b^2 + h^2}}{h} * \frac{1}{\pi d t} - \frac{\pi^2 EI}{L^2} \frac{1}{A} \leq 0 \quad \text{veya} \quad (2.14)$$

$$g_2(h, d) = \frac{P}{2} * \frac{\sqrt{b^2 + h^2}}{h} * \frac{1}{\pi d t} - \frac{\pi^2 E (d^2 + t^2)}{8 (h^2 + b^2)} \leq 0 \quad (2.15)$$

Sonuç olarak optimizasyon problemi aşağıdaki formatta yazılabilir. Kısıtlar normalize edilerek yazılmıştır. Burada $x_1 = d$ ve $x_2 = h$ olarak yazılmıştır.

$f = 2 \pi t d \sqrt{x_2^2 + b^2}$ minimum yapan koşulları sağlamak şartı ile

$$\frac{P}{2 \sigma_y} * \frac{\sqrt{b^2 + x_2^2}}{x_2} * \frac{1}{\pi x_1 t} - 1 \leq 0 \quad (2.16)$$

$$\frac{4P}{x_2} * \frac{(x_2^2 + b^2)^{1.5}}{\pi^2 E (x_1^3 t + x_1 t^3)} - 1 \leq 0 \quad (2.17)$$

kısıtları verilmiştir. Bu soruyu eğer ABC algoritması ile çözülecek ise bu problemin ilk önce d ve h değişkenlerinin sınırları belirlenmelidir. Sınırlar $10 \leq h \leq 100$ in (254 cm) ve $1 \leq d \leq 10$ in (25,4 cm) arasında seçilirse;

1. Arı sayısı (N) ve maksimum iterasyon sayısı (MNC) seçilir. Bu problem için $N=10$ ve $MNC=50$ olarak seçilmiştir.
2. Denklem 2.1 kullanılarak ilk tasarım değişkenlerinin değeri ve bu tasarım değişkenleri kullanılarak amaç fonksiyonu ve kısıt değerleri hesaplanır. Bu problemi kolaylaştırmak için $r=1$ olarak alınarak problem statik ceza katsayısı ile çözülmüştür.

Örnek 2.6. için aşağıda verilen Çizelgelerde amaç fonksiyonu olan $F(x)$ 'in birimi in^3 tür.

Çizelge 2.7. Kafes elemanın ilk tasarım değişkenlerinin değeri ve bu tasarım değişkenleri kullanılarak amaç fonksiyonunun değerleri

Arı No	x_1 için λ	$1 < X_1 < 10'$	x_2 için λ	$10 < X_2 < 100'$	f (amaç fonksiyonu)	g_1	g_2	$F(x)$
1	0.194	2.745	0.380	44.236	471.457	0.593	1.522	1468.503
2	0.198	2.781	0.695	72.505	539.558	0.083	1.133	1196.021
3	0.097	1.869	0.206	28.538	305.233	2.449	9.559	3970.292
4	0.589	6.300	0.881	89.256	1326.372	-0.578	-0.808	1326.372
5	0.985	9.868	0.943	94.881	2136.669	-0.740	-0.949	2136.669
6	0.560	6.038	0.365	42.850	1031.832	-0.256	-0.757	1031.832
7	0.114	2.029	0.462	51.618	358.609	0.900	4.797	2401.653
8	0.152	2.370	0.168	25.120	383.912	2.062	4.766	3005.125
9	0.536	5.825	0.906	91.529	1240.439	-0.550	-0.755	1240.439
10	0.157	2.410	0.134	22.077	387.678	2.404	5.118	3303.834

3. Kısıtsız amaç fonksiyonunun ($F(x)$) en iyilerinden 5 âdeti alınır ve arılar istihdam edilen arılar olur ve bu arıların bulunduğu yerler ise çözümün aranacağı yer olur. Bu problem için 6, 2, 9, 4 ve 1 numaralı arılar istihdam edilen arılar olarak seçilir.

Çizelge 2.8. Kafes eleman için istihdam edilen en iyi ilk beş arı

Arı No	x_1 için λ	$1 < X_1 < 10'$	x_2 için λ	$10 < X_2 < 100'$	f (amaç fonksiyonu)	g_1	g_2	$F(x)$
6	0.5598	6.0379	0.3650	42.8503	1031.8320	-0.2564	-0.7575	1031.8320
2	0.1979	2.7809	0.6945	72.5052	539.5580	0.0833	1.1333	1196.0210
9	0.5361	5.8252	0.9059	91.5287	1240.4390	-0.5504	-0.7554	1240.4390
4	0.5888	6.2996	0.8806	89.2559	1326.3720	-0.5784	-0.8083	1326.3720
1	0.1939	2.7448	0.3804	44.2361	471.4566	0.5926	1.5223	1468.5030

4. Her bir besin kaynağına gidecek izleyici arıların sayısı Denklem 2.3 kullanılarak aşağıdaki gibi hesaplanır.

$$p_6 = \frac{\frac{1}{1031.832}}{\frac{1}{1031.832} + \frac{1}{1196.021} + \frac{1}{1240.439} + \frac{1}{1326.372} + \frac{1}{1468.503}} = 0,2395 \text{ ise izleyici arı sayısı } 1,197 \approx 1$$

$$p_2 = \frac{\frac{1}{1196.021}}{\frac{1}{1031.832} + \frac{1}{1196.021} + \frac{1}{1240.439} + \frac{1}{1326.372} + \frac{1}{1468.503}} = 0,2066 \text{ ise izleyici arı sayısı } 1,033 \approx 1$$

$$p_9 = \frac{\frac{1}{1240.439}}{\frac{1}{1031.832} + \frac{1}{1196.021} + \frac{1}{1240.439} + \frac{1}{1326.372} + \frac{1}{1468.503}} = 0,1992 \text{ ise izleyici arı sayısı } 0,8996 \approx 1$$

$$p_4 = \frac{\frac{1}{1326.372}}{\frac{1}{1031.832} + \frac{1}{1196.021} + \frac{1}{1240.439} + \frac{1}{1326.372} + \frac{1}{1468.503}} = 0,168 \text{ ise izleyici arı sayısı } 0,9316 \approx 1$$

$$p_1 = \frac{\frac{1}{1468.503}}{\frac{1}{1031.832} + \frac{1}{1196.021} + \frac{1}{1240.439} + \frac{1}{1326.372} + \frac{1}{1468.503}} = 0,0995 \text{ ise izleyici arı sayısı } 0,8415 \approx 1$$

Buna göre 6, 2, 9, 4, 1 besin kaynaklarına 1 er izleyici arı ve birer adette istihdam edilen arı sayısı gidecektir.

5. Sırası ile 6, 2, 9, 4 ve 1 numaralı besin kaynaklarına bir adet istihdam edilen arı ve yukarıda hesaplandığı kadar izleyici arı gönderilir. Örnek olarak 6 numaralı besin kaynağının ilk değerleri bir önceki tabloda verilmiştir. İstihdam edilen arı ve izleyici arı için x_1 ve x_2 değişken değerleri Denklem 2.4 kullanılarak aşağıdaki tabloda hesaplanmıştır. Örnek olarak 6 numaralı besin kaynağına giden istihdam arı için

$$yeni_6 x_1 = 6,038 + 0,156 * (6,038 - 2,781) = 6,545$$

$$yeni_6 x_2 = 42,85 - 0,636 * (42,850 - 44,236) = 43,732$$

yeni tasarım değişkenleri hesaplanmış olur. Amaç fonksiyonlu da aşağıdaki gibi hesaplanır.

$$f_{yeni} = 2 * \pi * 0,25 * x_1 * \sqrt{x_2^2 + 100^2} = 1122.0979$$

Daha sonra $g_1(x_1, x_2)$ ve $g_2(x_1, x_2)$ fonksiyonları hesaplanır.

$$g_1(x_1, x_2) = \frac{100}{2*36} * \frac{\sqrt{100^2 + 43,732^2}}{43.732} * \frac{1}{\pi 6.545 * 0.25} - 1 = -0,326$$

$$g_2(x_1, x_2) = \frac{400}{43,73} * \frac{(43.73^2 + 100^2)^{1.5}}{\pi^2 * 29000 * 6,545 * 0,25 * (6,545^2 + 0,25^2)} - 1 = -0,812$$

Son işlem olarak sınırsız amaç fonksiyonun değeri hesaplanmalıdır. Burada kısıtların her ikisi de negatif olmasını anlamı kabul edilebilir sınırlar içerisinde olduğunu gösterir. Dolayısı ile $f(x) = F(x)$ olur. Yapılan işlemler izleyici arı içinde yapılarak $F(x) = 1122,143$ bulunur. İşlemlere başlamadan önce 6 numaralı besin için $F(x) = 1031,832$ bulunmuştur. Şimdi bulunan $F(x)$ 'ler küçük olduğu için önceki adımdaki değer yine en iyi çözüm olarak kabul edilir. Aynı işlem diğer besin kaynakları içinde yapılarak en iyi çözümler kendi aralarında seçilerek 2. adıma geçilir.

Çizelge 2.9. Kafes eleman için istihdam edilen arılar ve varsa izleyici arılar için amaç fonksiyonu

Arı No	x1			x2			f(x)	g1	g2	F(x)
	k	ϕ	$1 < X1 < 10'$	k	ϕ	$10 < X2 < 100'$				
6	Önceki Adım		6.038			42.850	1031.832	-0.256	-0.757	1031.832
	İstihdam	2	0.156	1	-0.636	43.732	1122.143	-0.326	-0.812	1122.143
	İzleyici 1	2	0.022	4	-0.244	54.186	1091.776	-0.393	-0.789	1091.776
2	Önceki Adım		2.781			72.505	539.558	0.083	1.133	1196.021
	İstihdam	6	-0.793	1	-0.760	51.018	945.713	-0.274	-0.681	945.713
	İzleyici 1	9	0.892	6	-0.480	58.259	181.793	2.513	43.561	8557.640
9	Önceki Adım		5.825			91.529	1240.439	-0.550	-0.755	1240.439
	İstihdam	4	-0.579	6	-0.602	62.203	1128.379	-0.451	-0.794	1128.379
	İzleyici 1	2	0.549	1	0.020	92.456	1300.424	-0.571	-0.784	1300.424
4	Önceki Adım		6.300			89.256	1326.372	-0.578	-0.808	1326.372
	İstihdam	6	0.927	6	-0.021	88.299	1370.908	-0.592	-0.829	1370.908
	İzleyici 1	1	0.232	6	-0.281	76.221	1407.060	-0.591	-0.872	1407.060
1	Önceki Adım		2.745			44.236	471.457	0.593	1.522	1468.503
	İstihdam	9	-0.898	6	-0.591	43.417	986.006	-0.229	-0.722	986.006
	İzleyici 1	4	0.844	9	-0.050	45.126	172.333	3.299	48.007	9014.112

6. İkinci adımda yapılan işlemler aynı birinci adımda olduğu gibidir.

Çizelge 2.10. Kafes eleman için yeni bir yenileme için istihdam edilen arılar ve varsa izci arılar için amaç fonksiyonu

Arı No	x1			x2			f(x)	g1	g2	F(x)
	k	ϕ	$1 < X1 < 10'$	k	ϕ	$10 < X2 < 100'$				
6	Önceki Adım		6.038			42.850	1031.832	-0.256	-0.757	1031.832
	İstihdam	1	-0.963	2	-0.721	48.739	1007.980	-0.300	-0.739	1007.980
	İzleyici 1	2	-0.205	2	0.847	35.932	984.714	-0.114	-0.711	984.714
2	Önceki Adım		5.363			51.018	945.713	-0.274	-0.681	945.713
	İstihdam	1	-0.872	9	-0.609	57.831	1035.608	-0.381	-0.745	1035.608
	İzleyici 1	9	0.780	6	0.063	51.967	847.633	-0.199	-0.555	847.633
9	Önceki Adım		6.100			62.203	1128.379	-0.451	-0.794	1128.379
	İstihdam	2	-0.663	6	0.655	79.421	1049.105	-0.456	-0.675	1049.105
	İzleyici 1	4	0.338	4	-0.085	64.509	1127.581	-0.459	-0.789	1127.581
4	Önceki Adım		6.542			88.299	1370.908	-0.592	-0.829	1370.908
	İstihdam	1	-0.734	1	0.183	97.630	1295.657	-0.571	-0.758	1295.657
	İzleyici 1	2	-0.364	2	0.031	90.409	1217.558	-0.541	-0.747	1217.558
1	Önceki Adım		5.758			43.417	986.006	-0.229	-0.722	986.006
	İstihdam	6	0.514	4	0.832	10.000	897.456	2.126	-0.019	2805.556
	İzleyici 1	6	-0.320	4	0.263	31.064	954.542	0.027	-0.664	980.511

7. 50. adımda gelindiğinde aşağıdaki çözümler bulunur.

Çizelge 2.11. Kafes eleman için 50. yenileme sonrası en son amaç fonksiyonu

Arı No	x1 için ϕ	$1 < X1 < 10'$	x2 için ϕ	$10 < X2 < 100'$	f (amaç fonksiyonu)	g1	g2	F(x)
6	0.1086	3.6252	-0.3818	55.8858	652.3291	-0.0001	0.0000	652.3291
2	-0.9918	3.6252	-0.1299	55.8762	652.3031	0.0001	0.0000	652.3595
1	-0.4647	3.6255	-0.6640	55.8803	652.3704	-0.0001	-0.0003	652.3704
4	0.3718	3.6263	-0.5041	55.8627	652.4759	-0.0001	-0.0009	652.4759
9	-0.0230	3.6267	0.2567	55.8539	652.5158	-0.0001	-0.0011	652.5158

3. SAYISAL ÖRNEKLER

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

1. Tek yükleme tipine maruz kalan on elemanlı kafes yapısı,
2. Tek yükleme tipine maruz kalan tek açıklıklı, 10 katlı ve 30 elemandan oluşan düzlem çelik çerçeve yapısı,
3. Tek yükleme tipine maruz kalan üç açıklıklı, 24 katlı ve 168 elemandan oluşan düzlem çelik çerçeve yapısı,
4. Dört yükleme tipine maruz kalan tek açıklıklı, 4 katlı ve 32 elemandan oluşan üç boyutlu uzay çelik çerçeve yapısı.

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

C# programı nesne tabanlı olduğu için geliştirilen program üç ana sınıfa (class) bölünmüştür. Bu sınıflar *Data*, *Bee* ve *Program* dır. *Data* sınıfı tasarım örneği için gerekli sınır şartlarını, malzeme özelliklerini ve tasarım değişkenlerinin özelliklerini içermektedir. Bu sınıfta yapı ile ilgili bilgiler mevcuttur. *Bee* sınıfı ise arının davranışını modellemek için geliştirilmiştir. Her bir arının adı, tasarım değişkenlerini ve arılar arasındaki yiyecek arama gibi davranışları modellemek için geliştirilmiştir. Eğer arının besin bulduğu yer uygun bölgede değil ise yine bu sınıf tarafından ceza fonksiyonu ile birlikte amaç fonksiyonları da yine burada hesaplanır. *Program* sınıfı ise Sap2000 programını çağırarak verilen kesit ve geometriye ait yapının göstereceği yer değiştirmeleri ve gerilmeleri bulmak için kullanılmıştır. Bu üç sınıf için sabitleri ve metotların adları Ek 1, Ek 2 ve Ek 3' de verilmiştir.

Geliştirilen optimizasyon programı Microsoft Windows 7 Professional 32 Bit işletim sistemli, Intel Core i5-2400 CPU 3.10 GHz işlemci gücüne sahip ve 4096 MB belleği olan bir kişisel bilgisayarda çalıştırılmıştır.

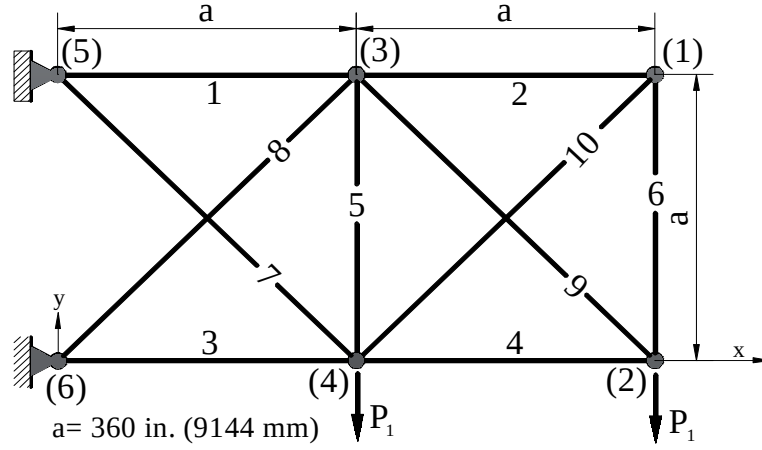
Optimizasyon programı çalıştırılmadan önce arı sayısı, Maksimum döngü ve Limit sayıları önceden tespit edilmelidir. Bu çalışmada sunulan bütün örneklerde;

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

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

3.1. On Elemanlı Çelik Kafes Sistemi

İlk tasarım örneği çeşitli tasarım yaklaşımları ve nümerik teknikleri karşılaştırmak için literatürde birçok kez kullanılan on elemanlı kafes sistemidir.



