

**KARADENİZ TEKNİK ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ**

**İNŞAAT MÜHENDİSLİĞİ ANABİLİM DALI**

**DEĞER KODLAMASI KULLANARAK KAFES SİSTEMLERİN GENETİK  
ALGORİTMA İLE MİNİMUM AĞIRLIKLI BOYUTLANDIRILMASI**

**139188**

**İnş. Müh. Tayfun DEDE**

**Karadeniz Teknik Üniversitesi Fen Bilimleri Enstitüsünde**

**“İnşaat Yüksek Mühendisi”**

**Ünvanı Verilmesi İçin Kabul Edilen Tezdir.**

**Tezin Enstitüye Verildiği Tarih : 07.08.2003**

**Tezin Savunma Tarihi : 27.08.2003**

**139188**

**Tez Danışmanı : Prof. Dr. Yusuf AYVAZ**

**Jüri Üyesi : Doç. Dr. Ayşe DALOĞLU**

**Jüri Üyesi : Doç. Dr. İsmail Hakkı ALTAŞ**

**Enstitü Müdürü : Prof. Dr. Yusuf AYVAZ**

**TRABZON 2003**

**T.C. KARADENİZ TEKNİK ÜNİVERSİTESİ  
DOKÜMANTASYON MERKEZİ**

## ÖNSÖZ

Bu çalışma Karadeniz Teknik Üniversitesi Fen Bilimleri Enstitüsü İnşaat Mühendisliği Anabilim Dalı'nda Yüksek Lisans Tezi olarak gerçekleştirilmiştir.

Yapay zekanın bir kolu olan genetik algoritmalarda gerçek değer kodlaması kullanılarak kafes sitemlerin minimum ağırlıklı olarak boyutlandırıldığı bu çalışmada süresince desteklerini ve yardımlarını esirgemeyen hocam Sayın Prof.Dr.Yusuf AYVAZ'a teşekkürlerimi sunmayı bir borç bilirim.

Değerli zamanlarını ayırarak tezimi değerlendiren hocalarım Sayın Doç.Dr.Ayşe DALOĞLU ile Sayın Doç.Dr.İsmail Hakkı ALTAŞ'a şükranlarımı sunarım.

Çalışma sürecinde bana yardım eden hocalarım ve tüm arkadaşlarıma teşekkür ederim.

Bugünlere gelmemi sağlayan ve bana destek olan aileme teşekkürümü bir borç bilirim.

Tayfun DEDE  
Trabzon 2003

## İÇİNDEKİLER

	<b><u>Sayfa No</u></b>
ÖNSÖZ.....	II
İÇİNDEKİLER.....	III
ÖZET.....	V
SUMMURAY.....	VI
ŞEKİLLER DİZİNİ.....	VII
TABLolar DİZİNİ.....	VIII
SEMBOLLER DİZİNİ.....	IX
1. GENEL BİLGİLER.....	1
1.1. Giriş.....	1
1.2. Yapı Optimizasyon Teknikleri.....	2
1.2.1. HÖRİSTİK Teknikler.....	2
1.2.1.1. Tabu Arama.....	3
1.2.1.2. Benzetilmiş Tavlama.....	3
1.2.1.3. Yapay Sinir Ağları.....	3
1.2.1.4. Genetik Algoritmalar.....	4
1.2.2. Genel Teknikler.....	6
1.2.2.1. Matematik Programlama Yöntemi.....	7
1.2.2.2. Optimumluk Kriteri Yöntemi.....	7
1.3. Çalışmanın Amaç ve Kapsamı.....	8
2. YAPILAN ÇALIŞMALAR, BULGULAR VE İRDELEME.....	9
2.1. Genetik Algoritma ile Yapı Optimizasyonu.....	9
2.1.1. Genetik Algoritmada Tasarım Değişkenlerinin Kodlanması.....	12
2.1.2. Genetik Algoritmada Başlangıç Neslinin Seçimi ve Önemi.....	16
2.1.3. Genetik Algoritmanın Özellikleri.....	17
2.1.4. Genetik Algoritmada Operatörler.....	17
2.1.4.1. Kopyalama Operatörü.....	18
2.1.4.2. Çaprazlama Operatörü.....	20
2.1.4.3. Mutasyon Operatörü.....	23

2.1.5.	Genetik Operatörlerle İlgili Öneriler .....	25
2.1.6.	Uygunluk Fonksiyonu .....	26
2.1.7.	Cezalandırılmış Amaç Fonksiyonu Değeri .....	26
2.1.8.	Genetik Algoritmada Toplum Nüfusu .....	27
2.2.	Çelik Çubukların Gerilme Tahkiki .....	27
2.2.1.	Çekmeye Çalışan Elemanlar .....	27
2.2.2.	Basınca Çalışan Elemanlar.....	28
2.3.	Düzlem ve Uzay Kafes Sistemlerin Yapısal Çözümlemesi .....	31
2.4.	Genetik Algoritmada Optimizasyon Süreci .....	35
2.5.	Kafes Sistemlerin Optimizasyon Örnekleri .....	38
2.5.1.	On Çubuklu Düzlem Kafes Sistem.....	38
2.5.1.1.	AISC-ASD'ye Göre On Çubuklu Sistemin Burkulmalı Tasarımı .....	39
2.5.1.2.	On Çubuklu Sistemin Burkulmasız Tasarımı .....	49
2.5.1.3.	On Çubuklu Sistemin TS-648' e Göre Burkulmalı Tasarım.....	51
2.5.2.	Dokuz Çubuklu Düzlem Kafes Sistem .....	52
2.5.3.	Yirmibeş Çubuklu Uzay Kafes Sistem .....	56
2.5.4.	Yetmişiki Çubuklu Uzay Kafes Sistem .....	59
2.5.5.	Elliki Çubuklu Uzay Kafes Sistem.....	63
3.	SONUÇLAR VE ÖNERİLER .....	67
4.	KAYNAKLAR.....	69
	ÖZGEÇMİŞ .....	72

## ÖZET

Genetik algoritma doğal seçim ilkesine dayanan bir optimizasyon yöntemidir. Diğer optimizasyon tekniklerinin aksine ayrık tasarım değişkenlerini kullanmaktadır. Genetik algoritma, çözüm dizilerinden oluşan bir başlangıç nesli ile kopyalama, çaprazlama ve mutasyon gibi doğal seçim operatörlerini kullanmaktadır.

Bu çalışmada genetik algortmada değer kodlaması kullanılarak kafes sistemlerin minimum ağırlıklı tasarımı amaçlanmıştır. Bu amaçla FORTRAN dilinde kodlanan program, yerdeğiştirme, gerilme ve stabilite sınırlayıcılarını içermektedir. Kafes sistemlerin analizinde Matris Deplasman Yöntemi kullanılmaktadır.

Bu çalışma dört bölümden meydana gelmektedir. İlk bölümde optimizasyon teknikleri ve çalışmanın amacı hakkında bilgi verilmektedir. İkinci bölümde literatürden alınan birkaç kafes sistem örneği genetik algoritma ile optimize edilmiştir. Elde edilen sonuçlar literatürdeki sonuçlarla karşılaştırılmıştır. Üçüncü bölümde ise bu çalışmadan çıkarılan sonuçlar ve bazı öneriler verilmektedir. Bu bölümü kaynaklar listesi izlemektedir.

Bu çalışmadan elde edilen sonuçlar, değer kodlaması kullanılarak hazırlanan programın kafes sistemlerin genetik algoritma ile minimum ağırlıklı olarak boyutlandırılmasında etkin bir şekilde kullanılabileceğini göstermektedir.

**Anahtar Kelimeler:** Minimum Ağırlıklı Boyutlandırma, Genetik Algoritma, Değer Kodlaması, Ayrık Tasarım Değişkenleri, Kafes Sistemler

## SUMMARY

### **Minimum Weight Design of Truss Structures Using Value Encoding in Genetic Algorithm**

Genetic algorithm is an optimization technique based on mechanism of natural selection. Unlike other optimization techniques, genetic algorithm uses discrete design variable. Genetic algorithm uses an initial population consisting of solution string and a set of natural selection operation such as reproduction, crossover and mutation.

The purpose of this study is to design the minimum weight truss structures by using value encoding in genetic algorithm. The genetic algorithm program coded in FORTRAN includes displacement, stress and stability constraints. In the analysis of truss structures, Matrix Analysis of Structures is used.

This study consists of four chapters. In the first chapter, some information about optimization technique and purpose of the study are given. In the second chapter, some truss structure examples taken from literature are solved by using genetic algorithm. Design results obtained in this study are compared with the design results given in the literature. In the third chapter, the conclusion drawn from this study and some recommendations are given. This chapter is followed by the list of references.

It is concluded that the program coded by using value encoding in genetic algorithm can be effectively used in the minimum weight design of truss structures.

**Keywords:** Minimum Weight Design, Genetic Algorithm, Value Encoding, Discrete Design Variables, Truss Structures

## ŞEKİLLER DİZİNİ

	<u>Sayfa No</u>
Şekil 1. Yapay sinir ağı modeli.....	4
Şekil 2. Genetik algoritma genel akış diyagramı.....	12
Şekil 3. On çubuklu kafes sistem.....	13
Şekil 4. İkilik kodlama sisteminde bireyin doğal bir kromozoma benzetilmesi.....	15
Şekil 5. Basınç etkisi altında çubuk elemanın davranışı .....	29
Şekil 6. Basınç elemanı için gerilme-burkulma grafiği.....	29
Şekil 7. Düzlem kafes elemanı için uç yer değiştirmeleri.....	31
Şekil 8. Uzay kafes elemanı için uç yer değiştirmeleri .....	32
Şekil 9. Sistem ve eleman eksen takımları .....	33
Şekil 10. On çubuklu düzlem kafes sistem ve yükleme durumu.....	38
Şekil 11. Birey ağırlığının iterasyon sayısı ile değişimi.....	47
Şekil 12. Dokuz çubuklu düzlem kafes sistem .....	52
Şekil 13. Optimum tasarıma sahip 7. bireyin iterasyona bağlı olarak cezalandırılmış amaç fonksiyonunun değişimi .....	55
Şekil 14. Ortalama cezalandırılmış amaç fonksiyonunun iterasyon sayısına bağlı olarak değişimi.....	55
Şekil 15. Yirmibeş çubuklu uzay kafes sistem.....	56
Şekil 16. Yetmişiki çubuklu uzay kafes sistem .....	60
Şekil 17. Elliiki çubuklu uzay kafes sistemin (a) üstten görünüşü, (b) yandan görünüşü.....	64

## TABLolar DİZİNİ

	<u>Sayfa No</u>
Tablo 1. Tasarım deęişkenleri olarak kullanılan enkesit alanları .....	13
Tablo 2. Tasarım deęişkenleri enkesit alanlarının kodlanması .....	15
Tablo 3. Bireylerin eşleşme havuzuna kopyalanması .....	18
Tablo 4. Eşleşme havuzunda bireylerin rasgele eşleşmesi.....	19
Tablo 5. Çaprazlama noktalarının belirlenmesi.....	21
Tablo 6. Tek noktalı çaprazlama.....	22
Tablo 7. İki noktalı çaprazlama .....	23
Tablo 8. On çubuklu sisteme ait en kesit alanları ve atalet yarıçapları.....	39
Tablo 9. On çubuklu sistem için genetik algoritmada başlangıç nesli.....	41
Tablo 10. On çubuklu sistem için yerdeęiştirmeler, uyum faktörü, kopya sayıları ve eşleşme havuzu .....	43
Tablo 11. On çubuklu sistem için çaprazlama, mutasyon ve yeni nesil oluşumu .....	45
Tablo 12. On çubuklu sistem için optimum tasarım sonuçları .....	46
Tablo 13. On çubuklu sistemin burkulmalı tasarım sonuçlarının karşılaştırılması .....	48
Tablo 14. On çubuklu sistemin burkulmasız tasarımı için kullanılan enkesit alanları .....	49
Tablo 15. On çubuklu sistemin burkulmasız tasarım sonuçlarının karşılaştırılması .....	50
Tablo 16. On çubuklu sistemin TS-648'e göre burkulmalı tasarım sonuçlarının karşılaştırılması .....	51
Tablo 17. Dokuz çubuklu sistemde tasarım için kullanılan enkesit alanları .....	53
Tablo 18. Dokuz çubuklu düzlem kafes sistemin tasarım sonuçlarının karşılaştırılması .....	54
Tablo 19. Yirmibeş çubuklu sisteme uygulanan kuvvetler .....	56
Tablo 20. Yirmibeş çubuklu sistem için gruplandırma sonuçları.....	57
Tablo 21. Yirmibeş çubuklu sistem için emniyet gerilmeleri .....	57
Tablo 22. Yirmibeş çubuklu sistemin tasarımında kullanılan enkesit alanları.....	58
Tablo 23. Yirmibeş çubuklu sistemin tasarım sonuçlarının karşılaştırılması.....	58
Tablo 24. Yetmişiki çubuklu sistem için yükleme durumları .....	61
Tablo 25. Yetmişiki çubuklu sistem için gruplandırma sonuçları .....	61
Tablo 26. Yetmişiki çubuklu sistemin tasarımında kullanılan enkesit alanları.....	62
Tablo 27. Yetmişiki çubuklu sistemin tasarım sonuçlarının karşılaştırılması.....	62