Şekil 3.1. On elemanlı kafes sistemin geometrisi ve yükleme şekli

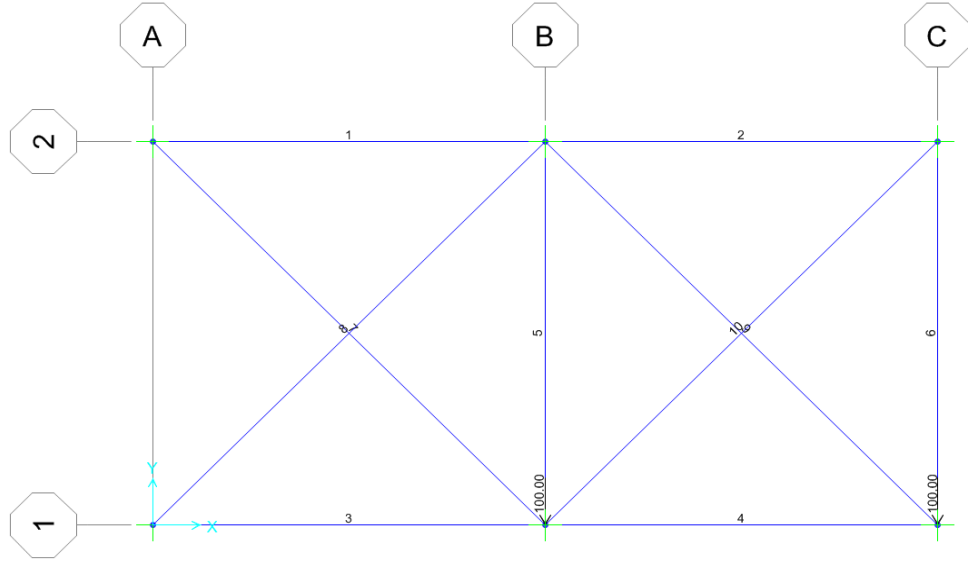
Bu kafes sistem ayrık tasarım değişkenli olarak genetik algoritma (GA) kullanılarak Rajeev ve Krishnamoorthy (1992), tavlama benzetimi (SA) kullanılarak Kripka (2004), hiyerarşi parçacık sürü optimizasyon algoritması (HPSO) kullanılarak Li vd. (2009), karınca koloni algoritması (ACO) kullanılarak Camp ve Bichon (2004), yapay arı kolonisi algoritması (ABC) kullanılarak Sonmez (2011b) tarafından çalışılmıştır.

Şekil 3.1.'de on eleman içeren kafes sistemin geometrisi ve yükleme şekli gösterilmiştir. Bu örnek literatürde birçok araştırmacı tarafından kullanılmıştır. $P_1 = 100$ kips (444,82 kN) olarak verilmiştir. Her bir kesit elemanı için Elastisite Modülü $E = 10000$ ksi (68,971 GPa) ve malzeme yoğunluğu $\rho = 0,10$ lb/in.³ (2768 kg/m³) olduğu varsayılmıştır. Her bir eleman için muhtemel en kesit alanlar $L = \{1.62, 1.80, 1.99, 2.13, 2.38, 2.62, 2.63, 2.88, 2.93, 3.09, 3.13, 3.38, 3.47, 3.55, 3.63, 3.84, 3.87, 3.88, 4.18, 4.22, 4.49, 4.59, 4.80, 4.97, 5.12, 5.74, 7.22, 7.97, 11.5, 13.5, 13.9, 14.2, 15.5, 16.0, 16.9, 18.8, 19.9, 22.0, 22.9, 26.5, 30.0, 33.5\}$ (in.²) olmak üzere 42 tasarım değişken kullanılmıştır. Her iki yönde olmak üzere her bir düğüm noktasının yer değiştirmesi ± 2 in. (± 50.8 mm) değerinden az olmalıdır. Sistem için müsaade edilen gerilme değeri ise ± 25 ksi (± 172.25 Mpa) olarak verilmiştir. Gerilme kısıtı ve yer değiştirme kısıtı aşağıdaki gibi ifade edilmiştir.

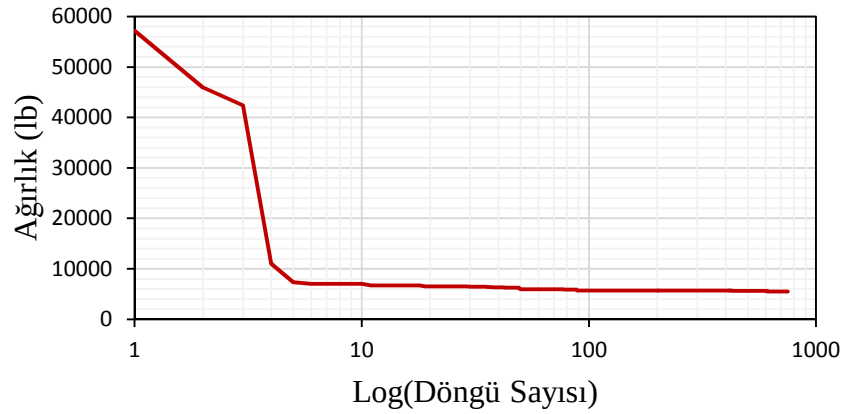
$$\sigma_{m,l} = \frac{\sigma_{m,l}}{\sigma_{m,müsaade}} - 1 \leq 0, \quad m = 1, 2, \dots, n \quad (3.1)$$

$$U_{k,l} = \frac{u_{k,l}}{u_{k,müsaade}} - 1 \leq 0, \quad k = 1, 2, \dots, n_n \quad (3.2)$$

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



Şekil 3.2. On elemanlı kafes sisteminin geometrisi ve yüklemesinin Sap2000 görünümü



Şekil 3.3. On elemanlı kafes sistemi için ağırlık ve döngü sayısı ilişkisi

Çizelge 3.1. de gösterildiği gibi Sap2000 ile yapay arı kolonisi algoritması (ABC) birlikte geliştirilerek oluşturulan program ile en iyi tasarım geliştirilmiştir. Geliştirilen tasarım ile on elemanlı kafes sistemin ağırlığı 5476,37 lb bulunmuştur. Bu sonuç Rajeev ve Krishnamoorthy (1992), Kripta (2004), Li vd. (2009), Camp ve Bichon (2004), Sonmez (2011b) çalışmalarından elde edilen ağırlıklardan daha hafiftir.

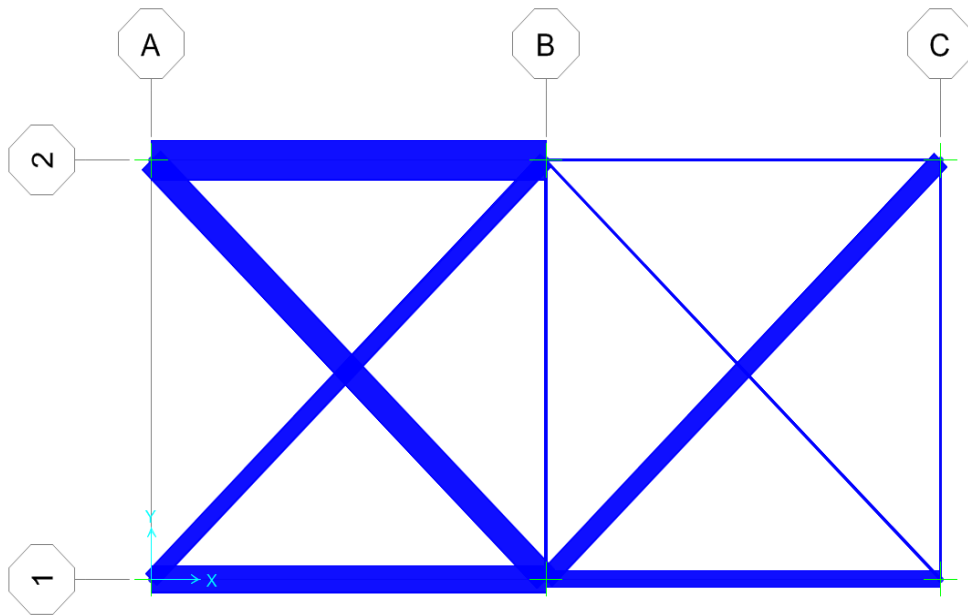
Optimum tasarım sonuçları elde edildikten sonra minimum ağırlığı veren tasarım için ağırlık-döngü sayısı ilişkisi Şekil 3.3.'de verilmiştir. Döngü sayısının artması ile birlikte amaç fonksiyonumuz yani verilen kısıtlara göre minimum ağırlığı bulma işlemi bu şekilde gösterilmiştir. Yatay eksen logaritmik olarak verilmiştir.

Çizelge 3.1. On elemanlı kafes sisteminin optimum sonuçları

Değişkenler		Optimum en kesit alanları (in^2)					
Numara	Değişken	GA	SA	HPSO	ACO	ABC	Bu Çalışma
1	A ₁	33,50	33,50	30,00	33,50	33,50	33,50
2	A ₂	1,62	1,62	1,62	1,62	1,62	1,62
3	A ₃	22,00	22,90	22,90	22,90	22,90	22,90
4	A ₄	15,50	14,20	13,50	14,20	14,20	13,90
5	A ₅	1,62	1,62	1,62	1,62	1,62	1,62
6	A ₆	1,62	1,62	1,62	1,62	1,62	1,62
7	A ₇	14,20	7,97	7,97	7,97	7,97	13,90
8	A ₈	19,90	22,90	26,50	22,90	22,90	22,90
9	A ₉	19,90	22,00	22,00	22,00	22,00	16,00
10	A ₁₀	2,62	1,62	1,80	1,62	1,62	1,62
En iyisi (lb)		5613,84	5490,74	5531,98	5490,74	5490,74	5476,37
Ortalaması (lb)		*	*	*	*	5510,35	5543,66
En kötüsü (lb)		*	*	*	*	5536,97	5711,65
Analiz sayısı (#)		*	*	50000	10000	25800	35000
Kısıt ihlali		Yok	Yok	Yok	Yok	Yok	Yok

*Değerler Önceki çalışmalarda verilmemiştir.

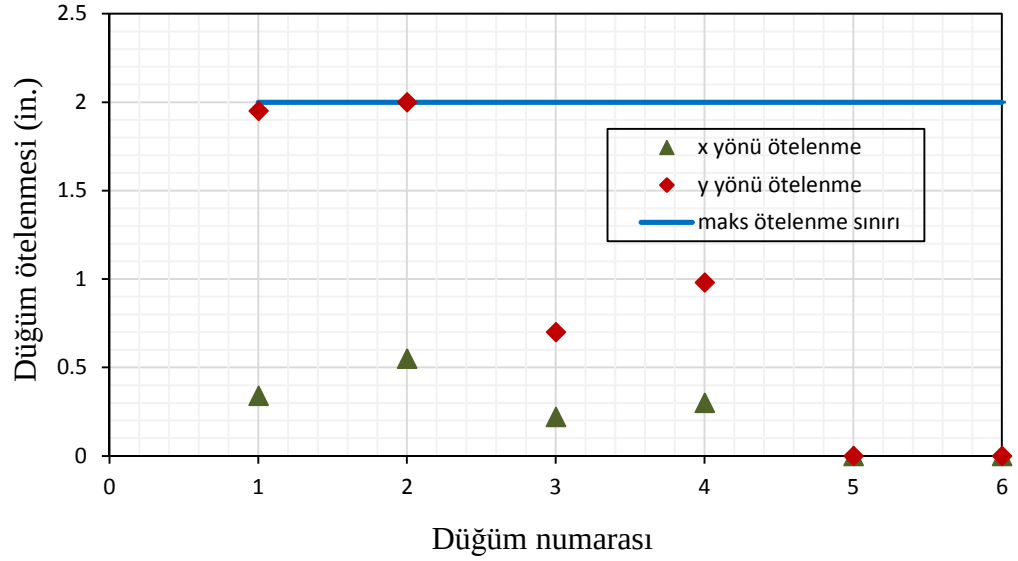
Çizelge 3.1.'de program bağımsız olarak 5 kere çalıştırılması sonucunda elde edilen en iyi ağırlık, ortalama ağırlık, en kötü ağırlık, toplam analiz sayısı ve verilen kısıtların ihlalleri verilmiştir.



Şekil 3.4. On elemanlı kafes sistemi için optimum tasarım şekli

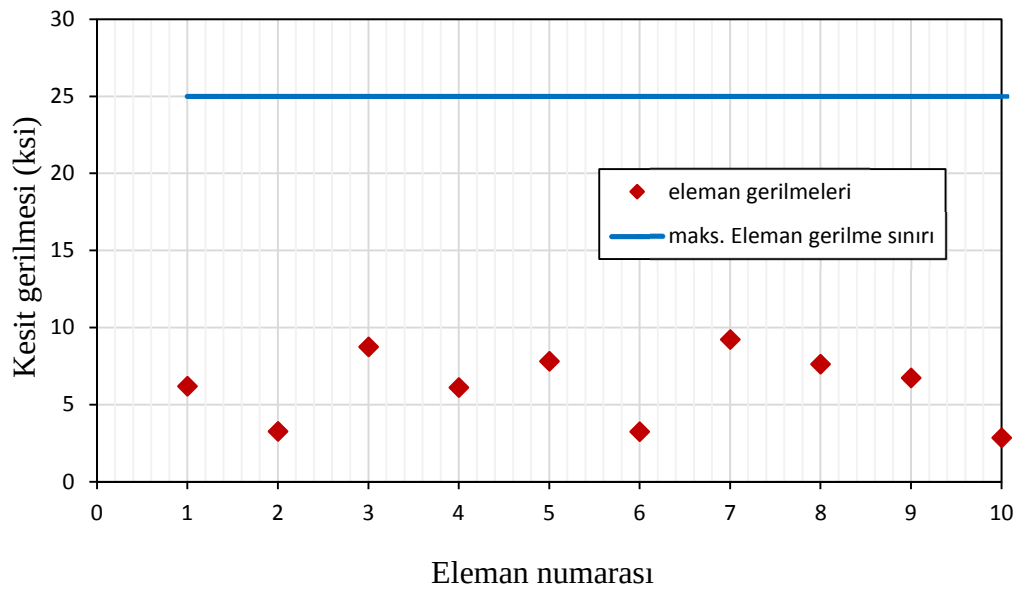
Elde edilen en iyi ağırlık için her bir elemana ait en kesit alanları Sap2000 programında her elemana ayrı ayrı atanarak Şekil 3.4.' de gösterilmiştir. On elemanlı kafes sisteminin tasarım optimizasyonu için gerekli olan deplasmanlar, gerilmeler Sap2000 programı ile elde edilmiştir.

Şekil 3.5.'de on elemanlı kafes sisteminde düğüm noktalarının x ve y doğrultusundaki yer değiştirilmeleri mutlak değerce gösterilmiştir. Yer değiştirmelerin hepsi müsaade edilen ± 2 in (50.8 mm) değerinden küçük veya eşit olduğu gözlenmiştir.



Şekil 3.5. On elemanlı kafes sistemi için x ve y doğrultusundaki yer değiştirmeler

Şekil 3.6.' da kafes için her bir elemandaki gerilme değerleri verilmiştir. Gerilme değerlerinin tamamı müsaade edilen ± 25 ksi değerinden düşük olduğu gözlemlenmiştir.



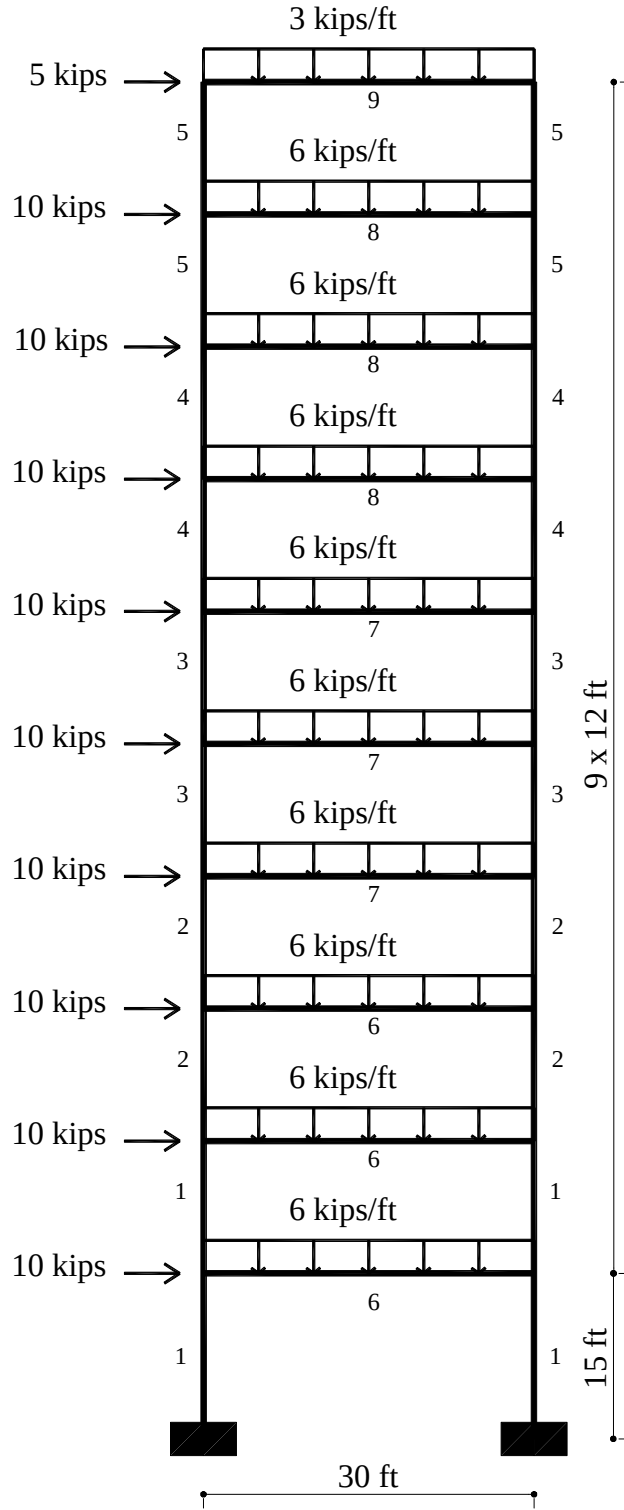
Şekil 3.6. On elemanlı kafes sistemi için eleman gerilmeleri

3.2. Tek Açıklıklı On Katlı Düzlem Çelik Çerçeve

Şekil 3.7.'de tek açıklıklı on katlı düzlem çelik çerçevenin boyutları, yükleme şekli, elemanların grup numaraları gösterilmiştir. Bu çelik çerçeve daha önceden Pezeshk vd., (2000) yılında genetik algoritma (GA) ile ve Değertekin (2012) ise yapay arı kolonisi algoritması (ABC) ile kullanılarak optimum tasarımı yapılmıştır.

Gerilme kıstasları Amerikan Çelik Yapılar Enstitüsü -Yük ve Dayanım Faktörü Tasarımı (AISC-LRFD, 2001) şartnamesinden alınmıştır. Kiriş elemanları LRFD' de bulunan ve Ek.4'de verilen 267 adet W kesitlerinden seçilmiştir, kolon elemanları ise sadece W14 ve W12 kesitlerinden seçilmiştir.

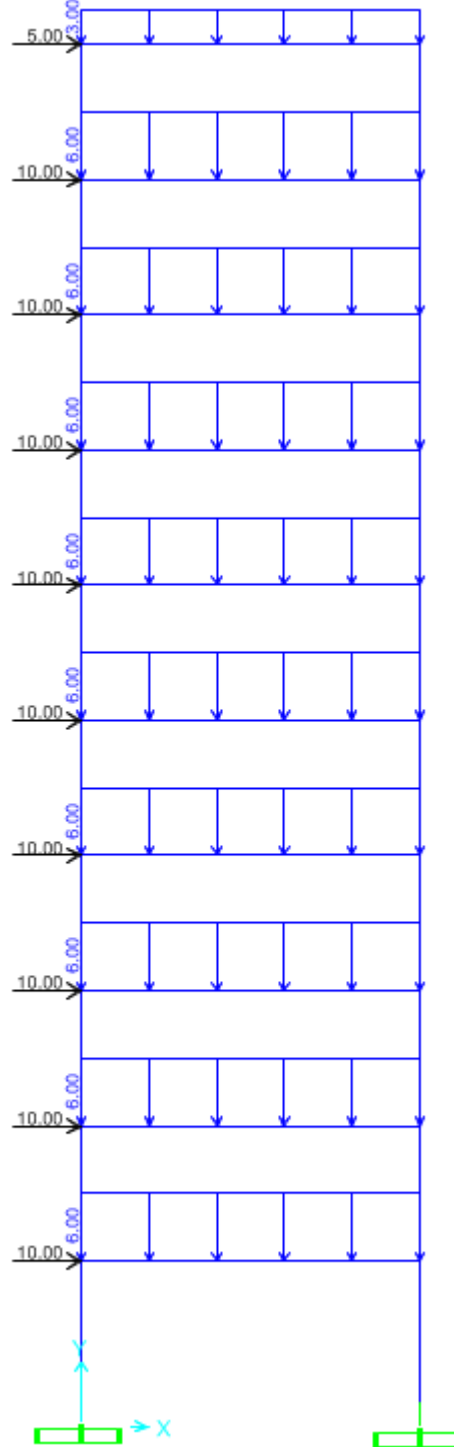
Sistem 22 düğüm noktası ve 30 elemandan oluşturulmuş olup, kolonlar 5 ayrı gruba ayrılmıştır. İlk iki kattaki kolonlar birinci grup, üç ve dördüncü kattaki kolonlar ikinci grup, beş ve altıncı kattaki kolonlar üçüncü grup, yedi ve sekizinci kattaki kolonlar dördüncü grup, dokuz ve onuncu kattaki kolonlar ise beşinci grup olarak alınmıştır. Kirişler ise üç ayrı gruptan oluşmuştur. Bir, iki ve üçüncü kattaki kirişler altıncı grup; dört, beş ve altıncı kattaki kirişler yedinci grup; yedi, sekiz ve dokuzuncu kattaki kirişler sekizinci grup ve son kattaki kirişler ise dokuzuncu ve son grubu oluşturmaktadır. Şekil 3.8.'de çerçeve sistemin Sap2000 programı ile yüklenmesi gösterilmiştir.



Şekil 3.7. Tek açıklıklı on katlı çerçeve sisteminin boyutlandırılması, yükleme şekli ve elemanların gruplandırılması

Sistem tek açıklıktan oluşmaktadır ve açıklık 30 ft (9.144 m) uzunluğundadır. Birinci kat yüksekliği 15 ft (4.572 m), diğer kat yükseklikleri ise 12 ft (3.657 m) uzunluğundadır. Sistem tek tip yükleme sahiptir, x doğrultusunda düğüm noktalarına

ilk dokuz kat için 10 kips (44.48 kN) ve son kat için 5 kips (22.24 kN) yatay yük etki etmiştir. Kirişlerin üzerinde ise ilk dokuz kat için 6 kips/ft (87.56 kN/m) son kat için 3 kips/ft (43.78 kN/m) düzgün yayılı yük etki ettirilmiştir. Mesnetlenme şekli olarak ankastre mesnet kullanılmıştır.



Şekil 3.8. Tek açıklıklı on katlı çerçeve sisteminin Sap2000 ile yüklenmesi

Sistemde maksimum yer deęiřtirme ve katlar arası yer deęiřtirme Denklem 3.3 ve 3.4 deki kısıtlara göre yapılmıřtır.

$$\beta_{ij}(I) = \frac{\delta_{jl}}{\delta_{ju}} - 1, \quad j = 1, \dots, m, l = 1, \dots, nl \quad (3.3)$$

$$\beta_{ij}(Z) = \frac{\Delta_{jil}}{\Delta_{ju}} - 1, \quad j = 1, \dots, ns, i = 1, \dots, nsc, l = 1, \dots, nl \quad (3.4)$$

Burada δ_{jl} yükleme řartından dolayı meydana gelen yer deęiřtirme, δ_{ju} ise maksimum yer deęiřtirme üst sınırıdır. Δ_{jil} ise yükleme durumundan dolayı meydana gelen i inci katta j inci kolondaki katlar arası yer deęiřtirmedir, Δ_{ju} ise katlar arası yer deęiřtirme sınırıdır. Katlar arası yer deęiřtirme kıstası ise $\delta_i < h_i/300$ řeklinde alınmıřtır (δ_i i inci kat yer deęiřtirmesi, h_i i inci kat yükseklięi).

Tek açıklıklı on katlı sistem için uyulması gereken boyut kıstaları Denklem 3.5 ve 3.6’da verilmiřtir.

$$\beta_k(I) = \frac{b_{fbk}}{b_{fck}} - 1, \quad k = 1, \dots, nj \quad (3.5)$$

$$\beta_n(I) = \frac{d_{un}}{d_{bn}} - 1, \quad n = 1, \dots, ncl \quad (3.6)$$

Burada b_{fbk} ve b_{fck} sırasıyla seęilen kiriř ve kolonların kenar genişlikleridir. d_{un} ve d_{bn} üst ve alt kattaki kolonlar için seęilen çelik kesitlerin derinlikleridir.

Sistem için Denklem 3.7 ve 3.8 de verilen gerilme kıstaları Amerikan Çelik Yapılar Enstitüsü - Yük ve Dayanım Faktörü Tasarımı (AISC-LRFD, 2001) řartnamesinden alınmıřtır. Her eleman eęilme momenti ve eksenel kuvvete maruz olduęu kabulü ile işlemler yapılmıřtır.

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

$$\beta_{il}(I) = \left(\frac{P_u}{\phi P_n} \right)_{il} + \frac{8}{9} \left(\frac{M_{ux}}{\phi_b M_{nx}} + \frac{M_{uy}}{\phi_b M_{ny}} \right)_{il} - 1.0 \quad i = 1, \dots, nm, l = 1, \dots, nl \quad (3.7)$$

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

$$\beta_{il}(I) = \left(\frac{P_u}{2\phi P_n} \right)_{il} + \left(\frac{M_{ux}}{\phi_b M_{nx}} + \frac{M_{uy}}{\phi_b M_{ny}} \right)_{il} - 1.0 \quad i = 1, \dots, nm, l = 1, \dots, nl \quad (3.8)$$

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

Eğer Denklem 3.7 ve 3.8 deki aksel kuvvet basınç ise, P_n taşınabilecek maksimum kuvvet, $\phi = \phi_c$ basınç için dayanım faktörüdür ve 0.85 değerindedir. Elemanların taşıyacağı maksimum kuvvet aşağıda Denklem 3.9' da verilmiştir.

$$P_n = A_g \times F_{cr} \quad (3.9)$$

$$F_{cr} = (0.658^{\lambda_c^2}) F_y \quad \lambda_c \leq 1.5 \text{ ise} \quad (3.10)$$

$$F_{cr} = \left(\frac{0.877}{\lambda_c^2} \right) F_y \quad \lambda_c > 1.5 \text{ ise} \quad (3.11)$$

$$\lambda_c = \frac{KL}{r\pi} \sqrt{\frac{F_y}{E}} \quad (3.12)$$

Burada A_g elemanın en kesit alanı, K etkili uzunluk faktörü (narinliğe bağlı katsayı), E eleman malzemesinin Elastisite Modülü; r atalet yarıçapı, L eleman uzunluğu, F_y çeliğin akma gerilmesidir. Etkili uzunluk faktörü (K) yanal ötelenmesi önlenmemiş sistemler için Denklem 3.13 ile hesaplanabilir.

$$K = \sqrt{\frac{1.6G_A G_B + 4.0(G_A + G_B) + 7.50}{G_A + G_B + 7.50}} \quad (3.13)$$

G_A ve G_B sınırlama faktörü Denklem 3.14 ile hesaplanabilir.

$$G = \frac{\sum(I_c/L_c)}{\sum(I_g/L_g)} \quad (3.14)$$

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

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

$$M_u = \beta_1 M_{nt} + \beta_2 M_{lt} \quad (3.15)$$

$$\beta_1 = \frac{C_m}{(1 - P_u/P_{e1})} \geq 1.0 \quad (3.16)$$

$$\beta_2 = \frac{1}{1 - \frac{\sum P_u}{\sum H} \left(\frac{\Delta_{oh}}{L} \right)} \geq 1.0 \quad (3.17)$$

M_{nt} = yanal deplasmanların önlendiği çerçevelerdeki momentler,

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

P_u = eleman aksel kuvvetidir,

C_m = yanal ötelenme durumuna göre kullanılan katsayı,

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

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

Δ_{oh} = belirtilen yükler altında kattaki yer değiştirmedir,

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

K = etkili uzunluk faktörüdür.

Denklem 3.15'deki M_{nt} ve M_{lt} 'yi hesaplamak için ilk önce yanal deplasmanları durdurmak için mesnetler yerleştirilmelidir. Daha sonra bu mesnetler meydana gelen kuvvetler ve yanal kuvvetler birlikte etki ettirilerek M_{lt} 'ler hesaplanmalıdır. Bu da işlemleri çok artıracığı için TS500'de verilen başka bir alternatif yöntem kullanılmıştır. Bu yönetime göre M_u Denklem 3.18'de ifade edilmiştir.

$$M_u = \beta_1 \times \beta_2 \times M \quad (3.18)$$

Denklem 3.18'de M ise her iki yön için ayrı ayrı gayri müsait yüklemelerden meydana gelen en büyük kolon momenti alınarak kullanılmıştır.

Şekil 3.9.'da farklı mesnetlenme tiplerine göre etkili uzunluk faktörleri (K) verilmiştir.

Basınç çubukları için burkulma boyları						
Basınç çubuğunun burkulma şekli	(a)	(b)	(c)	(d)	(e)	(f)
Teorik burkulma boyu çarpanı (K)	0.5	0.7	1.0	1.0	2.0	2.0
Tavsiye edilen boyu çarpanı (K)	0.65	0.80	1.2	1.0	2.10	2.0
Mesnet tanımları	 Dönme ve ötelenme tutulu  Dönme serbest ve ötelenme tutulu  Dönme tutulu ve ötelenme serbest  Dönme ve ötelenme serbest					

Şekil 3.9. Etkili uzunluk faktörü (TS 648)

C# programı ile geliştirilen yapay arı koloni algoritması (ABC) literatürde kullanılan bu tasarım örneğine uygulanmıştır. Programın performansını görmek için birbirinden bağımsız olarak beş kere çalıştırılmıştır, optimum tasarım için en iyi ve en kötü sonuçlar, en üst kattaki maksimum yer değiştirme, katlar arası yer değiştirme elde edilmiştir. En iyi tasarım ağırlığı ve buna denk gelen kesitler, ortalama tasarım ağırlığı, maksimum yer değiştirme, katlar arası maksimum yer değiştirme ve standart sapma Çizelge 3.2. 'de verilmiştir.

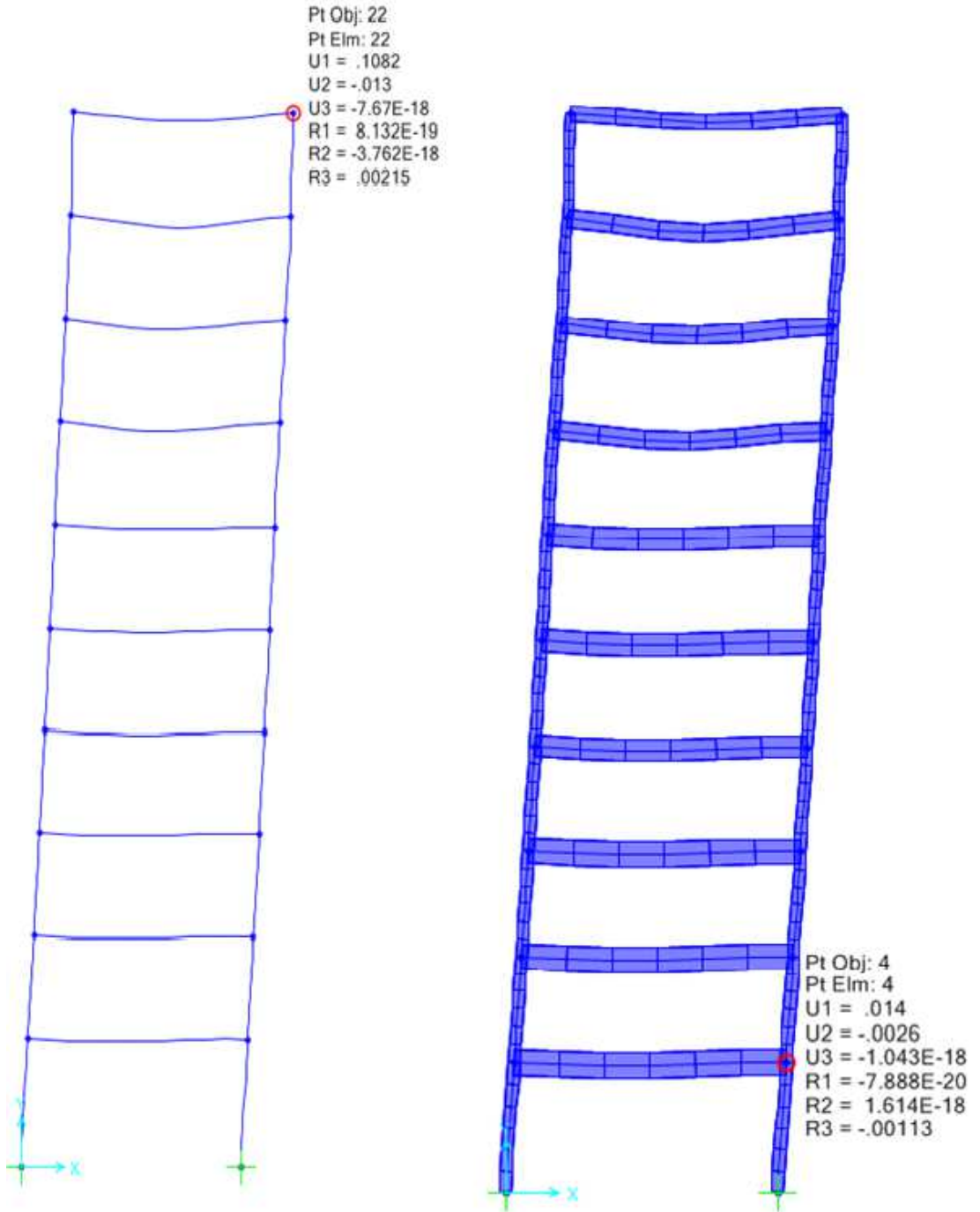
Malzeme için Elastisite Modülü $E = 200\text{GPa}$, malzeme akma gerilmesi $f_y = 248,2\text{ MPa}$ alınmıştır. Maksimum yer değiştirme sınırı $12,495\text{ cm}$ ve katlar arası maksimum yer değiştirme sınırı birinci kat için $1,52\text{ cm}$, diğer katlar için $1,219\text{ cm}$ olarak verilmiştir.