Tablo 28. Elliiki çubuklu sistemin tasarımında kullanılan tasarım deęişkenleri.....	65
Tablo 29. Elliiki çubuklu sistemin gruplandırılma sonuçları .....	66
Tablo 30. Elliiki çubuklu sistemin tasarım sonuçları.....	66



## SEMBOLLER DİZİNİ

A	: En kesit alanı
C	: Ceza fonksiyonu
$c_i$	: Sınırlayıcıların ihlal edilme katsayısı
E	: Elastisite modülü
F	: Dış yük vektörü
f	: Eleman eksen takımındaki yük vektörü
F <sub>c</sub>	: Uyum faktörü
F <sub>i</sub>	: Uyum derecesi
F <sub>ort</sub>	: Ortalama uyum derecesi
g <sub>i</sub>	: Sınırlayıcı fonksiyonu
I	: Atalet momenti
i	: Atalet yarıçapı
K	: Sistem rijitlik matrisi
K <sup>s</sup>	: Sistem eksen takımında eleman rijitlik matrisi
K <sup>e</sup>	: Eleman eksen takımında eleman rijitlik matrisi
L	: Elemanın boyu
lx	: Eleman x ekseninin sistem X eksenine yaptığı açının kosinüsü
ly	: Eleman y ekseninin sistem Y eksenine yaptığı açının kosinüsü
lz	: Eleman z ekseninin sistem Z eksenine yaptığı açının kosinüsü
m	: Sınırlayıcı sayısı
mx	: Eleman x ekseninin sistem X eksenine yaptığı açının kosinüsü
my	: Eleman y ekseninin sistem Y eksenine yaptığı açının kosinüsü
mz	: Eleman z ekseninin sistem Z eksenine yaptığı açının kosinüsü
n	: Güvenlik katsayısı
nx	: Eleman x ekseninin sistem X eksenine yaptığı açının kosinüsü
ny	: Eleman y ekseninin sistem Y eksenine yaptığı açının kosinüsü
nz	: Eleman z ekseninin sistem Z eksenine yaptığı açının kosinüsü
N	: Eksenel kuvvet
N <sub>kr</sub>	: Burkulmaya neden olan kritik eksenel kuvvet
P	: Ceza fonksiyonu katsayısı

$S_k$	: Burkulma boyu
$T$	: Dönüşüm matrisi
$U$	: Yerdeğiştirme vektörü
$u_{ix}$	: $i$ noktasının $x$ yönündeki yerdeğiştirmesi
$u_{jx}$	: $j$ noktasının $x$ yönündeki yerdeğiştirmesi
$v_{iy}$	: $i$ noktasının $y$ yönündeki yer eğiştirmesi
$v_{jy}$	: $j$ noktasının $y$ yönündeki yerdeğiştirmesi
$W$	: Ağırlık fonksiyonu
$w_{iz}$	: $i$ noktasının $z$ yönündeki yerdeğiştirmesi
$w_{jz}$	: $j$ noktasının $z$ yönündeki yerdeğiştirmesi
$\alpha$	: Eleman $x$ eksenini ile sistem $X$ eksenini arasındaki açı
$\beta$	: Eleman $y$ eksenini ile sistem $Y$ eksenini arasındaki açı
$\delta$	:Yerdeğiştirme
$\delta_u$	: Müsaade edilen yerdeğiştirme
$\phi$	: Cezalandırılmış amaç fonksiyonu
$\phi_{max}$	: Cezalandırılmış amaç fonksiyonunun maksimum değeri
$\phi_{min}$	: Cezalandırılmış amaç fonksiyonunun minimum değeri
$\gamma$	: Eleman $z$ eksenini ile sistem $Z$ eksenini arasındaki açı
$\lambda$	: Narinlik modülü
$\lambda_p$	: Plastik narinlik sınırı (kritik narinlik)
$\rho$	: Malzeme birim hacim ağırlığı
$\sigma$	: Gerilme
$\sigma_a$	: Akma gerilmesi
$\sigma_{bcm}$	: Basınç emniyet gerilmesi
$\sigma_{çem}$	: Çekme emniyet gerilmesi
$\sigma_e$	: Elastik gerilme
$\sigma_{kr}$	: Kritik gerilme
$\sigma_u$	: Müsaade edilen gerilme

## 1. GENEL BİLGİLER

### 1.1. Giriş

Yapı tasarımı optimizasyonu mühendislikte gittikçe önem kazanan vazgeçilmez bir konudur. Enerji, malzeme ve parasal kaynaklar kısıtlı olduğu için bir çok mühendislik dalında optimizasyona ihtiyaç duyulmaktadır. Örneğin, inşaat mühendisliğinde ekonomik ve bazı diğer sebeplerden dolayı yapı ağırlığının minimum olması istenmektedir. Bunun için birtakım optimizasyon teknikleri kullanılarak minimum ağırlıklı yapı elde edilmeye çalışılmaktadır. Ancak minimum ağırlığı bulmaya çalışırken yapı emniyeti de göz önünde bulundurulmalıdır.

Optimum tasarım düşüncesi 17. yüzyılda ortaya çıkmıştır. Bilgisayar teknolojisi sayesinde son 40 yılda yapı tasarımı üzerine bir çok optimizasyon teknikleri geliştirilmiştir. Yapı tasarımı üzerine günümüze kadar çeşitli optimizasyon algoritmaları oluşturulup, literatürde optimizasyonla ilgili yaklaşık olarak 150 kitap ve 2500'den fazla araştırma yazıları olduğu tahmin edilmektedir [1].

Genel olarak mühendislik tasarımında amaç, tasarımı gerçekleştirecek olan bilgisayar programı veya bir algoritma oluşturulduktan sonra optimum tasarımı belirlemektir. Optimum tasarım aranırken bilgisayar programlarına birtakım sınırlayıcılar ilave edilir. Sınırlayıcıları kullanmaktaki amaç optimizasyon probleminde istenmeyen çözümlerin oluşmasını önlemektir. Bu sınırlayıcılar, bir yapının ağırlığı, stabilitesi, kalitesi veya bu unsurların birleşimi şeklinde olabilir. Sınırlayıcılar problemin amacına ve tipine uygun olarak seçilmektedir [2]. Örneğin, bina türü bir yapının çatısını oluşturan çelik elemanlardan oluşan kafes sistemin optimizasyonu yapılırken çelik elemanların toplam ağırlığı, elemanlarda oluşacak gerilmeler veya sistemin düğüm noktalarının yerdeğiştirmeleri birer sınırlayıcı olarak kullanılabilir.

Çeşitli sebeplerden dolayı yapı tasarımı optimizasyonu üzerine araştırmalar halâ devam ettirilmektedir. Bu sebeplerden en önemli olanı mühendislerin sürekli olarak tasarım yapmalarıdır. Bu tasarımlar için gerekli olan zaman kısıtlıdır. Diğer taraftan, projenin başlatılmasından bitişine kadar geçen zaman ekonomiyile orantılıdır. Bu durum yapı tasarım optimizasyonunun, zaman bakımından ekonomiye olan etkisini göstermektedir.

Yapı tasarımı optimizasyon teknikleri 1960'lı yıllardan beri sürekli geliştirilmektedir. Ancak tüm yapı tasarımı problemleri için etkili, sağlam ve doğru çözüm veren bir yöntem tam olarak mevcut değildir [3]. Günümüzde arařtırmaların çoęu bu konu üzerine odaklanmaktadır.

## **1.2. Yapı Optimizasyon Teknikleri**

Yapı optimizasyon tekniklerini 2 ana gruba ayırabiliriz. 1) Geleneksel teknikler, 2) Hristik teknikler. Geleneksel teknikler matematiksel programlama ve optimumluk kriteri yntemleridir. Hristik teknikler ise Tabu arama, Benzetilmif Tavlama, Yapay Sinir Aęları ve Genetik Algoritmalarıdır. Bu alıřmada yapı optimizasyon tekniklerine kısaca deęinildikten sonra genetik algoritma teknięi detaylı olarak ele alınmaktadır.

### **1.2.1. Hristik Teknikler**

Optimizasyon probleminde, problemin amacı belirlendikten sonra bu problem birtakım matematiksel ifadelerle bilgisayarlar da kodlanırlar. Yapının fiziki zellikleri matematiksel anlatımlarla belirtilir. Ancak her optimizasyon probleminde, problem matematiksel ifadelerle anlatılamayabilir. Bu tr durumlarda bu problemlerin bilgisayarlar tarafından zlemeyeceęini sylemek doęru olmaz. Bu tr durumlar iin Hristik teknikler geliřtirilmiřtir.

Hristik teknikler herhangi bir problemi zmek iin geliřtirilmiř olan mantıksal yaklařımlardır [4]. Bunlar, kesin matematiksel ifadeler yerine, sadece problemin doęru sonuca gitmesini gerektirecek bir takım mantıksal yaklařımlar iermektedirler. Dolayısıyla bulunan sonuların en iyi zm olduęu iddia edilemez.

Hristik teknikler matematiksel ifadelere gre daha sade ifadeler ierir. Ancak zme ulařılırken daha fazla iterasyon yapar ve daha ok zaman kullanırlar. Bu yntemlerin kullanılması bilgisayarlar sayesinde hızlı bir řekilde artmaktadır. Hristik teknikler bilgisayar oyunlarında ve ticari yazılımlarda da kullanılmaktadır [4].

### **1.2.1.1. Tabu Arama**

Tabu kelimesi dokunulmaz yasaklı şey anlamına gelmektedir. Tabu aramalarının amacı, problemin çözümünde, her iterasyonda, problemin çözümünün istenmeyen sonuçlarına gitmesinden kaçınmaktır. Uygun bir başlangıç çözümü ile başlayan tabu araştırma algoritması bu uygun çözüme benzer çözümler ve uygun olmayan çözümler üreterek listeler. Herhangi bir iterasyonda daha önce ürettiği çözümleri rasgele kullanırken bu çözümlerin tabu listesinde kaydettiği uygun olmayan çözümlerden olmamasına dikkat eder. Seçtiği çözüm bir önceki çözümden daha iyi ise onunla yer değiştirilir [5, 6].

Tabu aramalarının diğer yöntemlerden en farklı özelliği problemin çözümünde kullanılan verileri kavrayabilme özelliğidir. Çözüm aşamasında herhangi bir uyumsuz veri, problemi iyi sonuçlara götürebilmesi için iyi ipuçları verebilir. Tabu aramaları, tabu listelerindeki verileri diğer yöntemlerden (Genetik Algoritma, Yapay Sinir Ağları, v.b) alabilir [7]. Tabu aramalarının verileri kavrayabilme özelliği, problemin çözümünde daha etkili ve ekonomik araştırma olanağı sağlar.

### **1.2.1.2. Benzetilmiş Tavlama**

Benzetilmiş tavlama, bir başlangıç verisiyle problemi optimize etmeye başlayıp iterasyonlar yaparak amaç fonksiyonunu değerlendirir. Amaç fonksiyonu minimize edilirken herhangi bir iterasyon adımı referans olarak alınır. Daha sonra bu iterasyondan itibaren tekrar optimize etme süreci başlatılır. Böylece iterasyonlarda sonuca erken yakınsamanın önüne geçilmiş olunur [8]. İlerleyen iterasyonlar referans olarak alındığı için fazla iterasyon sayısı oluşmadan global optimum çözümlere yakınsama sağlanır.

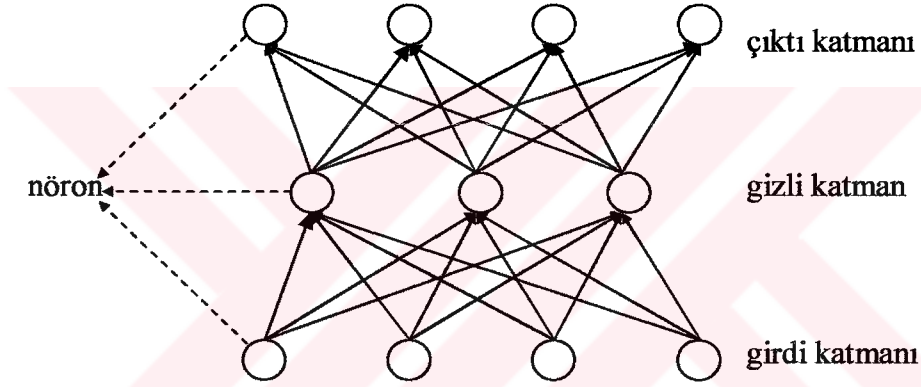
Benzetilmiş tavlama amaç fonksiyonuyla ilgili çok az varsayım yaptığı için sonuçlar güvenilirdir. Ancak sonuçların güvenilirliği problemin hassasiyetine göre değişmektedir. Benzetilmiş tavlama çok zor problemlerle lokal optimize edici olarak kullanılabilir [8].