Çizelge 3.2. Tek açıklıklı on katlı çelik çerçeve sisteminin optimum sonuçları

Grup sayısı	GA Pezeshk vd. (2000)	ABC Degertekin (2012)	ABC Bu çalışma
1	W 14 × 233	W 14 × 233	W 14 × 211
2	W 14 × 176	W 14 × 176	W 14 × 193
3	W 14 × 132	W 14 × 145	W 14 × 176
4	W 14 × 99	W 14 × 99	W 14 × 159
5	W 12 × 65	W 12 × 58	W 14 × 61
6	W 36 × 150	W 36 × 150	W 33 × 130
7	W 33 × 130	W 33 × 118	W 30 × 116
8	W 27 × 94	W 27 × 84	W 24 × 55
9	W 16 × 50	W 24 × 68	W 21 × 62
Ağırlık (kN)	313,13	308,68	305,76
En üst kat ötelenme(cm)	*	9,98	10,82
Katlar arası maks. Ötelenme(cm)	*	1,23	1,4
Ortalama ağırlık (kN)	*	314,153	313,134
Standart sapma	*	3,72	4,6

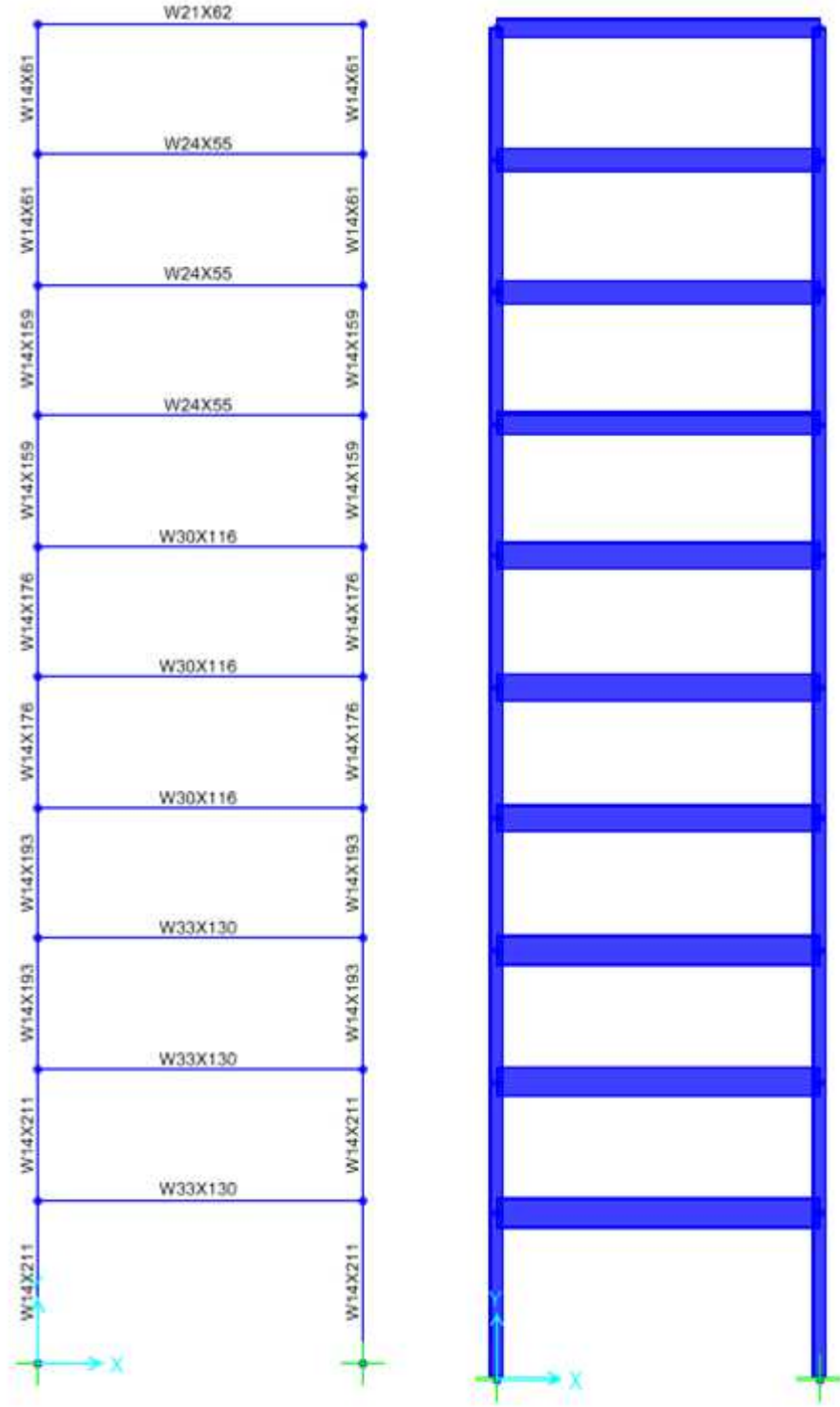
*Değerler önceki çalışmalarda verilmemiştir.

Çizelge 3.2.'de görüldüğü gibi minimum tasarım ağırlığı $305,76\text{ kN}$ bulunmuştur bu değer Pezeshk vd. (2000) ve Degertekin (2012) çalışmalarındaki optimum tasarımdan daha hafif bir tasarım olmuştur. Maksimum yer değiştirme son katta $10,82\text{ cm}$ olarak hesaplanmıştır bu değer maksimum değer olan $12,495\text{ cm}$ den daha küçüktür. Katlar arası maksimum yer değiştirme birinci katta $1,4\text{ cm}$ olarak hesaplanmıştır bu değer birinci kattaki maksimum değer olan $1,52\text{ cm}$ den daha küçük bir değer olarak ölçülmüştür.



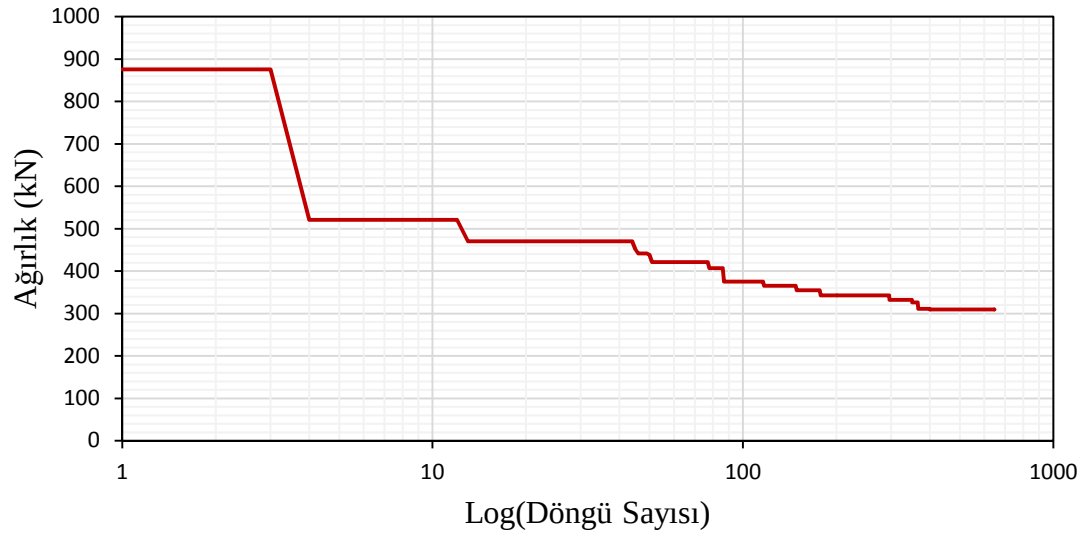
Şekil 3.10. Tek açıklıklı on katlı çerçeve sistemi için maksimum tepe yer değiştirmesi ve maksimum katlar arası yer değiştirmenin yeri (m)

Şekil 3.10.'da en son katta ve katlar arası oluşan maksimum yer değiştirmeler gösterilmiştir. 0,1082 m olarak ölçülen değer x doğrultusunda oluşan maksimum yer değiştirmedir. Diğer doğrultu olan y yönünde ise 0,013 m yer değiştirme ölçülmüştür. En iyi tasarım için oluşan x doğrultusundaki katlar arası maksimum yer değiştirme ise 0,014 m olarak hesaplanmıştır. Şekil 3.13.'de ise katlar arası yer değiştirmelerin, maksimum üst sınır ile ilişkisi verilmiştir.



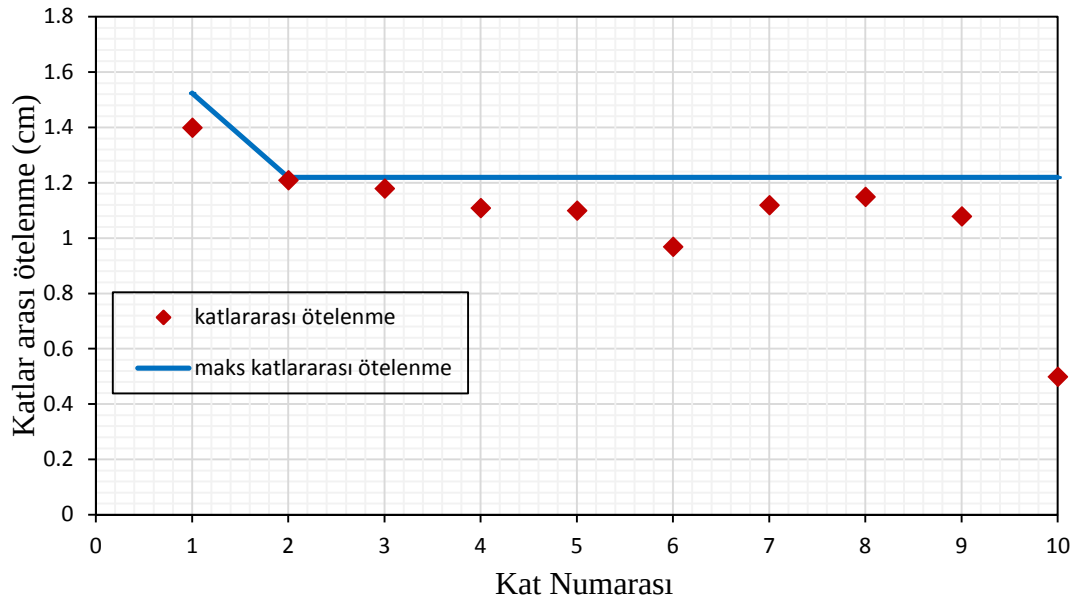
Şekil 3.11. Optimum tasarım sonucu oluşan kesitler ve düzlemsel görünümleri

Optimum tasarım sonucu elde edilen en iyi sonuç için kesitler ve düzlemsel görünümleri Sap2000 programı yardımı ile Şekil 3.11.'de gösterilmiştir.

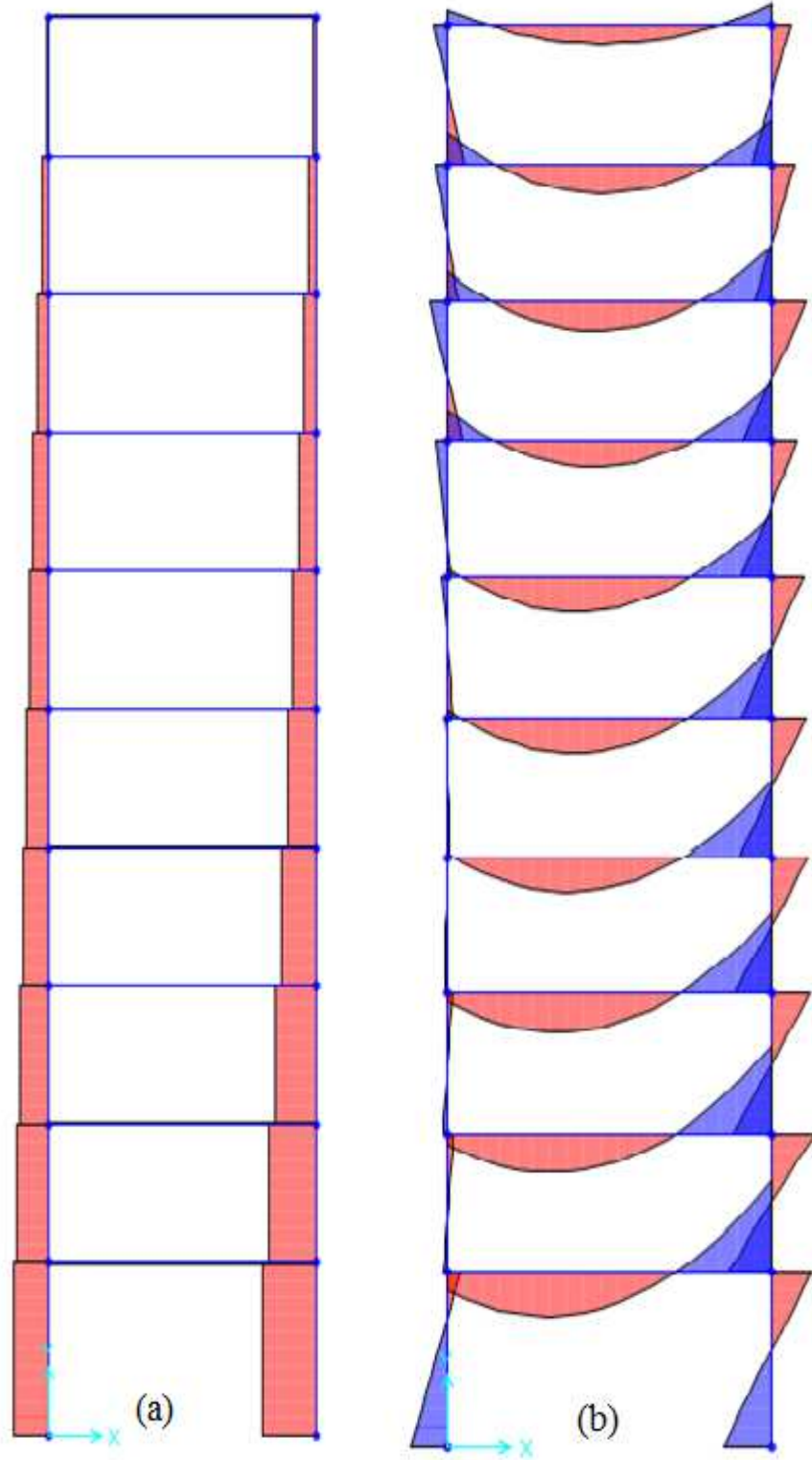


Şekil 3.12. Tek açıklıklı on katlı çerçeve sistemi için ağırlık ve döngü sayısı ilişkisi

Optimum tasarım sonuçları elde edildikten sonra minimum ağırlığı veren en iyi tasarım için ağırlık-döngü sayısı ilişkisi Şekil 3.12.'de verilmiştir. Yatay eksen logaritmik olarak verilmiştir.



Şekil 3.13. Tek açıklıklı on katlı çerçeve sistemi x doğrultusundaki yer değiştirmeler



Şekil 3.14. Tek açıklıklı on katlı çerçeve sistemi (a) eksenel kuvvet, (b) moment diyagramı

Optimum tasarım sonucu elde edilen en iyi sonuç için Sap2000 programı yardımı ile çelik çerçeve sisteminin eksenel kuvvet ve moment diyagramı Şekil 3.14.'da gösterilmiştir.

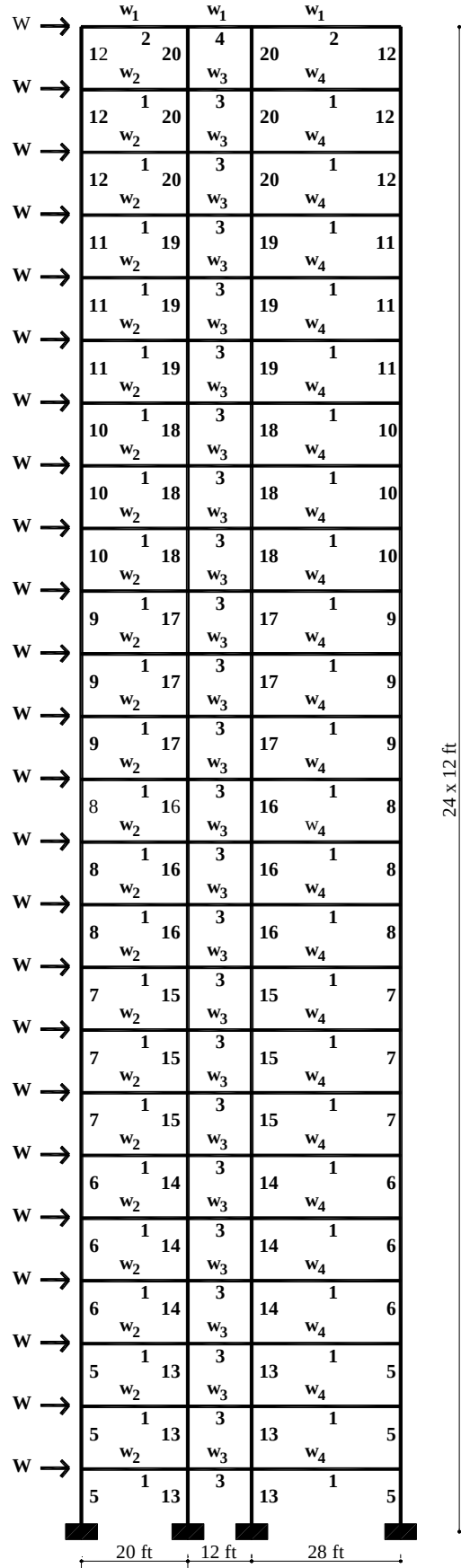
3.3. Üç Açıklıklı Yirmi Dört Katlı Düzlem Çelik Çerçeve

Şekil 3.15.'de üç açıklıklı yirmi dört katlı düzlem çelik çerçevenin boyutları, yükleme şekli, elemanların grup numaraları gösterilmiştir. Bu çelik çerçeve Şekil 3.15.'de verilen boyutlandırma, yükleme ve gruplandırılmış şekilde Camp vd., (2004) tarafından karınca koloni algoritması (ACO) ile, harmoni arama metodu (HS) ile Degertekin (2012) tarafından, geliştirilmiş karınca koloni algoritması (IACO) ile Kaveh ve Talatahari (2010) tarafından ve öğretme öğrenme tabanlı algoritma (TLBO) ile Togan (2013) tarafından kullanılarak optimum tasarımı yapılmıştır.