### **1.2.1.3. Yapay Sinir Ağları**

Yapay sinir ağları, insan beyninin biyolojik sistemini taklit eden esaslara dayalı bir işletim sistemidir. Yapay sinir ağlarına son yıllarda yoğun bir ilgi duyulmaktadır. Çoğu

yapay sinir ağı uygulaması geri yayılım yöntemine dayanmaktadır. Geri yayımlı sinir ağı, yapay nöronlar adı verilen birbiriyle bağlantılı çok sayıda elemandan oluşmaktadır. Elemanlar 2 yada daha fazla tabakaya mantıklı bir şekilde dizilmekte ve birbirlerini belirlenen bir ağırlıktaki işbirliği ile etkilemektedirler. Bu ölçütsel ağırlıklar birbiriyle bağlantılı olan elemanların arasındaki etkinin çeşidini ve kuvvetini bildirmektedirler [4].

Şekil 1’de görüldüğü gibi 3 katman birbiriyle bağlantılıdır. Her katmandaki elemanlar bir sonraki katmanın her elemanı ile bağlantılıdır. Bu 3 katmandan biri girdi katmanı, diğeri gizli katman ve üçüncüsü ise çıktı katmanıdır [4, 9]. Gizli katmanda veriler arasındaki ve verilerle çıktılar arasındaki ilişkiler değerlendirilmektedir. Gizli katman sayısı problemin türüne bağlı olarak belirlenmektedir.



Şekil 1. Yapay sinir ağı modeli

Genel olarak, doğru etkileşim ağırlıklarının belirlenmesinde geri öğrenme işlemi [9] kullanılmaktadır. Verilerle çıktılar arasında uygun bir işbirliğinin kurulabilmesi için ağ eğitilmelidir. Ağ bir kez eğitildiği zaman bundan sonra verilen verilerden arzu edilen çıktılara ulaşmak için hızlı bir ağ çizelgesi oluşur.

#### 1.2.1.4. Genetik Algoritmalar

Genetik algoritmalar yapay zekanın gittikçe genişleyen bir kolu olan evrimsel hesaplama tekniğinin bir parçasını oluşturmaktadır. Adından da anlaşılacağı üzere evrimsel hesaplama olan genetik algoritma Darwin’in evrim teorisinden esinlenerek oluşturulmuştur [10].

Herhangi bir problemin genetik algoritmayla çözümü, problemi sanal olarak evrimden geçirmek suretiyle yapılmaktadır [11].

Evrimsel hesaplama ilk olarak 1960'lı yıllarda I. Rechenberg tarafından "Evrimsel Stratejileri" isimli eserinde tanıtılmıştır. Onun fikri daha sonra başka araştırmacılarında ilgisini çekmiş ve geliştirilmiştir. John Holland, evrim sürecinin bir bilgisayar yardımı kullanılarak bilgisayara anlayamadığı çözüm yöntemlerini öğretilbileceğini düşünmüştür [11]. Genetik algoritma böylece John Holland tarafından bu düşüncenin sonucu olarak bulunup onun öğrencileri ve arkadaşları tarafından geliştirilmiştir. 1992 yılında John Koza genetik algoritmayı kullanarak çeşitli görevleri yerine getiren programlar geliştirip bu yönteme genetik programlama adını verdi.

Genetik algoritma, evrim mekanizmasını kullanan bir stratejidir [12]. Genetik algoritmanın esası en iyinin hayatta kalmasına ve adaptasyonuna dayanır. Ancak diğer yöntemlerde de olduğu gibi genetik algoritma en iyi çözümü vereceğini garanti edemez. Buna karşın genetik algoritmaların bulduğu çözümler diğer yöntemlerin çözümüne göre daha pratiktir.

Genetik algoritmalar, problemin amaç fonksiyonu için fazla bilgiye gereksinim duymazlar. Probleme çözüm olabilecek bireylerden oluşan bir topluluk üzerinde bir takım operatörler gerçekleştirilerek sonuç bulur. Topluluk içerisindeki her birey bir seri karakterin oluşturduğu dizilerle ifade edilir. Bu karakterler genellikle ikilik kodlama sisteminde 1 ve 0 tamsayılarından oluşmaktadır.

Başlangıç topluluğunu oluşturan bireyler bilgisayar yardımıyla rasgele seçilir. Basit olarak genetik algoritma 3 kısımdan oluşur. Birinci kısım tasarım değişkenlerinin kodlanması, ikinci kısım her bir bireyin uyumluluğunun araştırılması ve üçüncü kısım ise yeni çözümler bulmak için genetik operatörlerin uygulanmasıdır. Bir bireyin uyumluluğu, bireyin amaç fonksiyonu ve sınırlayıcılar arasındaki dengesiyle ifade edilir. Yani, birey amaç fonksiyonu için çalışırken sınırlayıcıları ihlal etmemelidir.

Genetik algoritmalar başlıca 3 operatörden oluşur. Bunlar, kopyalama, çaprazlama ve mutasyon operatörleridir. Bu operatörler problemin hassasiyetine göre belirli bir oranda kullanılmaktadır.

Kopyalama işlemi uyumluluğu yüksek olan bireylerin, oluşturulacak olan yeni topluluğa değiştirilmeden aktarılmasıdır. Bu düşünce Darwin'in doğal seçim ilkesinden esinlenerek ortaya atılmıştır [10]. Kopyalamadan sonraki adım çaprazlama ve mutasyon



operatörlerini, kopyalamayla oluşturulan topluluğa uygulayarak yeni bir topluluğu oluşturmaktır.

Çaprazlama işlemi, belirli bir oranda uygulanarak, topluluktaki bireyler yerine yeni bireyler yaratmak için kullanılırlar. Bunun için toplumdaki bireyler rasgele eşleştirilir. Eşlerin çaprazlama operatörünün tipine göre bir veya daha fazla parçası karşılıklı olarak değiştirilir. Değişim sonucu oluşan yeni birey eski bireyin çocukları olarak tanımlanır. Çaprazlama işleminden sonra topluluktaki birey sayısının değişmemesine dikkat edilmelidir. Çünkü amaç birey sayısını artırmak yada azaltmak değil yeni bireylerin topluma kazandırılmasıdır.

Çaprazlama işleminden sonra mutasyon operatörü topluluktaki bireylere uygulanır. Literatürde, mutasyon oranının düşük olması vurgulanmıştır. Çünkü mutasyon, topluluktaki bireylerin çözüme erken gitmesini önlerken hiç çözüm bulunmamasına da sebep olabilmektedir. Mutasyon, bireyi oluşturan tasarım değişkenlerinin rasgele değiştirilmesidir. Mutasyon sayesinde iyi uyumlu bir birey elde edilirken, çözüme yakınsayan iyi uyumlu bir birey de topluluktan atılabilir. Bu durum mutasyonun en kötü dezavantajıdır.

Literatürde genetik algoritmada tasarım değişkenleri genellikle ikilik kodlama sistemiyle yapılmıştır. Ancak ikilik kodlamanın dışında değer kodlaması, permütasyon kodlaması ve ağaç kodlaması da mevcuttur [13]. Bu çalışmada değer kodlaması kullanılmaktadır. Kodlama türünün probleme göre seçilmesi diğer operatörler için kolaylık sağlamaktadır. Değer kodlaması için operatörlerin uygulanması ikilik kodlamadaki gibi yapılabilir [13].

### **1.2.2. Genel Teknikler**

Geleneksel yapı optimizasyon teknikleri iki kısma ayrılır. Bunlardan birincisi matematik programlama diğeri ise optimumluk kriteri yöntemidir. Bu iki yöntem literatürde sıkça rastlanmaktadır.

### 1.2.2.1. Matematik Programlama Yöntemi

Matematik programlama belirli sayıda değişkene sahip olan bir problemin, bir takım sınırlayıcıları kullanarak, optimum çözümünün aranmasıdır. Matematik programlama genel olarak lineer programlama ve lineer olmayan programlama olmak üzere ikiye ayrılır. Lineer programlamada amaç fonksiyonu ve sınırlayıcılar, tasarım değişkenlerinin lineer bir kombinasyonu olarak ifade edilir. Lineer programlama ilk kez Foulkes tarafından minimum ağırlık problemi için kullanılmıştır [1].

Matematik programlar ilk zamanlar hesaplamaların çok zaman almasından dolayı araştırmacılar tarafından pek fazla ilgi görmemiştir [14]. Gellatly ve Berke optimumluk kriteri yönteminin matematik programlamaya göre daha etkin olduğunu vurgulamışlardır [1]. Matematik programlama aşağıda belirtilen sebeplerden dolayı etkin şekilde çalışmamaktadır.

- Problem içinde çok miktarda tasarım değişkeni vardır.
- Çok sayıda sınırlayıcı mevcuttur.
- Sınırlayıcıları değerlendirmek için çok fazla hesap ve zaman gerekmektedir.

Yukarıda bahsedilen zorlukları yenmek için Schmit ve Farshi bir takım yaklaşık kabuller yapılmasını önermişlerdir [1]. Böylece tasarım değişkenlerinin ve sınırlayıcıların sayısı azaltılmış oldu. Dolayısıyla da hesap hacmi azalıp gereken zaman da azalmış olmaktadır.

### 1.2.2.2. Optimumluk Kriteri Yöntemi

Optimumluk kriteri yöntemi global optimizasyon için türev bilgilerini ve yeterince özel koşulları kullanan bir optimizasyon yöntemidir. Optimumluk kriteri tasarım değişkenlerinin sürekli olması esasına dayanır. Bu yöntem yapı elemanlarının tek bir enkesit özelliğini kullanmaktadır. Yapı elemanının diğer özellikleri, seçilen enkesit tasarım değişkeninin bir fonksiyonu olarak ifade edilirler [4].

Bu yöntem yapının fiziki özelliklerini hesaba katarak yapının davranışı ile ilgili bir kriter belirler. Yapı bu kriteri sağlayacak şekilde boyutlandırıldığında tasarım için de en

uygun boyutlar verilmiş olmaktadır. Optimumluk kriteri yöntemi, mühendislik tasarımlarında yaygın olarak kullanılmaktadır.

### **1.3. Çalışmanın Amaç ve Kapsamı**

Bu tezin hazırlanmasındaki amaç düzlem ve uzay kafes sistemlerinin minimum ağırlıklı tasarımını gerçekleştirmektir. Bu çalışmada ayırık değişkenli çelik profiller kullanılmıştır. Çelik çubuk elemanlarında aksenal basınç ve burkulma hesapları yapılmıştır.

Bu çalışmada literatürde alışlagelmiş yöntem olan ikilik kodlama yerine, genetik algoritmada değer kodlamasıyla tasarım programları yazılmaktadır. Genetik algoritmada, değer kodlaması sistemine literatürde pek rastlanılmamaktadır.

Kafes sistemlerin çözümü için Matris Deplasman Yöntemi kullanılmıştır. Elde edilen sonuçlar literatürdekilerle karşılaştırılmıştır.

## 2. YAPILAN ÇALIŞMALAR, BULGULAR VE İRDELEME

### 2.1. Genetik Algoritma ile Yapı Optimizasyonu

Bugüne kadar yapı sistemlerinin optimizasyonu için bir çok yöntem geliştirilmiştir. Bu yöntemlerin büyük çoğunluğu tasarım değişkenlerinin sürekli olması esasına dayanır [15]. Oysa mühendislikte pratik problemlerde tasarım değişkenleri ayrık değişkenlidir [16,17]. Çünkü yapıyı oluşturan elemanlar genelde standart olarak mevcuttur. Örneğin, çelik profiller belirli standartlarda üretilirler. Dolayısıyla optimizasyonlardan bulunan optimum tasarım değişkenlerinin pratikte mevcut olması istenir. Genetik algoritma ayrık tasarım değişkenlerini kullandığı için bulunan optimum tasarım değişkenleri hiç değiştirilmeden pratikte kullanılabilir.

Genetik algoritma Darwin'in evrim teorisini dikkate alarak optimize edilecek problemi sanal bir evrim sürecinden geçirir. Bu süreci gerçekleştirirken yine doğadaki evrim sürecini taklit ederek bir takım operatörler kullanılır. Basit genetik algoritmada bu operatörler, seçim (kopyalama), çaprazlama ve mutasyondur. Bu operatörlere ilerleyen bölümlerde açıklık getirilecektir. Aşağıda genetik algoritmada kullanılan bir takım terimler verilmektedir.

- **Birey:** Literatürdeki genel adı kromozomdur. Problemin çözümü olabilecek tasarım değişkenlerinin belirli sıraya göre dizilmesinden oluşmaktadır.
- **Ebeveyn birey:** Genetik operatörlere uğrayıp yeni birey (çocuk birey) oluşturabilen herhangi bir bireydir.
- **Topluluk:** Literatürdeki genel adı popülasyondur. Belirli sayıda bireyin bir araya gelmesinden oluşmaktadır.
- **Nesil:** Daha iyi ebeveyn bireyler elde etmek için belirli sayıda bireyin katılımıyla genetik algoritmada oluşturulan her hangi bir toplumdur.
- **Uygunluk:** Tasarımın performansının ölçüsüdür.
- **Genetik operatörler:** Ebeveyn bireyler arasında bilgi iletişimini sağlayan planlardır.

Genetik algoritmada bir bireyin görevi, genetik doğadaki bir kromozoma benzetilebilir. Bireyler literatürde genellikle ikilik kodlama sistemiyle oluşturulmuştur. Bu

çalışmada ise değer kodlaması daha uygun görülmüştür. Örneğin, 30 tasarım değişkenine ve 6 elemana sahip bir optimizasyon probleminde bir birey 5 8 30 19 21 3 şeklinde ifade edilebilir. Burada 5, yapıyı oluşturan birinci elemanın 5 no'lu tasarım değişkeninden oluştuğunu gösterir. Diğer sayılar 8, 30, 19, 21 ve 3 sırasıyla yapıyı oluşturan 2., 3., 4., 5. ve 6. elemanların 8, 30, 19, 21 ve 3 no'lu tasarım değişkenlerinden oluştuğunu gösterir. Bu örnekte yapının toplam 6 elemandan oluştuğu göz önüne alınmıştır. Değer kodlama sisteminde bir bireyin dizi uzunluğu yapıyı oluşturan eleman sayısı kadardır. Oysa ikilik kodlama sisteminde dizi uzunluğu aynı örnek için daha fazla olacaktır. Çünkü değer kodlamasında herhangi bir bireyin bir elemanı tek bir tamsayı ile temsil edilebilirken ikilik kodlamada bu elemanı temsil etmek için 1 ve 0 tamsayılarından oluşan belirli uzunluktaki dizi kullanılmaktadır.

Genetik algoritma, optimizasyon sürecine başlayabilmek için bir başlangıç nesline gerek duyar. Bu başlangıç nesli bilgisayar tarafından rasgele belirlenir. Başlangıç neslinin kötü bireylerden oluşması durumunda optimum tasarımı bulmak için bilgisayar tarafından yapılacak iterasyon sayısı (nesil sayısı) çok fazla olacaktır. Bu yüzden başlangıç neslinin iyi seçilmesi önemlidir.

Başlangıç nesli belirlendikten sonra tüm bireylerin uygunluk fonksiyonları belirlenir. En yüksek uyumluluğa sahip bireyler daha iyi bir nesil oluşturmak üzere kopyalanırlar. Uyumluluğu düşük olan bireyler topluluktan dışarı atılırlar. Yani, Darwin'in teorisine göre iyi bireyler yaşar, kötü bireyler ölür. Topluluktaki birey sayısının sabit kalması amacıyla, kopyalanacak birey sayısı topluluktan atılan birey sayısı kadardır. Kopyalama çeşitli şekillerde olur. Literatürde geleneksel olarak en iyi bireyden bir tane, daha sonra ikinci iyi bireyden bir tane olmak üzere topluluktan atılan birey sayısınca kopyalama devam ettirilmesi önerilmektedir. Kopyalamanın amacı iyi bireylerin saklanıp nesiller boyunca devam ettirilmesidir. Böylece iyi ebeveyn bireylerden iyi çocuk bireyler oluşacağı ve oluşacak her yeni neslin bir önceki nesle göre daha iyi olacağı umulmaktadır.

Bir sonraki nesle kopyalanan bireyler eşleşme havuzu denilen yerde toplanırlar. Eşleşme havuzunda her bir birey rasgele olarak başka bir bireyle eşleştirilir. Birey kendi kopyasıyla tesadüfen eşleştirilmiş olabilir. Bu durumda oluşacak yeni birey eski bireyin aynısı olacaktır.

Eşleşme havuzundaki her bir birey çifti çaprazlama noktası denilen kısımlardan parçalanırlar. Bu parçalar eşler arasında karşılıklı olarak değiştirilir. Bu işleme genetik algoritmada çaprazlama adı verilir. Parçalar belirli sayıda tasarım değişkeni içerirler.

Parçaların hangi kurallara göre belirlenip çaprazlamaya tabii tutulacağı, çaprazlama operatörü tarafından tayin edilir. Literatürde birkaç çeşit çaprazlama yönteminden bahsedilmektedir. Bunlar kısım 2.1.4.2.'de açıklanmaktadır.

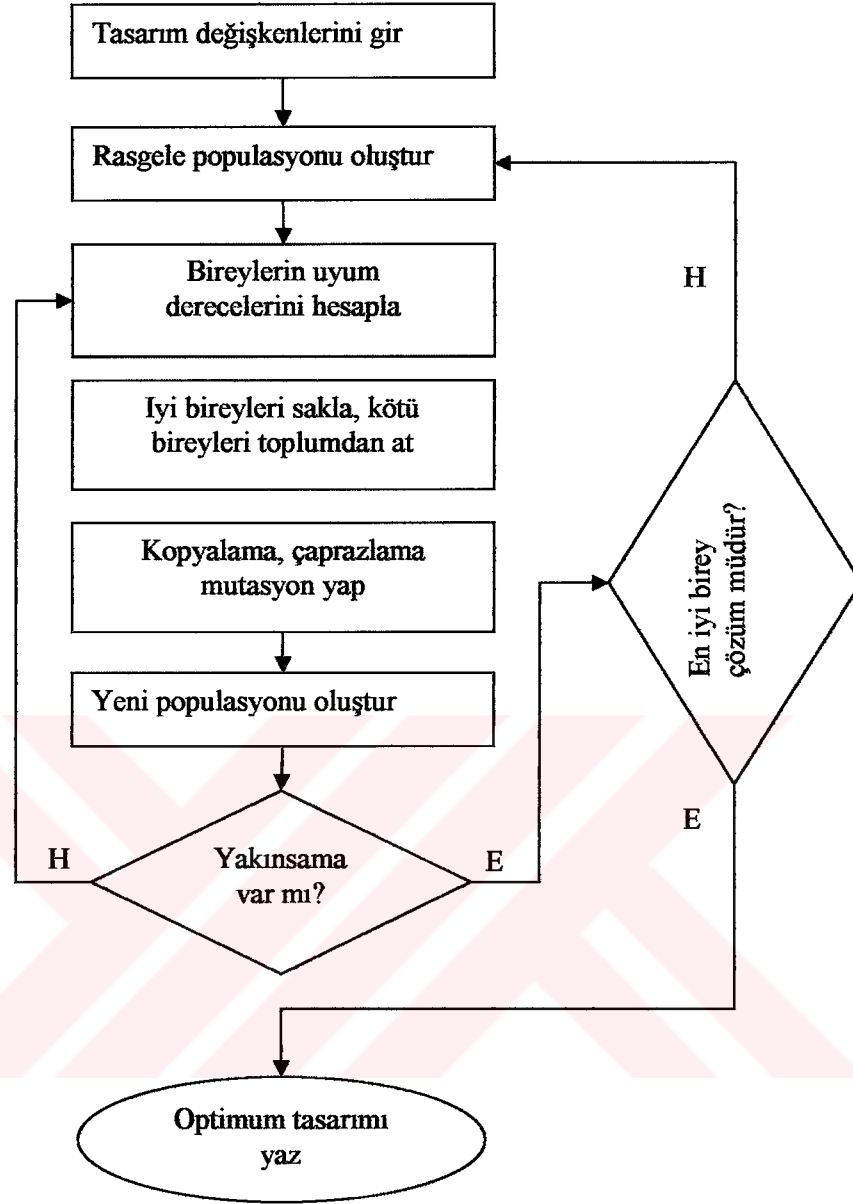
Toplumdaki tüm bireyler çaprazlandıktan sonra, toplam birey sayısı sabit kalmak şartıyla, yeni çocuk bireyler toplumu oluşturur. Bu yeni toplum içerisinde en uygun bireyi, yani optimum tasarımı aramadan önce yeni bireyler belirli bir oranda mutasyona uğratılır. Mutasyon, tasarım değişkenlerinin rasgele değiştirilmesiyle yapılır.

Mutasyon operatörü de devreye girdikten sonra eski bireylerden yeni bir nesil oluşturulmuş olur. Oluşan bu yeni nesil eski nesillere göre daha iyidir. Çünkü uygunluk fonksiyonu, nesillerdeki kötü uyumlu bireyleri topluluktan dışarı atmaktadır. Böylece oluşan her yeni nesil bir önceki nesilden daha uyumlu olmaktadır. Yani nesil sayısı arttıkça optimum tasarıma yaklaşılr.

Genetik algoritma, toplum üzerinde yaptığı operatörler ve değerlendirmelerden sonra eğer bireylerin çoğu aynı ise optimum tasarımı bulmuş sayılır. Bunun için bir yakınsama yüzdesi kullanıcı tarafından belirlenir. Örneğin %75'lik bir yakınsama dikkate alınmışsa, genetik algortmada nesillerde oluşan bireylerin %75'i aynı bireyden oluştuğu takdirde algoritma durdurulmaktadır.

Genetik algortmada yakınsama oranı yakalandığı halde her zaman optimum tasarım elde edilmiştir denilemez. Çünkü yakınsama oranının sağlandığı nesil uyum derecesi çok düşük bireylerden oluşmuş bir topluluk olabilir. Bu durumda başlangıç neslini değiştirip yeni bir topluluk oluşturmak yoluna gidilir.

Genetik algoritma her ne kadar optimum tasarımı bulacağını garanti edemese de, çoğunlukla diğer yöntemlere göre daha pratik sonuçlar vermektedir. Bu çalışmada çelik profillerin enkesit alanları tasarım değişkenleri olarak alınmıştır. Tasarım değişkenleri değer kodlama sisteminde kodlanmaktadır. Böylece bireyler, bireylerden de topluluklar oluşturularak genetik algoritma süreci başlatılmaktadır. Genetik algoritma için bir akış diyagramı en basit hali ile Şekil 2'de verilmektedir.



Şekil 2. Genetik algoritma genel akış diyagramı

### 2.1.1. Genetik Algoritmada Tasarım Değişkenlerinin Kodlanması

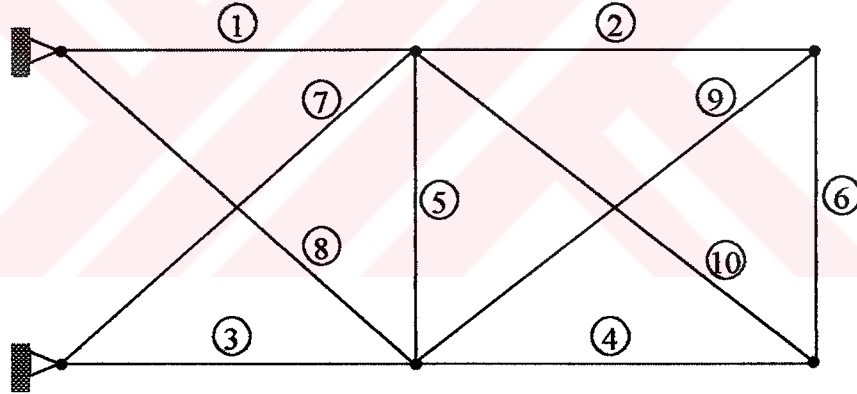
Kodlama işlemi tasarım değişkenlerinin genetik algoritma tarafından tanınabilmesi için yapılmaktadır. Bu çalışmada tasarım değişkenleri olarak çelik profillerin enkesit alanları kullanılmıştır. Literatürde daha öncede belirtildiği gibi ikilik kodlama, permütasyon kodlaması, değer kodlaması ve ağaç kodlaması olmak üzere 4 kodlama türüne rastlanmaktadır. Seçilecek olan kodlama türü işlemlerin kolaylığı açısından problem



türüne bağlı olarak belirlenmektedir. Bu çalışmada kullanılan değer kodlamasının uygulaması şu şekildedir .