Sistemde maksimum yer değiştirme ve katlar arası yer değiştirme kısıtları Denklem 3.3, 3.4 ve Denklem 3.7, 3.8' de verilen gerilme kısıtları Amerikan Çelik Yapılar Enstitüsü - Yük ve Dayanım Faktörü Tasarımı (AISC-LRFD, 2001) şartnamesinden alınmıştır. Lineer olmayan analiz için bir önceki örnekteki Denklem 3.18 ile ikinci dereceden etkiler göz önüne alınmıştır. Kiriş ve kolon elemanları LRFD' de bulunan ve Ek.4'de verilen 267 adet W kesitlerinden seçilmiştir.

Çelik çerçeve sistem 100 düğüm noktası ve 168 elemandan oluşturulmuştur. Kirişler dört ayrı grup, kolonlar ise on altı ayrı gruptan oluşmak üzere çelik çerçeve toplam yirmi gruptan oluşmuştur.

Çelik çerçeve tek tip yüklemeden oluşmaktadır. Yükleme için $W = 5761.85 \text{ lb (25,63 kN)}$, $w_1 = 300 \text{ lb/ft (4,38 kN/m)}$, $w_2 = 436 \text{ lb/ft (6,36 kN/m)}$, $w_3 = 474 \text{ lb/ft (6,92 kN/m)}$ ve $w_4 = 408 \text{ lb/ft (5,95 kN/m)}$ olarak verilmiştir.



Şekil 3.15. Üç açıklıklı 24 katlı çerçeve sisteminin boyutlandırılması, yükleme şekli ve elemanların gruplandırılması

C# programı ile geliştirilen yapay arı koloni algoritması (ABC) literatürde Camp vd., (2004) tarafından karınca koloni algoritması (ACO) ile, harmoni arama metodu (HS) ile Degertekin (2012) tarafından, geliştirilmiş karınca koloni algoritması (IACO) ile Kaveh ve Talatahari (2010) tarafından ve öğretim öğrenme tabanlı algoritma (TLBO) ile Togan (2013) tarafından kullanılan bu tasarım örneğine uygulanmıştır. Programın çok katlı düzlem çelik çerçeveler üzerinde performansını görebilmek için birbirinden bağımsız olarak 5 kere çalıştırılmıştır, optimum tasarım için en iyi ve ortalama sonuçlar, en üst kattaki maksimum yer değiştirme, katlar arası yer değiştirme elde edilmiştir. En iyi tasarım ağırlığı ve buna denk gelen kesitler, ortalama tasarım ağırlığı, analiz sayısı, gerilme ihlali ve deplasman ihlali Çizelge 3.3. 'de verilmiştir.

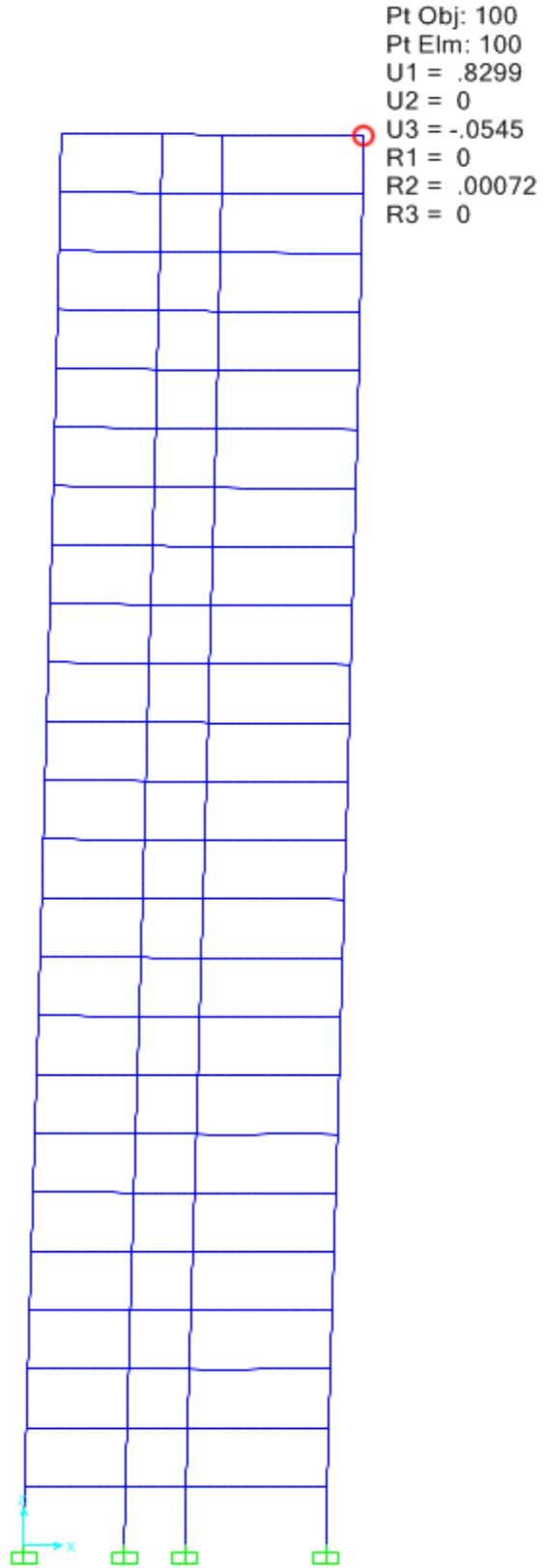
Malzeme için Elastisite Modülü $E = 29732$ ksi (200GPa), malzeme akma gerilmesi $f_y = 33,4$ ksi (248,2 MPa) alınmıştır. Maksimum yer değiştirme sınırı 29,20 cm ve katlar arası maksimum yer değiştirme sınırı tüm katlar için 1,216 cm olarak verilmiştir. Mesnetlenme biçimi ankastre seçilmiştir.

Optimum tasarım sonucu minimum tasarım ağırlığı 227903 lb, ortalama ağırlık ise 232379 lb bulunmuştur. Yapılan program sonucunda yapay arı koloni algoritması için gerilme ihlali çıkmayıp, 0,0082 deplasman ihlali çıkmıştır. Yapılan programa göre Camp vd. karınca koloni algoritmasındaki tasarım için gerilme ihlali olarak 1,680 ve deplasman ihlali olarak 0,053 kısıt ihlali, Degertekin'in harmoni arama metoduna göre oluşturduğu tasarım için gerilme ihlali olarak 1,858 ve deplasman ihlali olarak 2,457 kısıt ihlali, Kaveh ve Talatahari'nin geliştirilmiş karınca koloni algoritması ile bulunduğu tasarımda gerilme ihlali olarak 4,714 ve deplasman ihlali olarak 4,846 kısıt ihlali, Togan'ın öğrenme öğretim tabanlı algoritması ile bulunduğu tasarım için gerilme ihlali olarak 8,836 ve deplasman ihlali olarak 6,401 kısıt ihlali görülmüştür.

Çizelge 3.3. Üç açıklıklı yirmi dört katlı çerçeve sisteminin optimum sonuçları

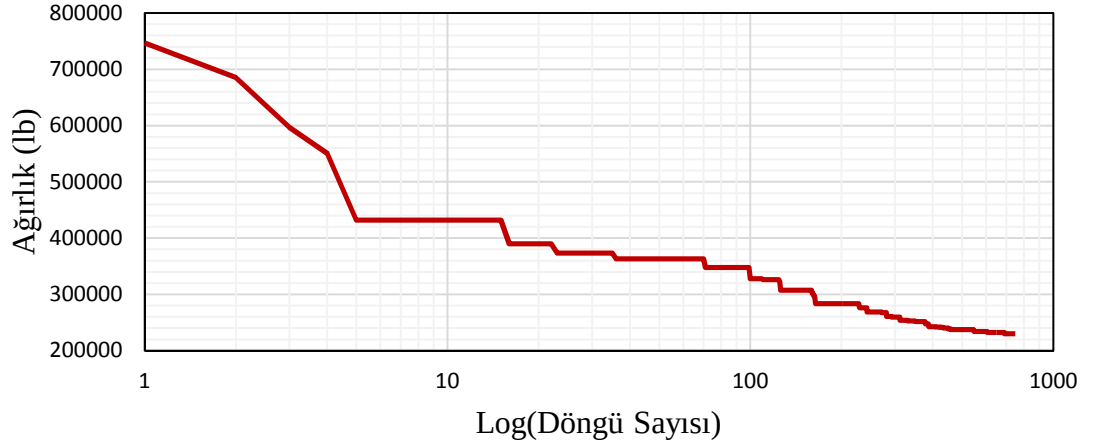
Eleman grup no.	AISC W kesitler				
	ACO Camp vd.	HS Degertekin	IACO Kaveh ve Talatahari	TLBO Togan	Bu Çalışma
1	W30 × 90	W30 × 90	W30 × 99	W30 × 90	W30 × 90
2	W8 × 18	W10 × 22	W16 × 26	W8 × 18	W12 × 14
3	W24 × 55	W18 × 40	W18 × 35	W24 × 62	W21 × 55
4	W8 × 21	W12 × 16	W14 × 22	W6 × 9	W12 × 30
5	W14 × 145	W14 × 176	W14 × 145	W14 × 132	W14 × 257
6	W14 × 132	W14 × 176	W14 × 132	W14 × 120	W14 × 145
7	W14 × 132	W14 × 132	W14 × 120	W14 × 99	W14 × 120
8	W14 × 132	W14 × 109	W14 × 109	W14 × 82	W14 × 109
9	W14 × 68	W14 × 82	W14 × 48	W14 × 74	W14 × 74
10	W14 × 53	W14 × 74	W14 × 48	W14 × 53	W14 × 53
11	W14 × 43	W14 × 34	W14 × 34	W14 × 34	W14 × 34
12	W14 × 43	W14 × 22	W14 × 30	W14 × 22	W14 × 22
13	W14 × 145	W14 × 145	W14 × 159	W14 × 109	W14 × 159
14	W14 × 145	W14 × 132	W14 × 120	W14 × 99	W14 × 132
15	W14 × 120	W14 × 109	W14 × 109	W14 × 99	W14 × 120
16	W14 × 90	W14 × 82	W14 × 99	W14 × 90	W14 × 109
17	W14 × 90	W14 × 61	W14 × 82	W14 × 68	W14 × 99
18	W14 × 61	W14 × 48	W14 × 53	W14 × 53	W14 × 53
19	W14 × 30	W14 × 30	W14 × 38	W14 × 34	W14 × 53
20	W14 × 26	W14 × 22	W14 × 26	W14 × 22	W14 × 22
Ağırlık(lb)	220465	214860	217464	203008	227903
Ortalama ağır.(lb)	*	*	*	*	232379
Analiz sayısı (#)	15500	14651	3500	12000	37500
Gerilme ihlali	1,680	1,858	4,714	8,836	0
Deplasman ihlali	0,053	2,457	4,846	6,401	0,0082

*Değerler önceki çalışmalarda verilmemiştir.



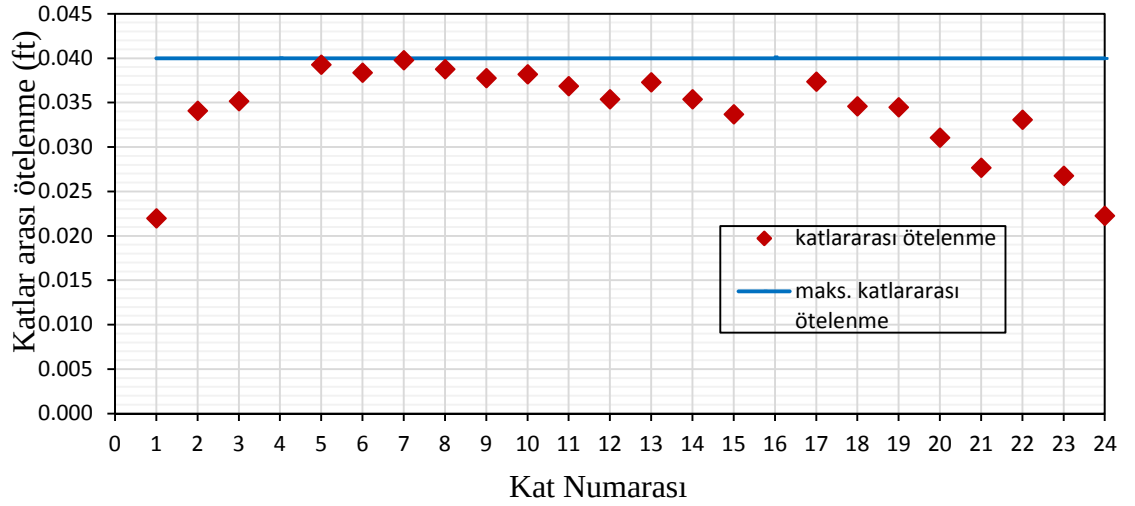
Şekil 3.16. Üç açıklıklı 24 katlı çerçeve sistemi için maksimum yer değiştirme (ft)

Son katta oluşan maksimum yer değiştirme Şekil 3.16.'da gösterilmiştir. 0,8299 ft (25,29 cm) olarak ölçülen değer x doğrultusunda oluşan yer değiştirmedir maksimum değerden (29,2 cm) küçüktür. Şekil 3.18'de ise katlar arası yer değiştirmeler maksimum sınırla ile ilişkisi gösterilmiştir.

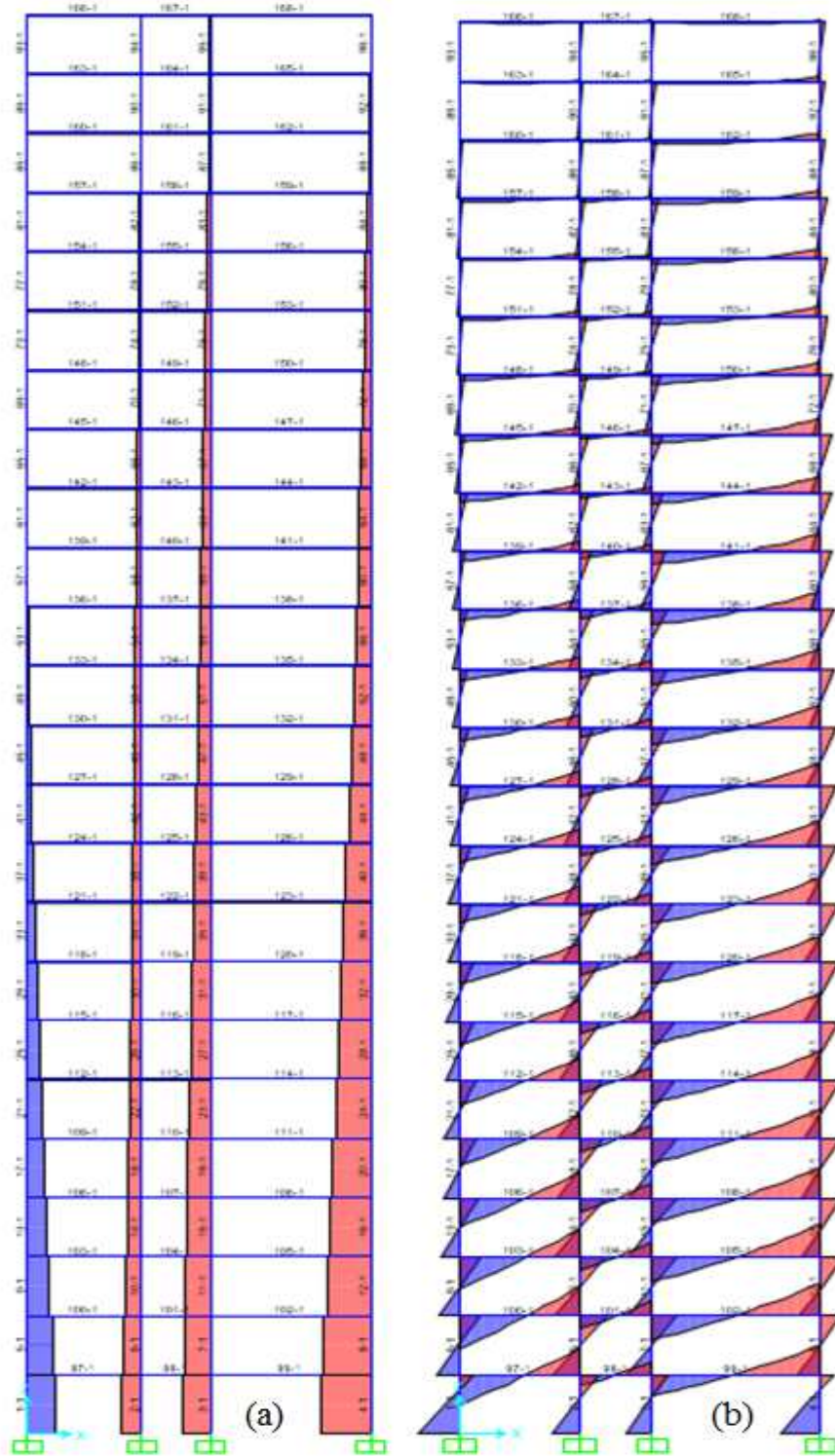


Şekil 3.17. Üç açıklıklı 24 katlı çerçeve sistemi için ağırlık ve döngü sayısı ilişkisi

Optimum tasarım sonuçları elde edildikten sonra minimum ağırlığı veren en iyi tasarım için ağırlık-döngü sayısı ilişkisi yukarıdaki şekilde verilmiştir. Yatay eksen logaritmiktir.