Optimizasyon problemi için seçilen tasarım değişkenlerine sıra numarası verilir. Tasarım değişkenleri algoritma tarafından kullanılacağı zaman bu sıra numarasıyla çağrılırlar. Sıra numaraları onluk sistemle kodlanır. Sıra numaraları onluk sistemde kodlandığı için bir bireyi oluşturan dizi uzunluğu ikilik kodlama sistemine göre oldukça küçüktür. Diziyi oluşturan tamsayıların sayısı sistemi oluşturan yapı elemanı sayısı kadardır. Dizi uzunluğundaki bu küçülme bilgisayar açısından, büyük dizi boyutlarıyla uğraşmadığı için, zaman tasarrufu sağlayıp her açıdan işlem hacmini azaltmaktadır.

Şekil 3’de tasarım değişkenlerinin değer kodlamasıyla kodlanmasını göstermek üzere 10 çubuktan oluşan düzlem kafes sistem örneği verilmektedir. Tasarım değişkeni olarak 20 adet çelik profil enkesit alanı dikkate alınmıştır. Tasarım değişkeni olarak kullanılan enkesit alanları sırasıyla Tablo 1’de verilmektedir.



Şekil 3. On çubuklu kafes sistem

Tablo 1. Tasarım değişkenleri olarak kullanılan enkesit alanları

Sıra no	Enkesit alanı (cm <sup>2</sup> )	Sıra no	Enkesit alanı (cm <sup>2</sup> )	Sıra no	Enkesit alanı (cm <sup>2</sup> )
1	10,45	8	24,71	15	66,45
2	13,74	9	30,19	16	75,48
3	14,26	10	35,74	17	87,10
4	15,16	11	39,74	18	90,97
5	17,10	12	45,68	19	99,35
6	18,97	13	53,23	20	110,32
7	20,52	14	58,90		



Burada tasarım deęişkenlerinin sayısı 20 olduęu için her hangi bir enkesit alanı deęişkeninin kodu tamsayı olarak en fazla 20 deęerini alır. Örneęin, 16 denildięi zaman program, sıra numarası 16 olan tasarım deęişkenini, 20 denildięi zaman ise program sıra numarası 20 olan tasarım deęişkenini çağırılmaktadır.

Şekildeki kafes sistemde daire içindeki sayılar eleman numaralarını göstermektedir. Bu sistemde 1. elemanın enkesit alanı  $53,23 \text{ cm}^2$ , 2. elemanın  $30,19 \text{ cm}^2$ , 3. elemanın  $110,32 \text{ cm}^2$ , 4. elemanın  $90,97 \text{ cm}^2$ , 5. elemanın  $10,45 \text{ cm}^2$ , 6. elemanın  $30,19 \text{ cm}^2$ , 7. elemanın  $66,45 \text{ cm}^2$ , 8. elemanın  $99,35 \text{ cm}^2$ , 9. elemanın  $75,48 \text{ cm}^2$  ve 10. elemanın  $110,32 \text{ cm}^2$  enkesit alanına sahip olduęunu düşünürsek sistemin çözümü olabilecek tasarım deęişkenlerini yan yana yazarak oluşturulacak dizi şu şekilde olmaktadır.

Birey : 53,23 30,19 110,32 90,97 10,45 30,19 66,45 99,35 75,48 110,32

Enkesit alanlarının deęerleri yerine sıra numaraları yazarak ařaęıdaki gibi dizi daha pratik duruma getirilebilmektedir.

Birey : 13 9 20 18 1 9 15 19 16 20

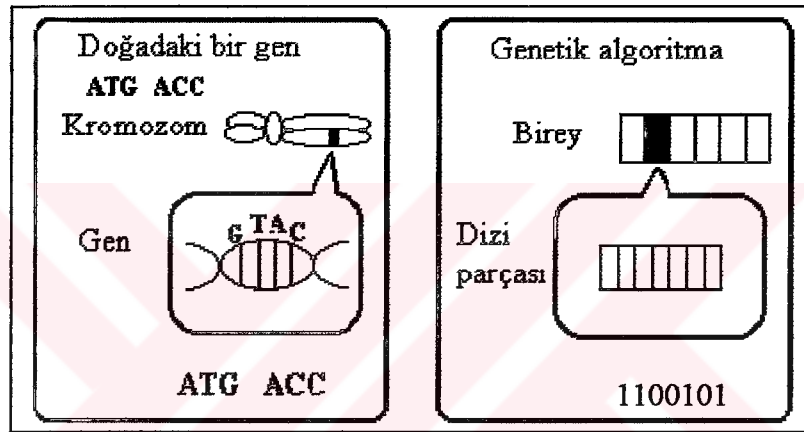
Bu diziden görüldüęü gibi dizi uzunluęu (enkesit alanı sayısı) 10'dur. Bu sayı sistemdeki eleman sayısına eşittir. Bu özellik deęer kodlamasının dięer kodlama sistemlerine göre en büyük avantajıdır.

İkilik kodlama sisteminde aynı dizi oluşturulmak istendięinde dizi uzunluęu artacaktır. Çünkü bir tasarım deęişkenini deęer kodlamasında tek bir tamsayı ile kodlarken, ikilik kodlama sisteminde aynı tasarım deęişkeninin kodlanması birden fazla tamsayının yan yana dizilmesiyle oluşmaktadır. Örneęin, bir tasarım deęişkeni 20 farklı deęer alsın. Bu 20 deęerden 17. herhangi bir tasarım deęişkenini temsil etsin. Bu durumda tasarım deęişkeninin deęer kodlama sistemindeki kodlanmış hali 17 iken, aynı tasarım deęişkeninin ikilik kodlama sistemindeki kodlanma şekli 10000 şeklinde olacaktır. Ařaęıda ikilik kodlamada bir dizinin şifresinin çözülmesi gösterilmektedir.

$$\begin{aligned} 10000 &= 2^0 \times 0 + 2^1 \times 0 + 2^2 \times 0 + 2^3 \times 0 + 2^4 \times 1 + 1 \\ &= 0+0+0+0+16+1 \\ &= 17 \end{aligned}$$

Tek bir deęişkeni temsil eden 10000 dizisi ikilik sistemde çözümlenerek oluşan sayıya 1 katılmasıyla onluk sistemde tasarım deęişkenine ait sıra numarası elde edilmektedir.

Şekil 4’de ikilik kodlama sisteminde bireylerin doğal kromozomlara benzetilmesi gösterilmektedir. Bireyler kromozomlara, bireyin dizisini oluşturan alt dizi parçaları genlere, alt diziyi oluşturan 1, 0 rakamları ise bir geni oluşturan ATG, ACC gibi biyolojik şifrelere benzetilerek genetik algoritmada bir problemin çözümü olabilecek bireyler (çözüm dizileri) oluşturulmaktadır. Tablo 2’de 10 çubuklu sistemde ikilik ve deęer kodlaması kullanıldığında ilk 10 tasarım deęişkeninin kodlanması gösterilmektedir.



Şekil 4. İkili kodlama sisteminde bireyin doğal bir kromozoma benzetilmesi

Tablo 2. Tasarım deęişkenleri enkesit alanlarının kodlanması

Sıra no	Enkesit alanı (cm <sup>2</sup> )	Kodlama türü	
		İkili	Deęer
1	10,45	00000	1 (10,45)
2	13,74	00001	2 (13,74)
3	14,26	00010	3 (14,26)
4	15,16	00011	4 (15,16)
5	17,10	00100	5 (17,10)
6	18,97	00101	6 (18,97)
7	20,52	00110	7 (20,52)
8	24,71	00111	8 (24,71)
9	30,19	01000	9 (30,19)
10	35,74	01001	10 (35,74)

Yukarıdaki 10 çubuklu sistem incelendiğinde ikilik kodlama sisteminde bir tasarım değişkeninin 5, tüm elemanların ise 50 adet tamsayının yan yana dizilmesiyle elde edildiği görülmektedir. Oysa aynı örnek için değer kodlamasında tüm elemanlar 10 adet sayı ile kodlanabilmektedir. Böylece dizi boyutları oldukça küçültülmüş olmaktadır. Bu örnek için tasarım değişkenlerinin sayısı en fazla 20 olduğu için ikilik kodlamada tek bir tasarım değişkeni 5 karakterden oluşmaktadır. Ancak başka problemlerde tasarım değişkenlerinin fazla olması durumunda ikilik kodlamada dizi uzunluğu daha da artarken değer kodlamasında dizi uzunluğu sistemdeki eleman sayısı veya grup sayısı ile sınırlı olmaktadır.

### **2.1.2. Genetik Algoritmada Başlangıç Neslinin Seçimi ve Önemi**

Genetik algoritmanın optimizasyon sürecine başlayabilmesi için bir başlangıç nesline ihtiyaç duyduğu daha önce belirtilmişti. Bu başlangıç nesline daha sonraki nesillerde bir takım operatörler uygulanarak optimum tasarım aranmaktadır. Genetik algoritmanın etkin bir araştırma yapabilmesi için araştırma alanının geniş tutulması gerekmektedir. Yani, bütün tasarım değişkenleri program tarafından değerlendirmeye alınabilmelidir. Bunun için ya bireyi oluşturan dizi uzunluğu fazla olmalıdır ya da başlangıç neslini oluşturan birey sayısı yeterince fazla olmalıdır.

Başlangıç nesli normalde bilgisayar tarafından rasgele seçilir [18]. Rasgele seçilen bireylerin uyum faktörlerinin düşük olması durumunda iterasyon sayısı artacak ve hatta hiç çözüm bulunamayabilecektir. Böyle durumlarda başlangıç nesli değiştirilerek tekrar optimizasyon süreci başlatılır. Bazı problemlerde başlangıç neslini değiştirmek yerine, başlangıç neslinin kullanıcı tarafından belirlenmesi tercih edilebilir. Başlangıç neslinin uyumlu bireylerden oluşması durumunda optimum tasarım daha kısa zamanda bulunup iterasyon sayısı azalacaktır. Ancak iterasyon sayısının kullanıcı tarafından belirlenen yakınsama yüzdesine bağlı olduğu da unutulmamalıdır. Yakınsama oranı yüksek tutulursa iterasyon sayısı oldukça fazla olacaktır. Yakınsama oranı problemin hassasiyetine göre belirlenmektedir.

Başlangıç neslinde dikkat edilmesi gereken bir diğer önemli husus gruplandırma. Gruplandırma, sistem içinde aynı özelliklere sahip olan elemanların bir yerde toplanmasıdır. Bunun amacı sistemdeki bazı elemanların aynı tasarım değişkeninden

oluşmasını sağlamaktır. Böylece hem bir bireyi oluşturan dizi uzunluğu kısaltılmış hemde iterasyon sayısı azaltılmış olmaktadır.

### 2.1.3. Genetik Algoritmanın Özellikleri

- Genetik algoritma sonlu sayıda dizi üzerinde çalışır. Bu diziler belirli sıraya göre dizilmiş olan tasarım değişkenlerinden oluşur.
- Genetik algoritma tek bir dizide değil tüm dizilerin oluşturduğu toplukta araştırma yapar. Böylece kısmi yakınsamaların önüne geçilir.
- Genetik algoritmadaki operatörlerde tasarım değişkenleriyle işlemler rasgele yapılır. Böylece veri topluluğundaki tüm değerler araştırmaya katılır.
- Uygunluk fonksiyonu genetik algoritmanın araştırmayı devam ettirebilmesi için gereken bir bilgidir. Ayrıca bir türev bilgisi gerekli değildir [19, 20]. Genetik algoritma bu özelliği ile en karışık problemleri çözebilir. Bu diğer optimizasyon tekniklerine göre genetik algoritmanın en faydalı özelliğidir.
- Sınırlayıcılarla ilgili bilgiler araştırma sürecinden bağımsızdır. Bu özellik çok zor problemlerde genetik algoritmayı pratik kullanımlı yapar.
- Genetik algoritma ayrık değişkenli tasarım değişkenleriyle çalıştığı için bulunduğu sonuçlar değiştirilmeden kullanılır. Örneğin, çelik elemanlardan oluşan kafes sistemde optimizasyon yapıldığında genetik algoritmanın bulacağı profil enkesit alanları pratik hayatta mevcuttur. Oysa optimumluk kriteri gibi yöntemlerden bulunan sonuçlar ise doğrudan mevcut olmayabilir [21].
- Genetik algoritmalar sınırlı sayıda tasarım değişkenleriyle çalışır [14]. Tasarım değişkenlerinin sayısı arttıkça dizi uzunluğu da artar. Oysa değer kodlamasında böyle bir sorun yoktur.

### 2.1.4. Genetik Algoritmada Operatörler

Daha öncede belirtildiği gibi başlıcaları kopyalama, çaprazlama ve mutasyon olmak üzere bir takım genetik operatörler vardır. Bu operatörlerin sayısı arttıkça genetik algoritma daha aktif çalışmaktadır. Çünkü genetik algoritma doğadaki evrim sürecini taklit etmektedir. Doğadaki evrimi tam yansıtabilme için bu 3 operatörün dışında başka

operatörlerin de kullanılması yarar sağlamaktadır. Örneğin, elitizm operatörü en iyi bireyin nesilden nesile aktarılması için kullanılmaktadır.

#### 2.1.4.1. Kopyalama Operatörü

Kopyalama işlemi, uyum derecesi kötü olan bireylerin yerine iyi uyumlu bireylerin getirilmesini sağlamaktadır. Öncelikle nesildeki tüm bireylerin uyum faktörleri bulunur. Uyum faktörlerinden en düşük değere yada kullanıcı tarafından belirlenen bir değerden daha düşük değere sahip olan bireyler topluluktan dışarı atılırlar. Böylece toplulukta kalan bireyler tamamen iyi bireylerdir. Ancak topluluktan atılan birey sayısı kadar topluma yeni birey kazandırılması gerekir. Bunun için iyi uyumlu bireylerden kopyalar üretilir.

Bu çalışmada bir bireyden sadece 1 adet kopya yapılmıştır. İkinci bir kopya birey gerekirse bu birey uyum faktörü en yüksek ikinci bireyden kopyalanır. Kötü bir bireyi topluluktan dışarı atmak için seçilen uyum faktörü değeri 0,5'dir. Yani uyum faktörü 0,5'den düşük olan bireyler topluluktan dışarı atılmaktadır.

Tablo 3'de, 4 elemandan oluşan bir sistemin genetik algoritma sürecinde herhangi bir neslinin 10 bireyden oluştuğu düşünülerek bu 10 bireyin uyum faktörlerine göre bir sonraki nesline kopyalanması gösterilmektedir.

Tablo 3. Bireylerin eşleşme havuzuna kopyalanması

Birey no	Nesil				Uyum faktörü	Kopya sayısı	Eşleşme havuzu			
1	20	29	20	6	1.283	1	20	29	20	6
2	12	23	25	24	1.276	1	12	23	25	24
3	21	29	12	5	1.170	1	21	29	12	5
4	14	1	20	21	0.464	0	4	31	15	24
5	13	10	2	18	0.956	1	13	10	2	18
6	19	27	10	18	1.243	1	19	27	10	18
7	26	17	14	8	1.153	1	26	17	14	8
8	31	22	15	16	1.211	1	31	22	15	16
9	4	31	15	24	1.342	2	4	31	15	24
10	12	3	18	24	0.592	1	12	3	18	24

Tablo 3'den de görüleceği gibi uyum faktörü 0,464 olan 4. birey topluluktan dışlanmaktadır. Bu bireyin yerine uyum faktörü en yüksek olan 9. bireyden bir adet kopyalanmaktadır. Böylece eşleşme havuzuna 9. bireyden 2 adet kopya ve diğer bireylerden birer adet kopya birey gönderilmiştir. Kopyalama sayesinde oluşan her yeni nesil bir önceki nesle göre daha iyi bireylerden oluşmaktadır.

**Eşleşme Havuzu:** Genetik algoritmada bireyler uyum faktörüne göre eşleşme havuzuna alınırlar. Burada çaprazlanacak olan bireylere rasgele eşler verilir. Çaprazlama birey ile bireyin eşi arasında gerçekleşir. Bir birey kendisiyle eşleştirilemez ancak eşleşme havuzunda bir bireyden iki adet mevcutsa bunlar tesadüfen birbirine eş olabilmektedir.

Eşleşme havuzunda bir birey sadece bir bireyle eşleşebilir, çok eşliliğe izin verilmemektedir. Tablo 4'de daha önceki örnekte eşleşme havuzuna alınan 10 bireyin birbiriyle eşleştirilmesi gösterilmektedir. Bu tablodan görüldüğü gibi 1. birey 7. bireyle eşleştiğinden 7. birey de 1. bireyden başka kendisine eş seçemez. Ayrıca tam eşleşme yapılabilmesi için birey sayısının çift olması gerekmektedir.

Tablo 4. Eşleşme havuzunda bireylerin rasgele eşleşmesi

Birey no	Eşleşme havuzu				Eş birey
1	20	29	20	6	7
2	12	23	25	24	5
3	21	29	12	5	10
4	4	31	15	24	9
5	13	10	2	18	2
6	19	27	10	18	8
7	26	17	14	8	1
8	31	22	15	16	6
9	4	31	15	24	4
10	12	3	18	24	3

Yine Tablo 4'ten görüldüğü gibi, 4. birey 9. bireyle eşleşmiştir. Bu bireyler birbirinin aynısıdır. Çünkü daha önce eşleşme havuzuna alınan bireylerden 9. bireyin uyum faktörünün yüksek olması nedeniyle bu bireyden 2 adet kopyalanmaktadır.

#### 2.1.4.2. Çaprazlama Operatörü

Çaprazlama operatörü genetik algoritmada optimum çözüme yakınsama şansını artırdığı için ve topluma yeni bireyler kazandırdığı için kullanılmaktadır [22]. Çaprazlama operatörüne başlayabilmek için eşleşme havuzunda rasgele eşleştirilen çift bireyler çaprazlama havuzuna alınır.

Uygulanacak olan çaprazlama operatörünün türüne göre eş bireylerin çaprazlama noktaları rasgele belirlenir [18]. Bireyler bu çaprazlama noktalarından parçalanırlar. Oluşan parçalar eşler arasında karşılıklı olarak değiştirilir. Eşler arasında yapılan bu dizi parçalarının değiştirilmesinden dolayı artık farklı bir diziye sahip olan yeni bireyler topluma kazandırılmış olunur. Genetik algoritmada oluşan bu yeni bireylere ebeveyn bireylerin çocukları da denilmektedir [23].

Çaprazlama operatörü sayesinde genetik algoritmaların araştırma alanları genişir. Çünkü her iterasyonda çaprazlama operatörü yeni bireyleri keşfederek genetik algoritmanın bu yeni bireyler üzerinde de araştırma yapmasını sağlar. Oluşan çocuk bireyler ebeveyn bireylerin yerlerine geçerler. Ebeveyn bireyler ise bir sonraki nesile alınmazlar.

Çaprazlama operatörünün etkin olabilmesi için birey çiftlerinin çaprazlama noktalarının rasgele seçilmesine ve her çiftin de çaprazlama noktasının aynı olmamasına dikkat edilmelidir. Çaprazlama noktası, bir bireyin dizi uzunluğundaki tamsayıların sıra numaralarından herhangi biridir.

Örneğin, değer kodlamasında dizi uzunluğu 7 olan bir bireyin çaprazlama noktası 1, 2, 3, 4, 5, 6 veya 7 olabilir. Tablo 5'de, 30 adet tasarım değişkenine ve 8 bireye sahip olan 8 elemanlı bir sistem için çaprazlama havuzundaki bireylerin iki noktalı olarak çaprazlama noktalarının nasıl yapıldığı gösterilmektedir.



Tablo 5. Çaprazlama noktalarının belirlenmesi

Birey no	Eş birey no	Çaprazlama havuzundaki bireyler	Çaprazlama noktaları	
			1.	2.
1	6	24 7 2 13 28 6 9 5	3	5
2	8	13 6 8 14 4 14 6 4	1	7
3	5	1 30 16 15 1 11 8 9	6	8
4	7	4 19 29 19 5 13 17 18	2	4
5	3	8 21 14 30 5 4 29 9	6	8
6	1	9 7 8 9 6 26 25 8	3	5
7	4	16 8 9 8 9 16 13 23	2	4
8	2	4 3 10 7 10 8 17 1	1	7
Dizi sıra no		1 2 3 4 5 6 7 8		

Tablo 5’de 1. ve 2. çaprazlama noktaları bilgisayar tarafından rasgele belirlenmiştir. Burada eş bireylerin çaprazlama noktaları aynı olmalıdır. Örneğin, 1. bireyin eşi 6. bireydir. Öyleyse 1. ve 6. bireyin çaprazlama noktaları aynı olmalıdır. Yine bu tabloya baktığımızda 1. ve 6. bireylerin çaprazlama noktaları aynı olup bu noktalar 3 ve 5’dir. Ayrıca programlamada işlem kolaylığı açısından 1. çaprazlama noktası 2. çaprazlama noktasından küçük olmalıdır.

Literatürde değer kodlamasına ilişkin çaprazlama örneklerine rastlanılmayıp, alışlagelmiş yöntem olan ikilik kodlama için uygulanan çaprazlama çeşitlerinin değer kodlaması için de kullanabileceği belirtilmektedir. Bu çalışmada iki çeşit çaprazlama yöntemi ele alınmaktadır.

**Tek Noktalı Çaprazlama :** Tek noktalı çaprazlamada, çaprazlama havuzundaki eş bireyler için sadece bir tane çaprazlama noktası rasgele belirlenir. Ebeveyn bireyler bu noktadan farklı uzunlukta iki dizi parçası olarak bölünürler . Dizi parçaları eş birey ile karşılıklı değiştirilerek yeni çocuk bireyler oluşur.

Bu çaprazlamada, çaprazlama noktasına kadar olan kısım ilk bireyden ve çaprazlama noktasından dizi sonuna kadar olan ikinci kısım ise eş bireyden alınarak 1. çocuk birey oluşturulur. İlk çocuk birey oluşturulduktan sonra ebeveyn bireylerden artan parçalarla 2. çocuk birey oluşturulur. Çaprazlama sonunda oluşan yeni bireyler ebeveyn bireylerin yerini alarak bir sonraki nesile devam ederler. Bu tür çaprazlamaya örnek Tablo 6’da verilmektedir.



Tablo 6. Tek noktalı çaprazlama

		Çaprazlama noktası							
Birey		13	4	7	17	30	9	18	4
Eş birey		9	14	8	29	2	5	17	7
Çaprazlamadan sonra									
1.Çocuk birey		13	4	7	17	30	5	17	7
2.Çocuk birey		9	14	8	29	2	9	18	4
Dizi sıra no		1	2	3	4	5	6	7	8

Bu tabloda birey ile eş bireyin çaprazlama noktası 5'dir. Birinci çocuk bireyin ilk kısmı, ebeveyn bireyin ilk 5 elemanının (13, 4, 7, 17, 30) alınmasıyla, ikinci kısmı ise eş bireyin çaprazlama noktasından sonra olan 5, 17, 7 elemanlarının alınmasıyla oluşturulmaktadır. İkinci çocuk bireyin ilk kısmı, eş bireyden geri kalan 9, 14, 8, 29, 2 ve ikinci kısmı ise ebeveyn bireyden kalan 9, 18, 4 elemanlarının birleştirilmesiyle oluşturulmaktadır.

**İki Noktalı Çaprazlama:** İki noktalı çaprazlamada eş bireyler için 2 adet çaprazlama noktası rasgele belirlenir. Bireyler arasında dizi parçası değişimi bu iki çaprazlama noktası arasında kalan kısmın karşılıklı değiştirilmesiyle yapılır. İki noktalı çaprazlama tek noktalı çaprazlamaya göre daha verimlidir. Bu çalışmada önce tek noktalı çaprazlama yapılmış, daha sonradan iki noktalı çaprazlamaya geçilmiştir. Yapılan çalışma sonucu iki noktalı çaprazlama ile daha kısa zamanda optimum tasarıma gidildiği fark edildikten sonra ikili çaprazlama ile çalışmalara devam edilmiştir. Tablo 7'de bir birey ile eşinin iki noktalı olarak çaprazlanması gösterilmektedir.

Tablo 7. İki noktalı çaprazlama

Çaprazlama noktaları								
Birey	13	4	7	17	30	9	18	4
Eş birey	9	14	8	29	2	5	17	7
Çaprazlamadan sonra								
1. çocuk birey	13	4	7	29	2	5	18	4
2. çocuk birey	9	14	8	17	30	9	17	7
Dizi sıra no	1	2	3	4	5	6	7	8

Tablo 7'den görüldüğü gibi birey ve eşi için çaprazlama noktaları 3 ve 6 olarak alınmıştır. Birinci çocuk birey oluşturulurken ebeveyn bireyin sadece çaprazlama noktaları arasında yani 3. ile 6. eleman arasında kalan 4., 5. ve 6. sıra numaralı 17, 30, 9 nolu elemanları eş bireyin aynı sıra numaralarına sahip olan 29, 2, 5 nolu elemanlarıyla yer değiştirilmiştir. Aynı şekilde ikinci çocuk birey oluşturulurken eş bireyin 4., 5., ve 6. sıra numaralı elemanları ebeveyn bireyinkilerle değiştirilmektedir.

#### 2.1.4.3. Mutasyon Operatörü

Mutasyon, genetik algoritmada erken yakınsamanın önüne geçerek daha iyi sonuçlar bulunmasını sağlar. Ayrıca kötü uyumlu bireylerin optimizasyon problemine çözüm olmasını önlemeye çalışır. Değer kodlamasında mutasyon bireyin dizi uzunluğundaki tasarım değişkenleri değerlerinin rasgele değiştirilmesiyle yapılır. Mutasyon genellikle aşağıda belirtildiği gibi yapılmaktadır [13].