Şekil 3.18. Üç açıklıklı 24 katlı çerçeve için katlar arası yer değiştirmeler



Şekil 3.19. Üç açıklıklı 24 katlı çerçeve sistemi için (a) eksenel kuvvet, (b) moment diyagramı

Optimum tasarım sonucu elde edilen en iyi tasarım için Sap2000 programı yardımı ile çelik çerçeve sisteminin eksenel kuvvet ve moment diyagramı Şekil 3.19.'da gösterilmiştir.

3.4. Tek Açıklıklı Dört Katlı Uzay Çelik Çerçeve

Şekil 3.20.'de tek açıklıklı dört katlı uzay çelik çerçevenin boyutları gösterilmiştir. Bu çelik çerçeve ilk kez bu çalışmada kullanılarak optimum tasarım için kullanılmıştır.

Gerilme kısıtları Amerikan Çelik Yapılar Enstitüsü - Yük ve Dayanım Faktörü Tasarımı (AISC-LRFD, 2001) şartnamesinden alınmıştır. Katlar arası yer değiştirme kısıtı ise $\delta < h/300$ şeklinde alınmıştır. Maksimum yer değiştirme kısıtı $\Delta < H/400$ şeklinde alınmıştır (Δ çelik çerçeve üzerinde oluşan maksimum yer değiştirme). Kiriş ve kolon elemanları için LRFD 'de sunulan 267 adet W kesitlerinden 64 adet W kesiti seçilmiştir. Seçilen kesitler aşağıda verilmiştir.

$L = \{ \text{"W8} \times 15", \text{"W8} \times 18", \text{"W8} \times 21", \text{"W8} \times 24", \text{"W8} \times 28", \text{"W8} \times 31", \text{"W8} \times 35", \text{"W8} \times 40", \text{"W8} \times 48", \text{"W8} \times 58", \text{"W8} \times 67", \text{"W10} \times 12", \text{"W10} \times 15", \text{"W10} \times 17", \text{"W10} \times 19", \text{"W10} \times 22", \text{"W10} \times 26", \text{"W10} \times 30", \text{"W10} \times 33", \text{"W10} \times 39", \text{"W10} \times 45", \text{"W10} \times 49", \text{"W12} \times 14", \text{"W12} \times 16", \text{"W12} \times 19", \text{"W12} \times 22", \text{"W12} \times 26", \text{"W12} \times 30", \text{"W12} \times 35", \text{"W12} \times 40", \text{"W12} \times 45", \text{"W12} \times 50", \text{"W12} \times 53", \text{"W12} \times 58", \text{"W14} \times 22", \text{"W14} \times 26", \text{"W14} \times 30", \text{"W14} \times 34", \text{"W14} \times 38", \text{"W14} \times 43", \text{"W14} \times 48", \text{"W14} \times 53", \text{"W14} \times 61", \text{"W14} \times 68", \text{"W16} \times 26", \text{"W16} \times 31", \text{"W16} \times 36", \text{"W16} \times 40", \text{"W16} \times 45", \text{"W16} \times 50", \text{"W16} \times 57", \text{"W16} \times 67", \text{"W18} \times 35", \text{"W18} \times 40", \text{"W18} \times 46", \text{"W18} \times 50", \text{"W18} \times 55", \text{"W21} \times 44", \text{"W21} \times 50", \text{"W21} \times 57", \text{"W21} \times 62", \text{"W24} \times 55", \text{"W24} \times 62", \text{"W24} \times 68"} \}.$

Çelik çerçeve sistem 20 düğüm noktası ve 32 elemandan oluşturulmuştur. Bu uzay çelik çerçeve altı gruptan oluşmaktadır. Kolonlar iki ayrı gruptan oluşmuştur. İlk iki katın tüm kolonları birinci grubu, üç ve dördüncü kattaki tüm kolonlar ise ikinci grubu oluşturmaktadır. Kirişler ise dört ayrı gruptan oluşmuştur. Bir ve ikinci kat x doğrultusundaki tüm kirişler üçüncü grubu, bir ve ikinci kattaki y doğrultusundaki tüm kirişler dördüncü grubu, üç ve dördüncü kattaki x doğrultusundaki tüm kirişler beşinci grubu, üç ve dördüncü kattaki y doğrultusundaki tüm kirişler ise altıncı ve son grubu oluşturmaktadır.

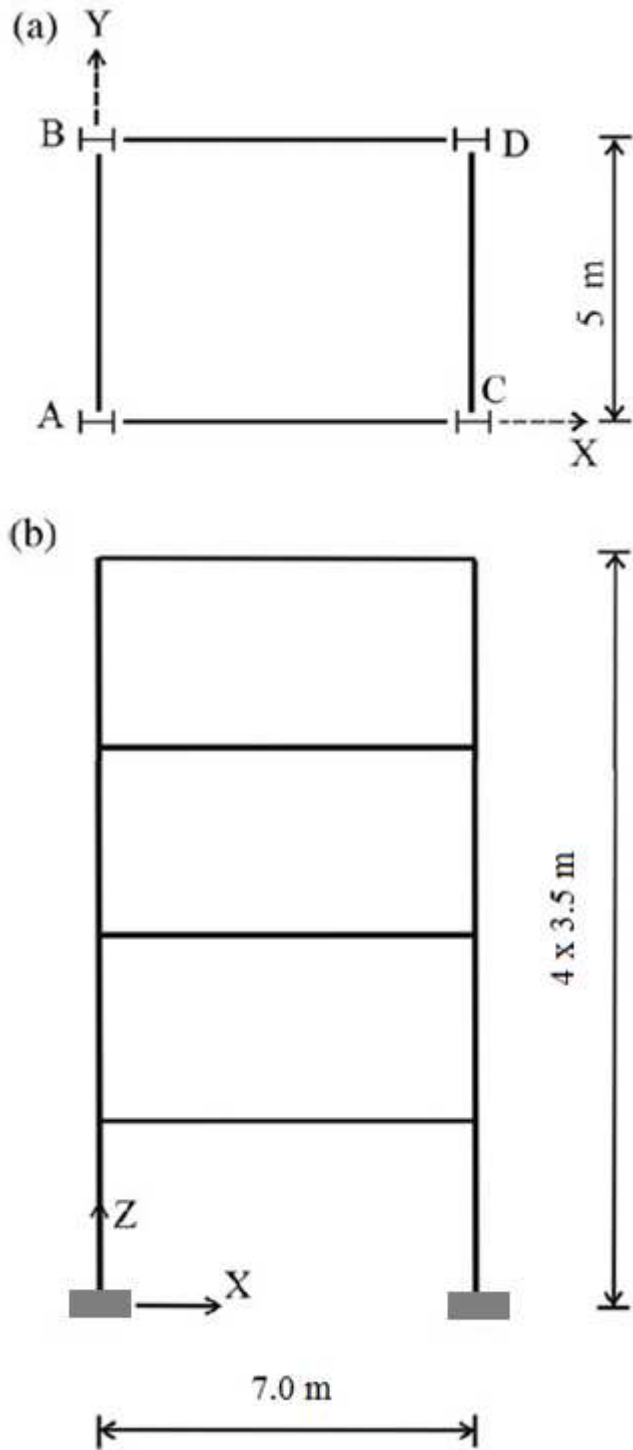
Uzay çelik çerçeve ölü yük (D), hareketli yük (L), çatı hareketli yükü (L_r) ve rüzgâr yükü (W) içeren dört farklı tip yükleme ile yüklenmiştir. Bu yüklemeler Amerikan Çelik Yapılar Enstitüsü - Yük ve Dayanım Faktörü Tasarımı (AISC-LRFD, 2001) şartnamesinden alınmıştır.

I: 1.4D,

II: 1.2D + 1.6L + 0.5L_r,

III: 1.2D + 1.6L_r + 0.5L,

$$IV: 1.2D + 1.3W + 0.5L + 0.5L_r$$



Şekil 3.20. Dört katlı otuz iki elemanlı uzay çerçeve: (a) plan, (b) yandan görünüş

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

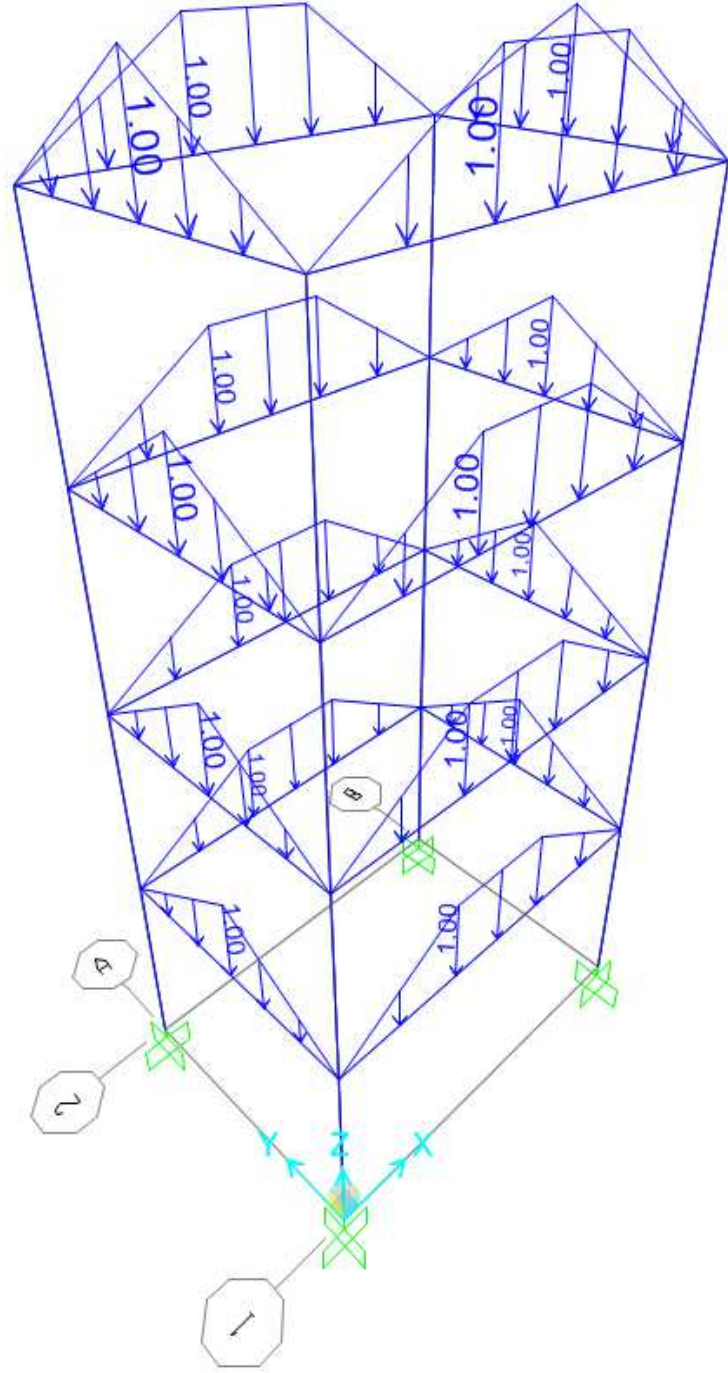
ise rüzgar önem katsayısıdır. Uzay çelik çerçevenin D yüklemesine maruz kaldığı varsayıldığında C_e katsayısı Uniform Building Şartnamesi (Tablo 16-G) 14 m bina yüksekliğine karşı gelen değer olarak 1,653 alınmıştır. C_q basınç için 0,8 emme için – 0,5 alınmıştır. Rüzgar önem katsayısı I_w 1 olarak alınmıştır.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

Çizelge 3.4. Bina yüksekliğine bağlı rüzgar faktörü C_e (UBC97 tablo 16-G)

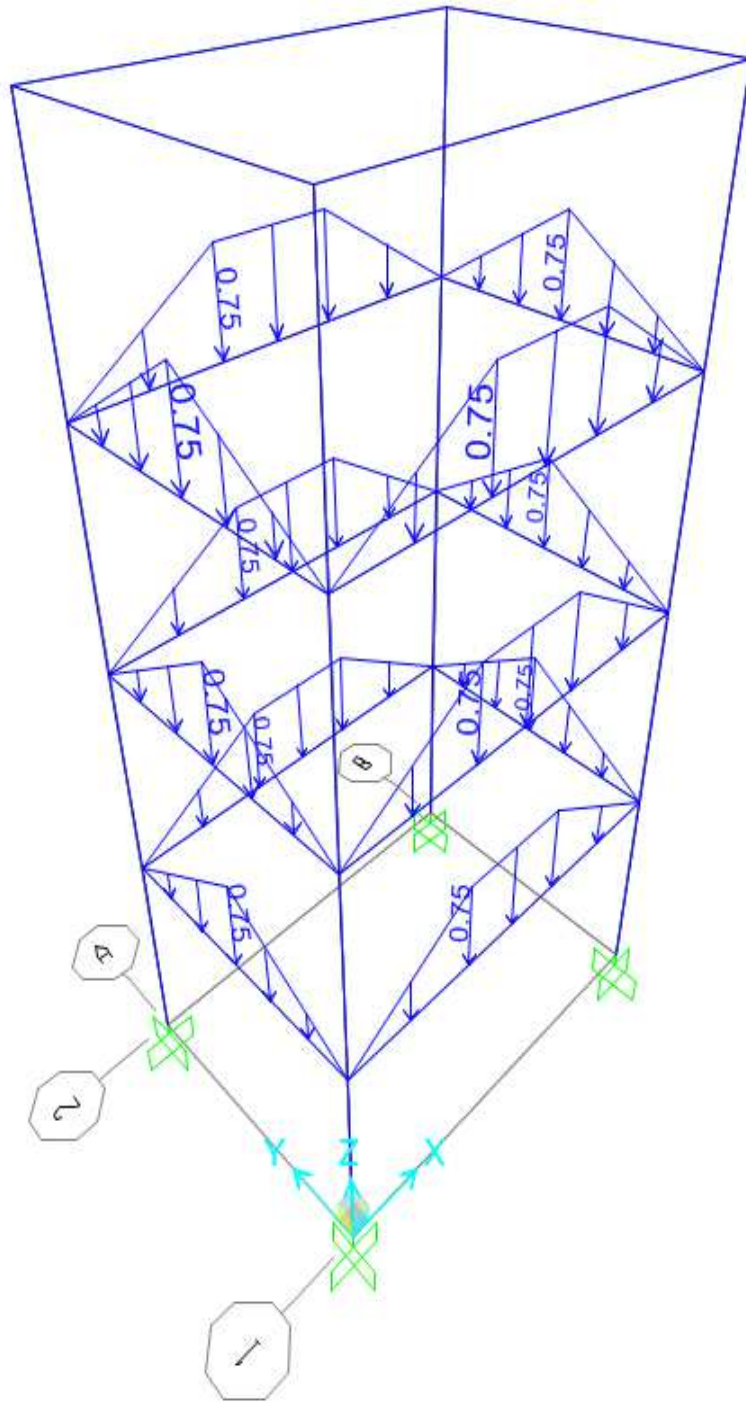
Bina Yerden Yüksekliği (feet)	D'ye maruz	C'ye maruz	B'ye maruz
0 – 15	1,39	1,06	0,62
20	1,45	1,13	0,67
25	1,50	1,19	0,72
30	1,54	1,23	0,76
40	1,62	1,31	0,84
60	1,73	1,43	0,95
80	1,81	1,53	1,04
100	1,88	1,61	1,13
120	1,93	1,67	1,20
160	2,02	1,79	1,31
200	2,10	1,87	1,42
300	2,23	2,05	1,63
400	2,34	2,19	1,80

1feet = 0. 304 m

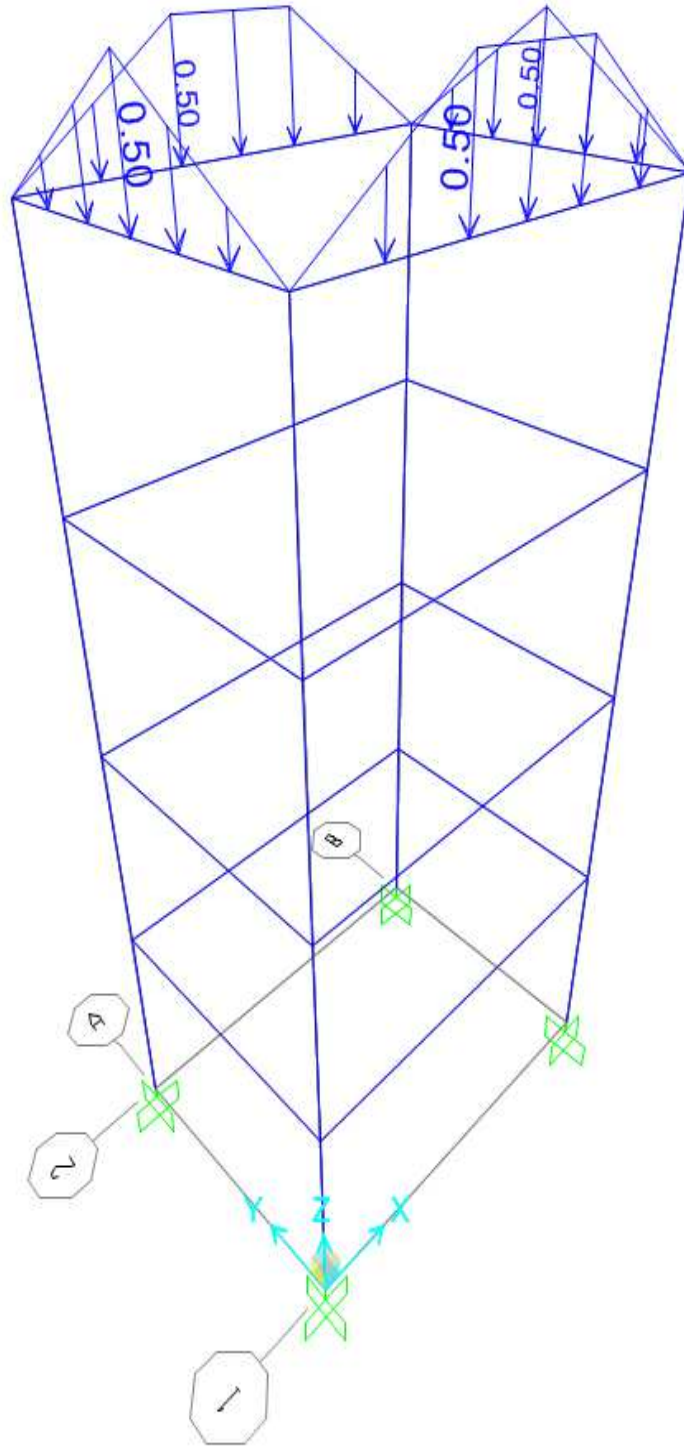
Çelik çerçeve sistemi Şekil 3.21., Şekil 3.22., Şekil 3.23. ve Şekil 3.24.'da görüldüğü gibi Sap2000 programında tasarlanmıştır. Sistemde maksimum yer değiştirme ve katlar arası yer değiştirme Denklem 3.3 ve 3.4 deki kısıtlara göre, gerilme durumu ise Denklem 3.7 ve 3.8 de verilen gerilme kısıtlarına göre yapılmıştır. Lineer olmayan analiz için Denklem 3.18 kullanılmıştır.