- Birkaç birey rasgele seçilip, değiştirilir.
- Sadece uygunluk faktörü yüksek olan bireylerin çözüme erken yakınsamasını önlemek için tasarım değişkenleri değiştirilir.
- Hiçbir birey değiştirilmez, yani mutasyon yapılmaz.

Daha öncede belirtildiği gibi mutasyon bir topluluktaki en iyi bireyi topluluktan dışarı atabileceği için bazı araştırmacılar mutasyonun kullanılmamasını, en azından kullanıldığı takdirde kullanılma oranının çok düşük alınması gerektiğini belirtmektedirler. Literatürde genetik algoritmalarla yapılan çalışmalarda mutasyon oranı oldukça düşüktür. Değer kodlamasında bireyin dizi uzunluğundaki bir tasarım değişkeni değerine küçük bir sayı ekleyerek yada çıkarılarak mutasyon yapılır.

Mutasyondan önceki birey : 1,29 5,68 2,86 4,11 5,55

Mutasyona uğramış birey : 1,29 5,68 2,73 4,22 5,55

Yukarıdaki örnekte görüldüğü gibi asıl bireyin 3. ve 4. elemanlarına küçük bir sayı ilave ederek yada çıkararak yeni bir birey elde edilmiştir. Ancak genetik algoritmada ayrık tasarım değişkenleri kullanıldığından mutasyona uğrayan bireyin yeni tasarım değişkenlerinin pratik hayatta mevcut olması gerekir. Yani, mutasyona uğrayan çelik profilleri enkesit alanları değeri mutasyona uğradıktan sonra da aldıkları değerler standartlarda olmalıdır.

Mutasyonu tasarım değişkenlerinin sıra numaraları ile yapmak daha kolaydır olmaktadır. Bunun için dört elemana ve 20 tasarım değişkenine sahip bir sistem için her hangi bir birey oluşturalım.

Birey : 1,12 4,80 5,69 8,70

Bu bireydeki  $1.12 \text{ cm}^2$  enkesit alanının 1. sıra nolu tasarım değişkeni,  $4.80 \text{ cm}^2$  enkesit alanının 5. sıra nolu tasarım değişkenine,  $5.69 \text{ cm}^2$  enkesit alanının 17. sıra nolu tasarım değişkenine ve  $8.70 \text{ cm}^2$  enkesit alanının ise 20. sıra nolu tasarım değişkeninden oluştuğunu düşünürsek bireyi sıra noları ile şöyle oluşturabiliriz.

Birey : 1 5 17 20

Mutasyona uğrayacak olan bireyin dizi uzunluğundaki hangi elemanın mutasyona uğrayacağı da yine bilgisayarca rasgele belirlenir. Bu birey için rasgele belirlenmiş olan 2. grubun mutasyona uğradığını kabul edelim. Bu durumda 2. dizi sıra numarasına sahip olan 5 nolu tasarım değişkeni yine bilgisayarca belirlenecek alan rasgele bir sayı ile

değiştirilecektir. Bu rasgele sayının 6 olduğunu düşünürsek meydana gelecek olan yeni birey 1-6-17-20 şeklinde olmaktadır.

Burada oluşan yeni birey gerçek değerleriyle 1,12- 2,67- 5,69- 8,70 şeklindedir. Görüldüğü gibi bu bireyin mutasyondan önceki 2. elmanı (4,80 cm<sup>2</sup>) daha önce standartlarda var olduğu belirtilen 20 tasarım değişkeninden 6. tasarım değişkeni olan 2,67 cm<sup>2</sup> değeriyle değişerek mutasyona uğramış olmaktadır.

### 2.1.5. Genetik Operatörlerle İlgili Öneriler

- Çaprazlama oranı, genellikle %80 - %95 arası olup yüksek alınmalıdır [13]. Ancak literatürde bazı problemler için bu oranın %60 olduğu görülmüştür. Bu çalışmada çaprazlama oranı %100 alınmıştır.
- Mutasyon oranı, çaprazlama oranının aksine çok düşük tutulmalıdır [24]. Literatürde en iyi mutasyon oranı % 0,5-1 arası verilmiştir. Bu çalışmada her nesilden 1 birey mutasyona uğratılmıştır.
- Toplam birey sayısının fazla olması genetik algoritmanın performansını yani çözüme ulaşma hızını artırmaz [13]. Literatürde birey sayısının 20-30 olması, bazı problemlerde 50-100 arası olması tavsiye etmektedirler. Bazı araştırmacılar birey sayısının dizi uzunluğuna bağlı olduğunu iddia etmektedirler. Örneğin ikilik kodlama sisteminde 32 gen varsa birey sayısının da 32 olması istenir. Oysa bu çalışmada değer kodlaması kullanıldığından böyle bir iddianın dikkate alınması gerekmemektedir.
- Genetik algortmada tüm değerler rasgele bulunur. Ancak başlangıç neslinin seçiminde, iterasyon sayısının fazla olması için, rasgelelik olmayabilir. Bunun dışında çaprazlama noktaları ve mutasyona uğrayacak birey tamamen rasgele seçilmelidir.
- Optimize edilecek problemin türüne göre ağaç kodlama, ikilik kodlama ve değer kodlaması tercih edilebilir. Örneğin, bir seyahat satıcılığı probleminde ağaç kodlaması kullanılır. Kodlama türünün seçimi genetik algoritmanın diğer operatörlerinin daha etkin çalışması açısından önemlidir.

### 2.1.6. Uygunluk Fonksiyonu

Uygunluk fonksiyonu, genetik algoritmanın en önemli özelliklerinden biridir. Bu fonksiyon algoritmayı istenen amaca yönelten tek mekanizmadır. Her hangi bir bireyin uygunluk fonksiyonu, bireyin problemin çözümü olup olamayacağını işaret eder. Uygunluk fonksiyonu tüm bireylere uygulanarak tüm bireyler için bir uygunluk değeri hesaplanır.

Uygunluk fonksiyonunun etkili bir değerlendirme yapabilmesi ve bir sonraki nesilde bireylerin istenen çözüme çok yaklaştığını söyleyebilmek için, uygunluk fonksiyonu eğrisi çok az uç noktalara sahip olmalıdır. Yani, tek bir maksimum yada minimum değerle beraber sabit bir artış yada azalış göstermelidir [10].

### 2.1.7. Cezalandırılmış Amaç Fonksiyonu Değeri

Mühendislikte bir çok optimizasyon problemi için, sınırlayıcıların tatmin edici performansı oldukça önemlidir. Oysa genetik algoritma sınırlayıcılarla çalışmaz. Sınırlayıcıları dikkate almak için ceza fonksiyonlarını kullanır. Ceza fonksiyonları sınırlayıcılar ile amaç fonksiyonunu birleştirerek problemi sınırlayıcısız bir hale dönüştürür.

Bireyler, öncelikle amaç fonksiyonunda değerlendirilmeye alınıp sınırlayıcılar tarafından kontrol edilir. Eğer bireyler sınırlayıcıları ihlal etmemişse bireyin ceza fonksiyonu ve amaç fonksiyonu aynı değeri alır. Eğer sınırlayıcılar ihlal edilmişse, çözümler uygun olmayıp bireyin ceza fonksiyonunun değeri anlamsız bir sayı olur. Bu durumda anlamlı ifadeler ortaya koymak için genellikle ceza fonksiyonları kullanılır. Amaç fonksiyonunun, ceza fonksiyonlarıyla birleştirilmiş haline cezalandırılmış amaç fonksiyonu adı verilir. Örneğin, bir uzay kafes sisteminde bir bireyin yapısal analizinden alınan sonuçlara göre belirli bir düğüm noktasının yerdeğiştirmesi 2 cm olsun. Aynı düğüm noktası için müsaade edilen maksimum yerdeğiştirme değeri 1 cm ise bu birey sınırlayıcıları ihlal etmiş olup, uygun bir çözüm olmaz.

### 2.1.8. Genetik Algoritmada Toplum Nüfusu

Genetik algoritmada toplum nüfusu, herhangi bir nesildeki birey sayısıdır. Doğal evrimde bir populasyonun nüfusu çevre koşulları tarafından belirlenir. Genetik algoritmada ise toplum nüfusu, probleme çözüm olabilecek birey (çözüm) sayısına bağlıdır [25].

Genellikle bir toplulukta ne kadar çok birey varsa, bir sonraki nesilde devamlılığını sürdüren birey sayısı ve çeşidi o denli çok olur. Çok sayıda ve çeşitte birey olması nesil içindeki tek bir türün baskın olmasını önler [25]. Ancak bir nesilde fazla birey olması, genetik algoritmanın bu kadar bireyi değerlendirmesi için çok yavaş işleyeceğine işaret etmektedir.

Eğer toplam birey sayısı az ise, genetik algoritmada başlangıç neslinde uyum faktörü yüksek olan bireyler bir sonraki nesillerde baskın olur. Yani erken yakınsama olur ki bu durum genetik algoritmada istenmez. Böyle durumlarda genetik algoritma çaprazlama operatörü yeni bir birey üretmekte zorlanıp, sadece mutasyon operatörü sayesinde yeni birey oluşturma şansı aranır.

Yeni bir nesil oluşturulduğunda birey sayısının bir önceki nesildekine eşit olması istenir. Bu durum, genetik algoritmada uyulması gereken bir koşul değildir. Eski bireyler yerlerini kendilerinden oluşan yeni bireylere verirler. Ancak iyi bireyler bir sonraki nesile hiç değiştirilmeden devam ettirilebilirler. Genetik algoritmada bu olaya elitizm adı verilir.

## 2.2. Çelik Çubukların Gerilme Tahkiki

Bu çalışmada çelik çubuklar, çekme çubuğu ve basınç çubuğu olarak boyutlandırılmıştır. Çekme çubuklarında sadece elemanda oluşan gerilmenin müsaade edilen emniyet gerilmesini aşp aşmadığı araştırılırken basınç çubuklarında ayrıca aksenal stabilite yani burkulma hesabı yapılmaktadır.

### 2.2.1. Çekmeye Çalışan Elemanlar

Boyuna doğrultuda çekme kuvveti taşıyan elemanlara çekme çubuğu denir. Çekme çubuklarında gerilme denetimi,  $A$ , çubuğun enkesit alanını,  $N$ , çubukta oluşan çekme kuvvetini,  $\sigma_a$  ise çeliğin akma gerilmesini göstermek üzere,

$$\sigma = \frac{N}{A} \leq \sigma_{\text{çem}} \quad (1)$$

bağıntısıyla yapılmaktadır.

Çekme emniyet gerilmesi ( $\sigma_{\text{çem}}$ ) faydalı enkesitte en fazla  $\sigma_{\text{çem}} \leq 0,6 \sigma_a$  olmalıdır.

### 2.2.2. Basınca Çalışan Elemanlar

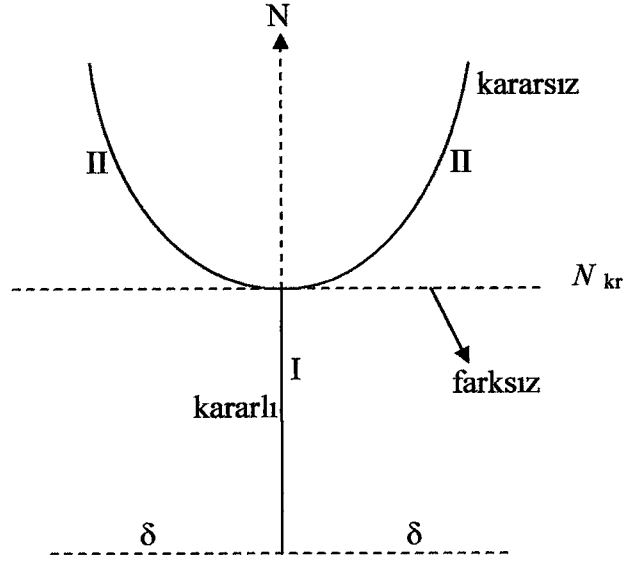
Basınç çubukları daha çok negatif (-) gerilmelerin hakim olduğu taşıyıcı elemanlardır. Basınç çubuklarının burkulması çelik malzemede dayanım sınırının aşılması ile değil stabilitesinin bozulmasıyla oluşur. Dolayısıyla burkulma stabilite problemi [26].

$N_{kr}$  çubuk elemanda kritik burkulma yükünü göstermek üzere bir basınç çubuğunda  $N$  merkezi kuvvetinin altında,  $N = N_{kr}$  oluncaya kadar olumsuz bir etki görülmez. Yani, çubuk eleman stabilitesini korumaktadır.  $N = N_{kr}$  olduğu zaman çubuk eleman kararsız bir duruma gelir.  $N > N_{kr}$  olduğu zaman ise çubukta burkulma oluşmaktadır. Çubuk elemanın kararsız kalması veya büyük yerdeğiştirmeler yapması yapı dengesi bakımından istenmeyen durumlardır.