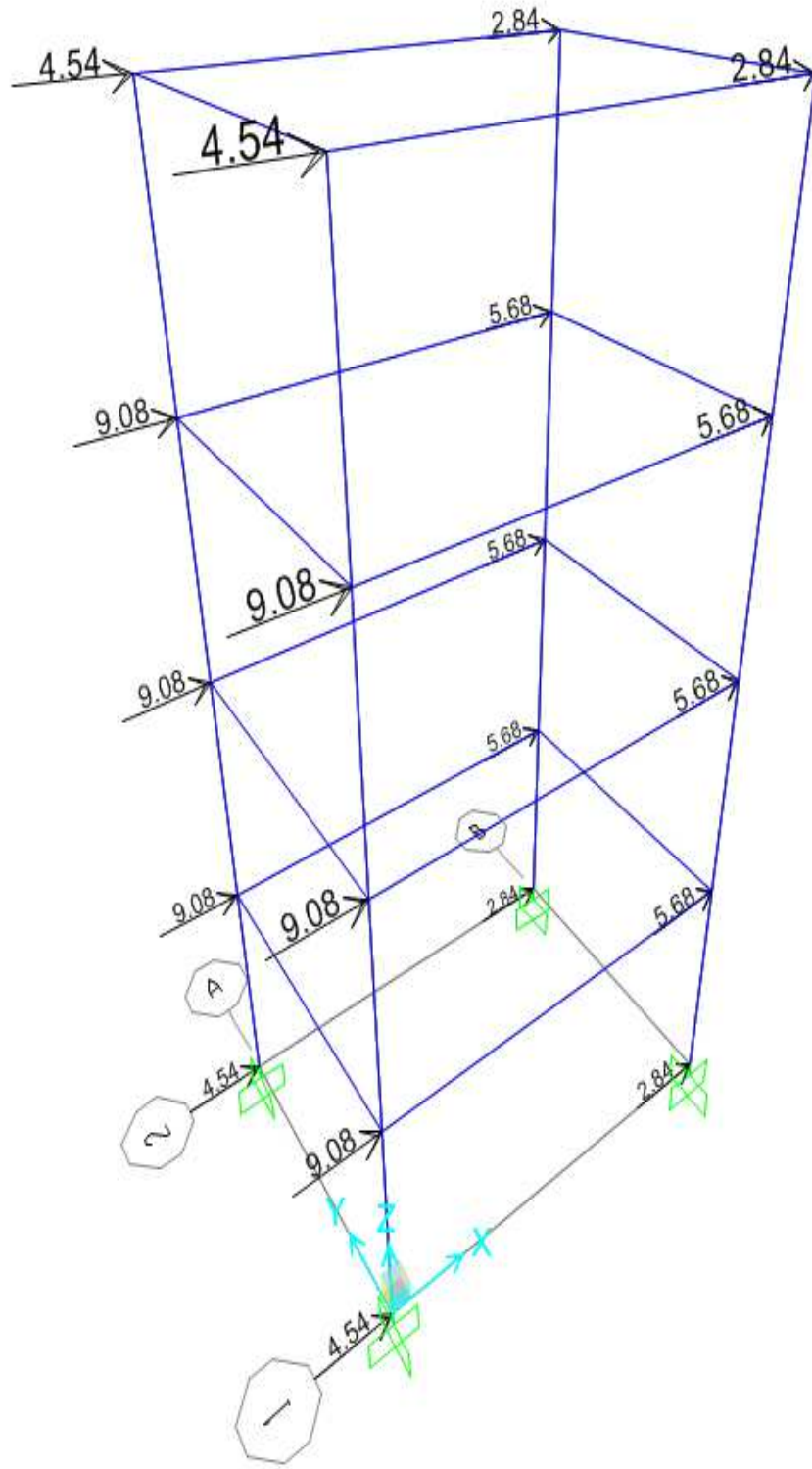
Şekil 3.21. Dört katlı otuz iki elemanlı uzay çerçeve (D) yüklemesi



Şekil 3.22. Dört katlı otuz iki elemanlı uzay çerçeve (L) yüklemesi



Şekil 3.23. Dört katlı otuz iki elemanlı uzay çerçeve (L_r) yüklemesi



Şekil 3.24. Dört katlı otuz iki elemanlı uzay çerçeve W yüklemesi

Yapılan program, yapay arı koloni algoritmasının performansını görmek için uzay çelik çerçevelerin optimum tasarımı için uygulanmıştır. Bunun için program birbirinden

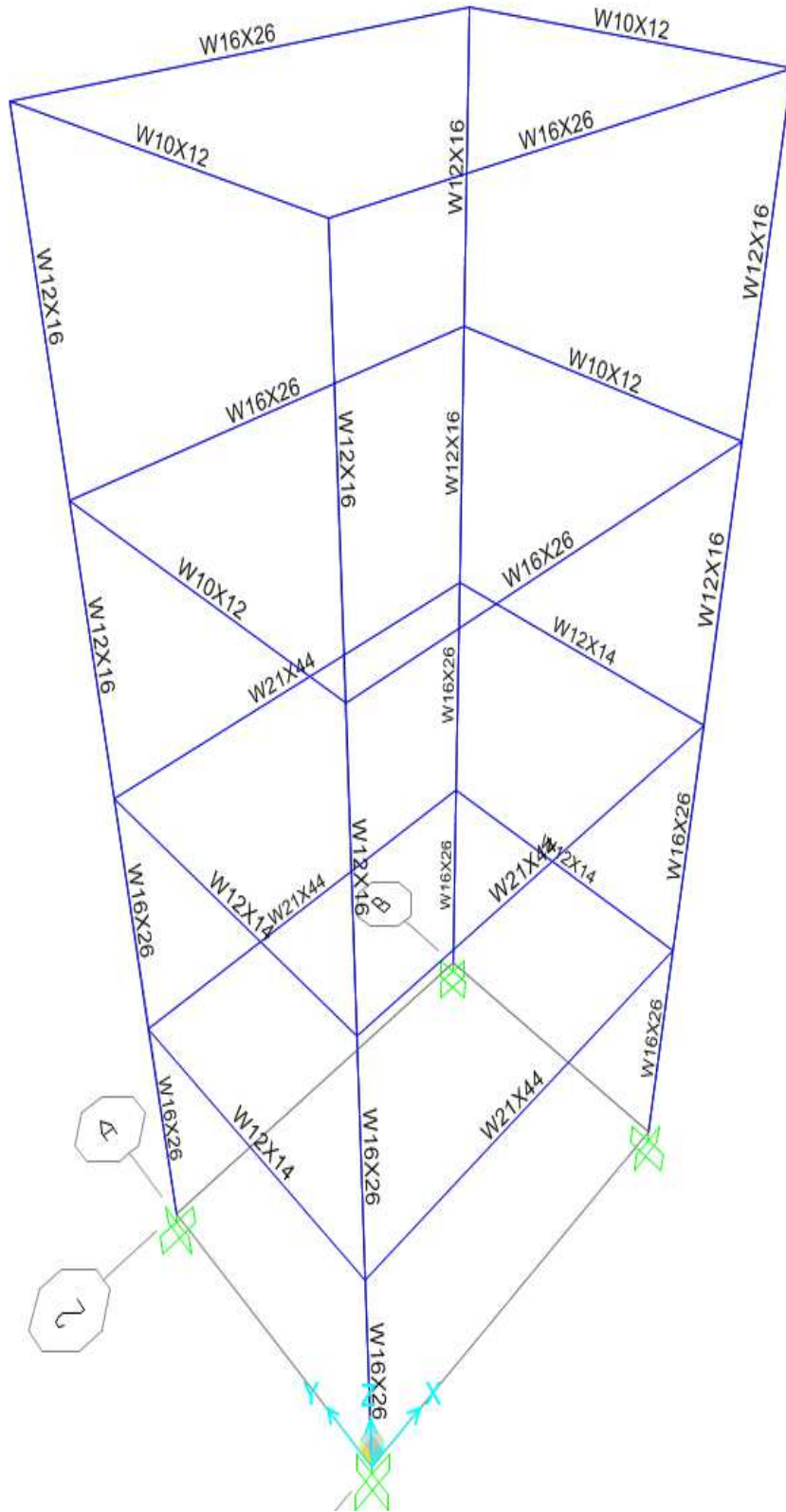
bağımsız olarak beş kere çalıştırılmıştır, optimum tasarım için en iyi ağırlık ve o ağırlığı oluşturan kesitler, ortalama ağırlık, en üst kattaki maksimum yer değiştirme, katlar arası yer değiştirme, toplam analiz sayısı, standart sapma elde edilmiştir ve Çizelge 3.5. 'de gösterilmiştir.

Malzeme için Elastisite Modülü $E = 29732$ ksi (200 GPa), malzeme akma gerilmesi $f_y = 33,4$ ksi (248,2 MPa) alınmıştır. Maksimum yer değiştirme sınırı 3,5 cm ve katlar arası maksimum yer değiştirme sınırı tüm katlar için 1,16 cm olarak verilmiştir. Mesnetlenme biçimi ankastre verilmiştir.

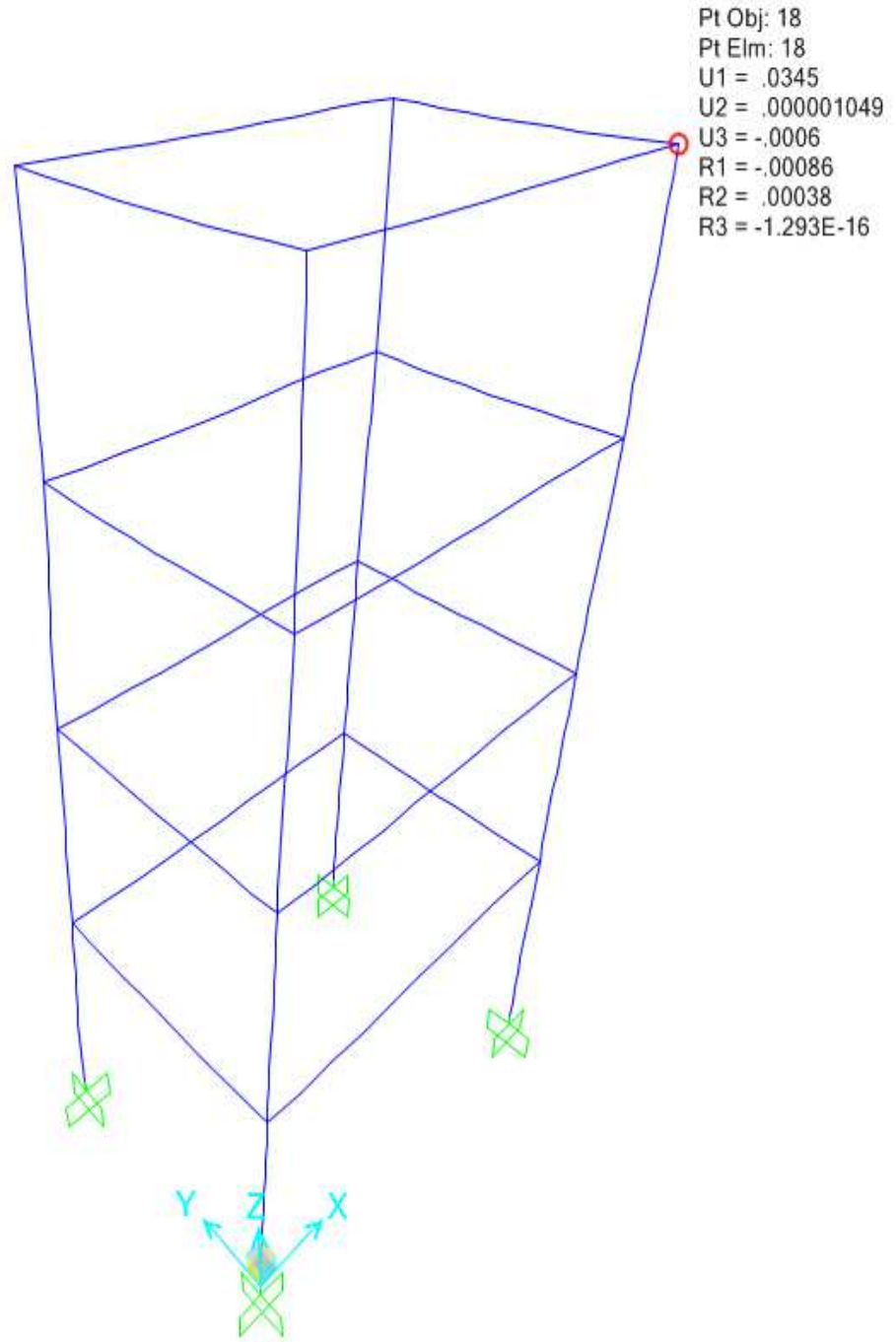
Çizelge 3.5. Dört katlı otuz iki elemanlı uzay çerçevenin optimum sonuçları

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

Çizelge 3.5.'de görüldüğü gibi minimum tasarım ağırlığı 54,69 kN, ortalama ağırlık ise 54,97 kN, en üst kat ötelenmesi 3,45 cm, katlar arası maksimum yer değiştirme, 1,14 cm, standart sapma ise 0,28 bulunmuştur.

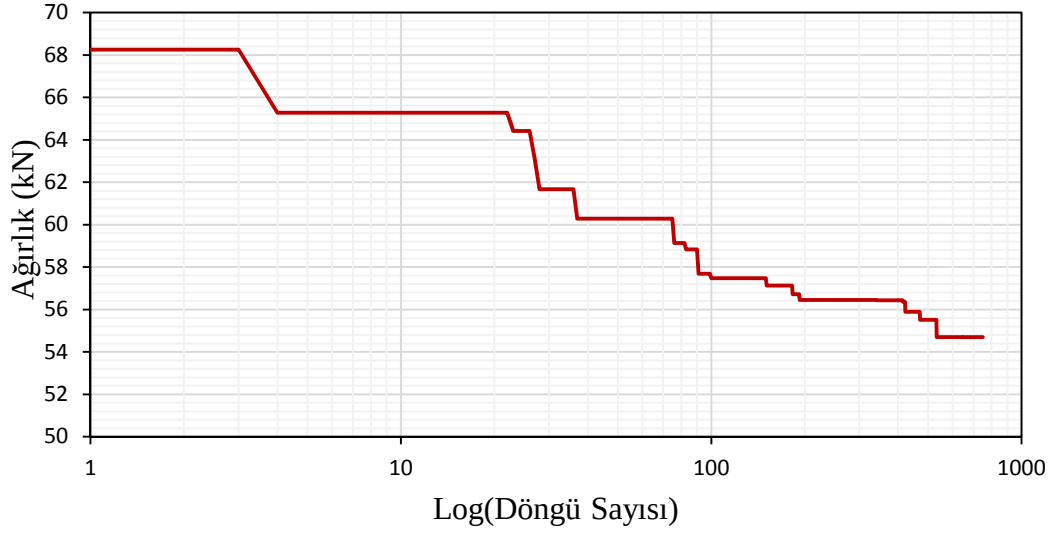


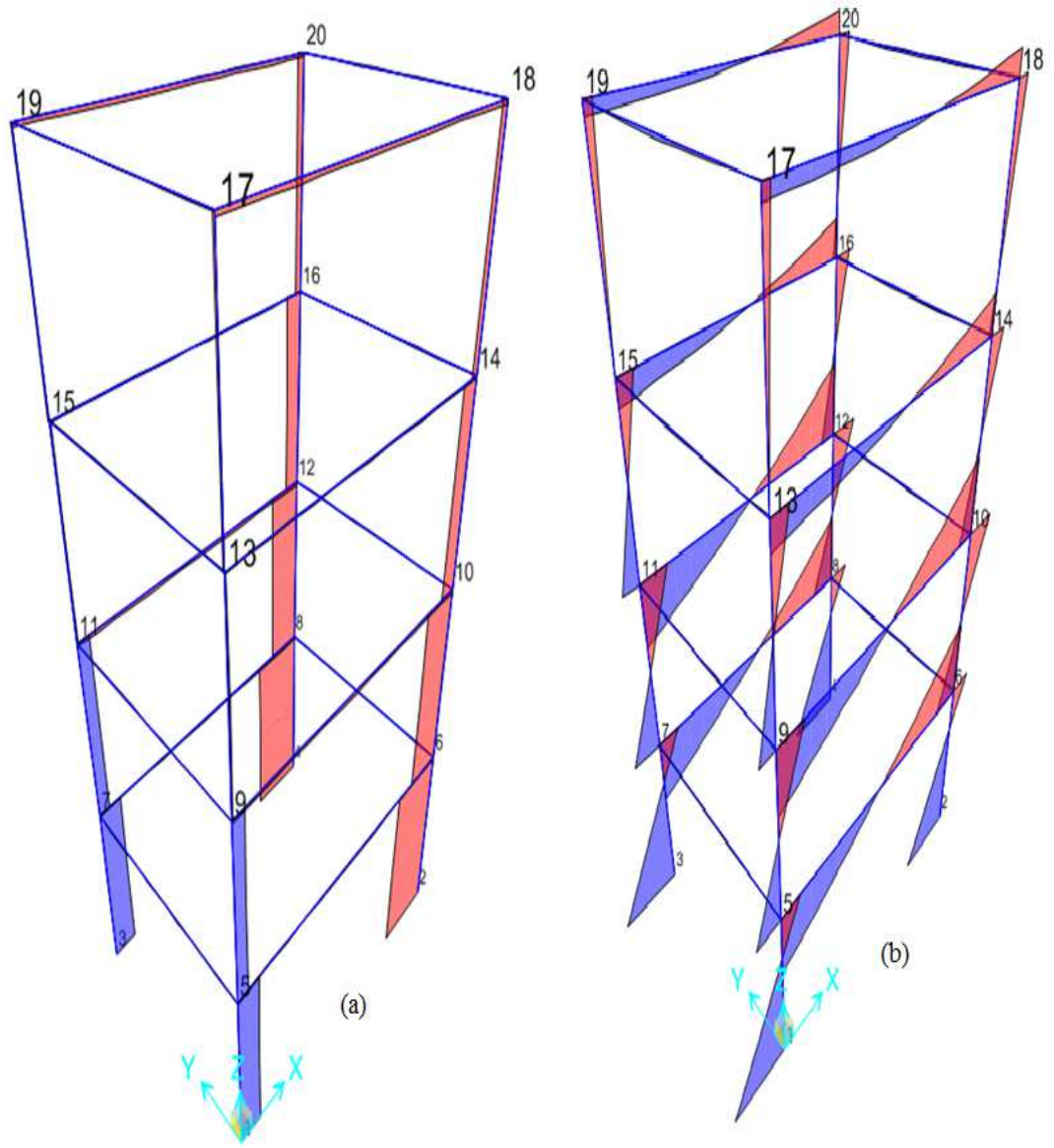
Şekil 3.25. Dört katlı otuz iki elemanlı uzay çerçeve optimum kesitlerle Sap2000 görünümü



Şekil 3.26. Dört katlı otuz iki elemanlı uzay çerçeve maksimum yer değiştirmesi (m)

Son katta oluşan maksimum yer değiştirme yukarıdaki şekilde gösterilmiştir. 0,0345 m (3,45 cm) olarak ölçülen değer x doğrultusunda oluşan yer değiştirmedir ve maksimum sınır değerinden (3,5 cm) küçüktür.





Şekil 3.29. Dört katlı otuz iki elemanlı uzay çerçeve için 4. yükleme (a) aksenal kuvvet, (b) moment diyagramı

4. TARTIŞMA VE SONUÇLAR

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

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

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

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

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

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

Bu çalışmada çelik çerçeveler için çözümler yapıldı. Hazırlanan programda gerekli düzenlemeler yapılarak betonarme yapıların ağırlık optimizasyonu yapılabilir.

KAYNAKLAR

- Akay, B., 2009. Nümerik optimizasyon problemlerinde yapay arı kolonisi (artificial bee colony) algoritmasının performans analizi, Doktora Tezi, Kayseri Erciyes Üniversitesi, Fen Bilimleri Enstitüsü, Kayseri.
- Arora, JS . 2007, Optimization of structural and mechanical systems, World Scientific, Singapore.
- Aydogdu, I., Akın, A. and Saka, M.P., 2012. Optimum design of steel space frames by artificial bee colony algorithm, 10th International Congress on Advances in Civil Engineering, Ankara, Turkey.
- Bonabeau, E., Dorigo, M. and Theraulaz, G., 1999. Swarm intelligence: from natural to artificial systems, Oxford University Press, Inc., New York, NY, USA, ISBN 0195131.
- Camp, C.V. ve Bichon, B.J., 2004. Design of space trusses using ant colony optimization, J Struct Eng, ASCE, 130, 741-751.
- Coello, C.A.C., 2002. Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: survey of the state of the art, Computer Methods in Applied Mechanics and Engineering, 191, 1245-1287.
- Deb, K. 2000. An efficient constraint handling method for genetic algorithms, Comput Methods Appl Mech Eng, 186(2-4), 311-338.
- Deb, K. ve Gulati, S. 2001, Design of truss-structures for minimum weight using genetic algorithms, Finite Elements in Analysis and Design, Vol: 37, Issue 5, ss: 447-465.
- Degertekin, S.O. ve Hayalioglu, M.S., 2013. Sizing truss structures using teaching-learning-based optimization, Computers and Structures, 119, 177-188.
- Degertekin, S.O., 2012. Improved harmony search algorithms for sizing optimization of truss structures, Computers and Structures, 92-93, 229-241.
- Degertekin, S.O., 2012. Optimum design of geometrically non-linear steel frames using artificial bee colony algorithm, Steel and Composite Structures, Vol. 12, No. 6, 505-522.
- Degertekin, S.O., Saka, M.P. ve Hayalioglu, M.S., 2008. Optimal load and resistance factor design of geometrically nonlinear steel space frames via tabu search and genetic algorithm, Engineering Structures, 30, 197-205.
- Dogan, E. ve Saka, M.P., 2012. Optimum design of unbraced steel frames to LRFD-AISC using particle swarm optimization, Advances in Engineering Software, 46, 27-34.
- Fu, J-F., Fenton, R.G. ve Cleghorn, W.L., 1991. A mixed integer-discrete-continuous programming method and its application to engineering design optimization, Engineering Optimization, Vol. 17, pp. 263-280.

- Gero, M.B.P., Garcia, A.B. ve Diaz, J.J.C., 2006. Design optimization of 3D steel structures: Genetic algorithms vs. classical Techniques, *Journal of Constructional Steel Research*, 62, 1303-1309.
- Hadidi, A., Kazemzadeh A. ve Azaf, S.K. 2010, Strutural optimization using artificial bee colony algorithm, 2nd International Conference on Engineering Optimization, September 6-9, Lizbon, Portugal.
- Hasancebi, O. ve Kazemzadeh A., 2012. An exponential big bang-big crunch algorithm for discrete design optimization of steel frames, *Computers and Structures*, 110-111, 167-179.
- Hasançebi, O. ve Azad S.K., 2013. Discrete size optimization of steel trusses using a refined bigbang-big crunch algorithm, *Engineering Optimization*, DOI:10.1080/0305215X.2012.748047.
- Hasancebi, O., Bahcecioglu, T., Kurc, O. ve Saka, M.P., 2011. Optimum design of high-rise steel buildings using an evolution strategy integrated parallel algorithm, *Computers and Structures*, 89, 2037-2051.
- Hua, H.M., 1983. Optimization of structures of discrete-sized elements, *Computers and Structures*, Vol. 17, No. 3, pp. 327-333.
- Kameshki, E.S. ve Saka, M.P., 2001. Genetic algorithm based optimum bracing design of non-swaying tall plane frames, *Journal of Constructional Steel Research*, 57, 1081-1097.
- Karaboga, D. ve Basturk, B., 2008. On the performance of artificial bee colony (ABC) algorithm, *Applied Soft Computing*, 8(1), 687-697.
- Karaboga, D., 2005. An idea based on honey bee swarm for numerical optimization, Technical Report TR06, Erciyes University, Engineering Faculty, Computer Engineering Department.
- Kaveh, A. ve Talatahari, S., 2009. A particle swarm ant colony optimization for truss structures with discrete variables, *Journal of Constructional Steel Research*, 65, 1558-1568.
- Kaveh, A. ve Talatahari, S., 2009. Particle swarm optimizer, ant colony strategy and harmony search scheme hybridized for optimization of truss structures, *Computers and Structures* 87, 267-283.
- Kaveh, A. ve Talatahari, S., 2010. An improved ant colony optimization for the design of planar steel frames, *Engineering Structures*, 32, 864-873.
- Kaveh, A., Azar, B.F., Hadidi, A., Sorochi, F.R. ve Talatahari, S., 2010. Performance-based seismic design of steel frames using ant colony optimization, *Journal of Constructional Steel Research*, 66, 566-574.
- Kripka, M., 2004, Discrete optimization of trusses by simulated annealing, *J. Braz. Soc. Mech. Sci. & Eng.* vol.26 no.2 Rio de Janeiro.

- Lee K.S. Greem, A.W. Lee, S. 2005, Harmony Search heuristic algorithm for discrete structural optimization, *Engineering Optimization*, 37, 7, 663-684.
- Li, L.J., Huang, Z.B. ve Liu, F., 2009. A heuristic particle swarm optimization method for truss structures with discrete variables, *Comput Struct*, 87, 435-444.
- Marler R.T. ve Arora J.S., 2004, Survey of multi-objective optimization methods for engineering, *Structural and multidisciplinary optimization*, Vol: 26, Issue 6, pp 369-395.
- Pereza R.E ve Behdinan K. 2007, Particle swarm approach for structural design optimization, *Computers & Structures*, Volume 85, Issues 19–20, October 2007, Pages 1579–1588.
- Pezeshk, S., Camp, C. ve Chen, D., 2000. Design of nonlinear framed structures using genetic optimization, *J. Struct. Eng.*, 126(3), 382-388.
- Rajeev S. ve Krishnamoorthy, C.S., 1992. Discrete optimization of structures using genetic algorithms. *J Struct Eng, ASCE* 118(5):1233-1251.
- Rao, S.S., 2009. *Engineering optimization – theory and practice*, Fourth Edition Wiley Eastern: New Delhi.
- Reinschmidt, K.F., 1971. Discrete structural optimization, *Journal of Structural Engineering*, ASCE, Vol 97, No. 1, pp. 133-156.
- Safari, D., Maheri, M.R. ve Maheri, A., 2011. Optimum design of steel frames using a multiple-deme GA with improved reproduction operators, *Journal of Constructional Steel Research*, 67, 1232-1243.
- Saka, M.P., 2009. Optimum design of steel sway frames to BS5950 using harmony search algorithm, *Journal of Constructional Steel Research*, 65, 36-43.
- Shin, D.K, Gürdal, Z. ve Griffin, O.H. Jr., 1990. A penalty approach for nonlinear optimization with discrete design variables, *Engineering Optimization*, Vol. 16, pp. 29-42.
- Sonmez, M., 2011. Artificial Bee Colony algorithm for optimization of truss structures, *Applied Soft Computing*, 11, 2, 2406-2418.
- Sonmez, M., 2011. Discrete optimum design of truss structures using artificial bee colony algorithm, *Struct Multidisc Optim*, 43, 85-97.
- Sonmez, M., Aydın, E. ve Karabork, T., 2013. Using an artificial bee colony algorithm for the optimal placement of viscous dampers in planar building frames, *Struct Multidisc Optim*, DOI 10,1007/s00158-013-0892-y.
- Sonmez, M., Sevim, O. ve Kılıç, M., 2013. Topology optimization of double-curved double-layer grids, 2nd International Balkans Conference on Challenges of Civil Engineering, Tirane-Albania.

- Teodorovic, D., 2009. Bee colony optimization (BCO), In: Lim CP et al (eds) Innovations in swarm intelligence, SCI 248, Springer, Heidelberg, pp 39-60.
- Toakley, R., 1968. Optimum design using available sections, Journal of Structural Engineering, ASCE, Vol 94, No. 5, pp. 1219-1241.
- Togan, V., 2012. Design of planar steel frames using Teaching–Learning Based Optimization, Engineering Structures, 34, 225-232.

EK 1: İkinci örnek için Data sınıfını oluşturan metotlar ve sabitler

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Threading;
namespace SAPandABC
{
    public class Data
    {
        public int n;
        public int sn;
        public int mnc;
        public int limit;
        public double[] dv;
        public int [,] Limits;
        public int ndv;
        public double t = 0.25;
        public double E = 10000;
        public double allow_str = 25;
        public double max_disp = 2;
        public double density = 0.1;
        public string inputFileName;
        public double [] L;
        public double [] Area;
        public Data() . . .
    }
}
```

EK 2: İkinci örnek için Bee sınıfını oluşturan metotlar ve sabitler

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.IO;
namespace SAPandABC
{
    public class Bee : IComparable
    {
        public int ID;
        public int [] Dvar;
        public double [] vect_rnd;
        public double f;
        public int [] lower;
        public int [] upper;
        public double un_f;
        public bool feasible;
        public double age;
        public double g1;
        public double g2;
        public Bee() . . .
        public Bee(int Id, Data data, Random rnd) . . .
        public void show(Data data) . . .
        public void WritetoFile(Data data) . . .
        public Bee Copy(Bee a) . . .
        #region IComparable Members
        #endregion
    }
}
```

EK 3: İkinci örnek için Program sınıfını oluşturan metotlar ve sabitler

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Collections;
using System.Threading;
using System.Windows.Forms;
using SAP2000v15;
using SapModelV15;
namespace SAPandABC
{
    class Program
    {
        static public Random rnd;
        static public ArrayList Colony;
        static public SAP2000v15.SapObject SapObject;
        static public SAP2000v15.cSapModel SapModel;
        static void Main(string[] args)
        static Bee f(Bee a,Data data)
        public static void Show(ArrayList Colony,Data data)
    }
}
```

EK 4. LRFD' de bulunan 267 adet W kesiti

W44×335	W36×150	W27×84	W18×158	W14×159	W12×19
W44×290	W36×135	W24×492	W18×143	W14×145	W12×16
W44×262	W33×354	W24×408	W18×130	W14×132	W12×14
W44×230	W33×318	W24×335	W18×119	W14×120	W10×112
W40×593	W33×291	W24×279	W18×106	W14×109	W10×100
W40×503	W33×263	W24×250	W18×97	W14×99	W10×88
W40×431	W33×241	W24×229	W18×86	W14×90	W10×77
W40×372	W33×221	W24×207	W18×76	W14×82	W10×68
W40×321	W33×201	W24×192	W18×71	W14×74	W10×60
W40×297	W33×169	W24×176	W18×65	W14×68	W10×54
W40×277	W33×152	W24×162	W18×60	W14×61	W10×49
W40×249	W33×141	W24×146	W18×55	W14×53	W10×45
W40×215	W33×130	W24×131	W18×50	W14×48	W10×39
W40×199	W33×118	W24×117	W18×46	W14×43	W10×33
W40×174	W30×477	W24×104	W18×40	W14×38	W10×30
W40×466	W30×391	W24×103	W18×35	W14×34	W10×26
W40×392	W30×326	W24×94	W16×100	W14×30	W10×22
W40×331	W30×292	W24×84	W16×89	W14×26	W10×19
W40×278	W30×261	W24×76	W16×77	W14×22	W10×17
W40×264	W30×235	W24×68	W16×67	W12×336	W10×15
W40×235	W30×211	W24×62	W16×57	W12×305	W10×12
W40×211	W30×191	W24×55	W16×50	W12×279	W8×67
W40×183	W30×173	W21×201	W16×45	W12×252	W8×58
W40×167	W30×148	W21×182	W16×40	W12×230	W8×48
W40×149	W30×132	W21×166	W16×36	W12×210	W8×40
W36×848	W30×124	W21×147	W16×31	W12×190	W8×35
W36×798	W30×116	W21×132	W16×26	W12×170	W8×31
W36×650	W30×108	W21×122	W14×808	W12×152	W8×28
W36×527	W30×99	W21×111	W14×730	W12×136	W8×24
W36×439	W30×90	W21×101	W14×665	W12×120	W8×21
W36×393	W27×539	W21×93	W14×605	W12×106	W8×18
W36×359	W27×448	W21×83	W14×550	W12×96	W8×15

W36×328	W27×368	W21×73	W14×500	W12×87	W8×13
W36×300	W27×307	W21×68	W14×455	W12×79	W8×10
W36×280	W27×258	W21×62	W14×426	W12×72	W6×8.5
W36×260	W27×235	W21×57	W14×398	W12×65	W6×25
W36×245	W27×217	W21×50	W14×370	W12×58	W6×20
W36×230	W27×194	W21×44	W14×342	W12×53	W6×15
W36×256	W27×178	W18×311	W14×311	W12×50	W6×16
W36×232	W27×161	W18×283	W14×283	W12×45	W6×12
W36×210	W27×146	W18×258	W14×257	W12×40	W6×9
W36×194	W27×129	W18×234	W14×233	W12×35	W5×19
W36×182	W27×114	W18×211	W14×211	W12×30	W5×16
W36×170	W27×102	W18×192	W14×193	W12×26	
W36×160	W27×94	W18×175	W14×176	W12×22	

ÖZGEÇMİŞ

Adı Soyadı : Özer SEVİM

Doğum Yılı : 1988

Eğitim Bilgileri

Lisans : Selçuk Üniversitesi,
Mühendislik-Mimarlık Fakültesi,
İnşaat Mühendisliği Bölümü, 2011

İletişim Bilgileri

Adres : Aksaray Üniversitesi, İnşaat Mühendisliği

Telefon : 0382 288 23 46

E-posta : ozersevim@aksaray.edu.tr