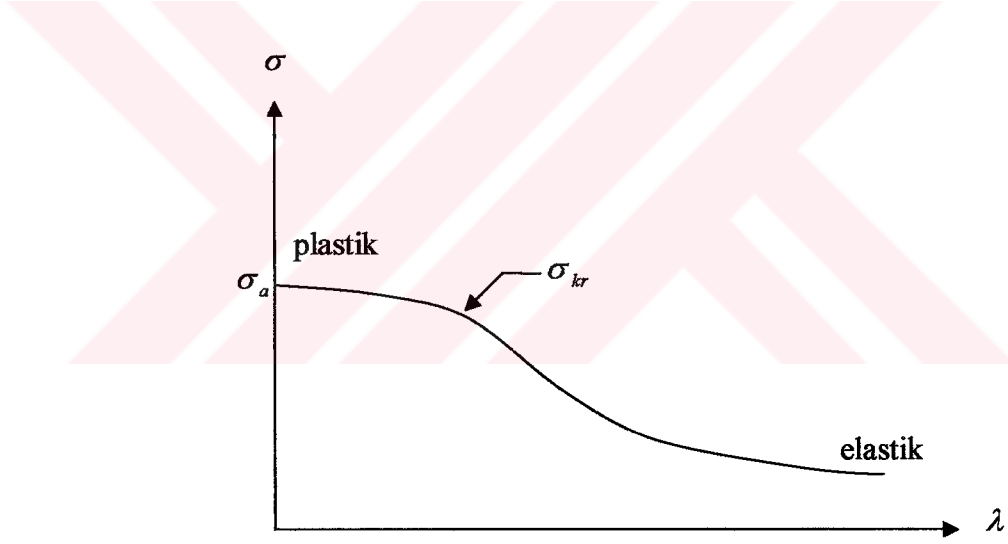
Yukarıdaki ifadeyi şekille açıklamak amacıyla Şekil 5'de sembolik olarak iki ucundan mafsallı bir basınç çubuğunun  $N$  merkezi yük etkisi altında davranışı gösterilmektedir.

$\delta$  yerdeğiştirme olmak üzere I. eğride gösterildiği gibi stabilitesini koruyan çubuk eleman, yükün  $N_{kr}$  değerine gelmesiyle II. eğrideki gibi büyük yerdeğiştirmeler yaparak gerilme sınırına gelmeden dengesini kaybetmektedir yani burkulmaktadır.

Basınca çalışan bu tür çubuk elemanların burkulmasını önlemek amacıyla çubuk elemanın enkesit alanı artırılabilir. Bu çalışmada basınca çalışan elemanların enkesit alanları genetik algoritma tarafından optimum değerini almaktadırlar. Şekil 6'da basınç elemanları için gerilme-burkulma grafiği verilmektedir.



Şekil 5. Basınç etkisi altında çubuk elemanın davranışı



Şekil 6. Basınç elemanı için gerilme-burkulma grafiği

Şekil 6'da ' $\sigma_{kr}$ ' kritik gerilmeyi göstermek üzere, bu değer gerilme-burkulma grafiğini elastik ve plastik sınırlar olmak üzere 2 kısma ayırmaktadır. Bu değere karşılık gelen burkulma değeri ise kritik burkulma olarak tanımlanmaktadır.



Elastik-plastik burkulmada çubuk eksenine dik ön eğrilik (ön kusur veya yüklemde kaçıklık) durumları da dikkate alınmalıdır. Bu bölgedeki gerilme, akma sınırı gerilmesiyle elastik sınır gerilmesi arasındadır.  $S_k$  çubuk elemanın burkulma boyunu,  $I$  atalet momentini,  $i$  atalet yarıçapını,  $E$  elastisite modülünü,  $\sigma_a$  akma sınırı gerilmesini,  $\sigma_e$  elastik gerilmeyi,  $\sigma_{kr}$  kritik gerilmeyi ve  $\lambda$  narinliği göstermek üzere burkulma hesabında aşağıdaki formüller kullanılmaktadır.

$$i = \sqrt{\frac{I}{A}} \quad (2)$$

$$\lambda = \frac{S_k}{i} \quad (3)$$

$$\sigma_{kr} = \sigma_a - \sigma_e = \sigma_a - \frac{\sigma_a^2}{4\pi^2 E} \lambda^2 \quad (4)$$

Burkulmanın elastik veya plastik bölgede olduğuna karar vermek için aşağıdaki bağıntı yardımıyla plastik narinlik sınırı yani kritik narinlik hesaplanmaktadır.

$$\lambda_p = \sqrt{\frac{2\pi^2 E}{\sigma_a}} \quad (5)$$

Kullanılacak olan basınç emniyet gerilmeleri plastik narinlik sınırına bağlı olarak aşağıdaki formüllerden hesaplanmaktadır.

$$\lambda \leq \lambda_p \quad \text{ise;} \quad \sigma_{bem} = \frac{\sigma_{kr}}{n} = \sigma_a \left[ \frac{1 - \frac{1}{2} \left( \frac{\lambda}{\lambda_p} \right)^2}{n} \right] \quad (6)$$

$$\lambda \geq \lambda_p \quad \text{ise;} \quad \sigma_{bem} = \frac{\sigma_a}{2n} = \frac{2}{5} \pi^2 \frac{E}{\lambda^2} \quad (7)$$

Yukarıdaki bağıntılardaki ‘ $n$ ’ burkulma hesabında emniyet katsayısı olup aşağıdaki bağıntılardan hesaplanmaktadır.

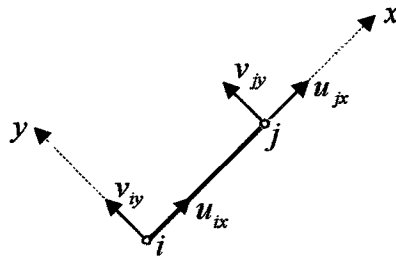
$$\lambda < 20 \quad \text{ise;} \quad n = 1,67$$

$$\lambda_p \geq \lambda \geq 20 \quad \text{ise;} \quad n = 1,5 + 1,2 \left( \frac{\lambda}{\lambda_p} \right) - 0,2 \left( \frac{\lambda}{\lambda_p} \right)^3 \geq 1,67 \quad (8)$$

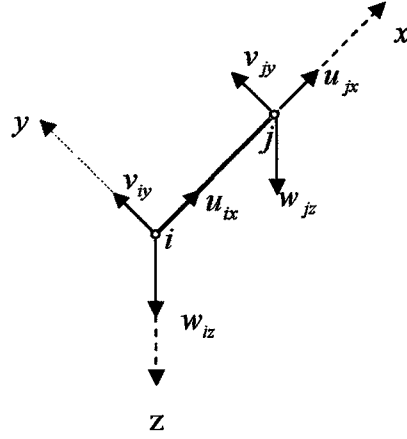
### 2.3. Düzlem ve Uzay Kafes Sistemlerin Yapısal Çözümlemesi

Bu çalışmada kafes sistemlerin yapısal analizini yapmak için matris deplasman yöntemi kullanılmaktadır. Aşağıda düzlem ve uzay kafes sistemlerle ilgili eleman rijitlik matrisleri, dönüşüm matrisleri, eleman ucu yerdeğiştirme ve eleman uç kuvvetlerini gösteren matris formları verilmektedir.

Şekil 7 ve Şekil 8’de,  $i, j$  çubuk elemanın düğüm noktalarını,  $x, y, z$  eleman eksen takımlarını,  $u_{ix}, v_{iy}, w_{iz}$  sırasıyla eleman eksen takımında  $x, y, z$  yönlerinde  $i$  noktasının yerdeğiştirmelerini ve  $u_{jx}, v_{jy}, w_{jz}$  ise sırasıyla eleman eksen takımında  $x, y, z$  yönlerinde  $j$  noktasının yerdeğiştirmelerini göstermektedir.



Şekil 7. Düzlem kafes elemanı için uç yerdeğiştirmeleri



Şekil 8. Uzay kafes elemanı için uç yerdeğiřtirmeleri

Uç yerdeğiřtirmeleri düzlem ve uzay kafes sistemleri için matris formunda, sırasıyla ařağıdaki gibi gösterilmektedir.

$$\{U\} = \begin{Bmatrix} u_{ix} \\ v_{iy} \\ u_{jx} \\ v_{jy} \end{Bmatrix} \quad \{U\} = \begin{Bmatrix} u_{ix} \\ v_{iy} \\ w_{iz} \\ u_{jx} \\ v_{jy} \\ w_{jz} \end{Bmatrix} \quad (9)$$

Eleman rijitlik matrisi düzlem kafes sistemler için,  $E$  elastisite modülünü,  $I$  atalet momentini,  $L$  çubuk elemanın boyunu göstermek üzere

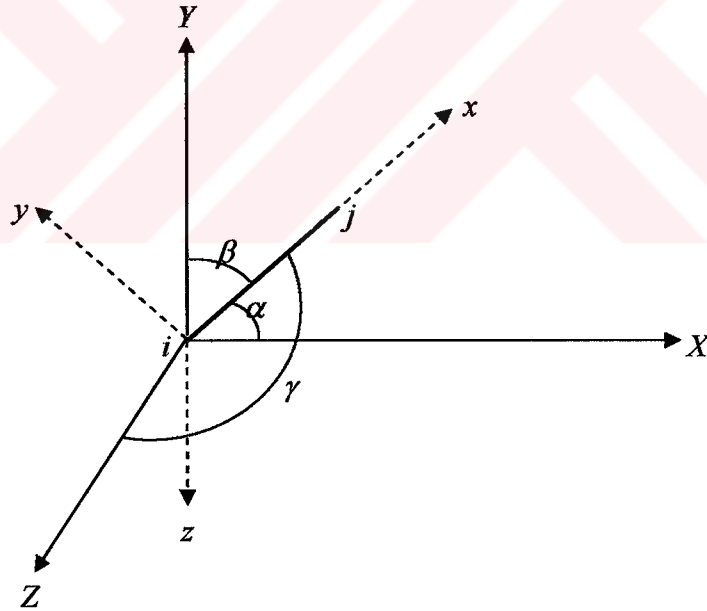
$$[K]_{ij}^e = \begin{bmatrix} \frac{EA}{L} & 0 & -\frac{EA}{L} & 0 \\ 0 & 0 & 0 & 0 \\ -\frac{EA}{L} & 0 & \frac{EA}{L} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (10)$$

bağıntısı, uzay kafes sistemler için ise

$$[K]_{ij}^e = \begin{bmatrix} \frac{EA}{L} & 0 & 0 & -\frac{EA}{L} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{EA}{L} & 0 & 0 & \frac{EA}{L} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (11)$$

bağıntısı ile hesaplanmaktadır.

Bu rijitlik matrislerinin, dönüşüm matrisleriyle sistem eksen takımındaki rijitlik matrislerine çevrilmesi gerekmektedir. Dönüşüm matrislerini bulmak için doğrultu kosinüsleri kullanılmaktadır. Doğrultu kosinüsleri, eleman eksen takımlarının sistem eksen takımlarıyla yaptıkları açıların kosinüsleri olarak tanımlanmaktadır. Şekil 9'da  $X, Y, Z$  sistem eksen takımlarını,  $x, y, z$  ise eleman eksen takımlarını göstermektedir.



Şekil 9. Sistem ve eleman eksen takımları

$\alpha, \beta, \gamma$  açıları sırasıyla eleman eksen takımlarının  $X, Y, Z$  sistem eksen takımları ile yaptıkları açılardır.

$l_x, m_x, n_x$  sırasıyla eleman eksen takımındaki  $x$  ekseninin sistem eksen takımındaki  $X, Y, Z$  eksenleriyle yaptığı açılarının kosinüslerini,  $l_y, m_y, n_y$  sırasıyla eleman eksen takımındaki  $y$  ekseninin sistem eksen takımındaki  $X, Y, Z$  eksenleriyle yaptığı açılarının kosinüslerini ve  $l_z, m_z, n_z$  sırasıyla eleman eksen takımındaki  $z$  ekseninin sistem eksen takımındaki  $X, Y, Z$  eksenleriyle yaptığı açılarının kosinüslerini göstermek üzere dönüşüm matrisleri düzlem kafes sistemler için

$$[T] = \begin{bmatrix} l_x & m_x & n_x \\ l_y & m_y & n_y \\ l_z & m_z & n_z \end{bmatrix} \quad (12)$$

bağıntısıyla, uzay kafes sistemler için ise

$$[T] = \begin{bmatrix} l_x & m_x & n_x & 0 & 0 & 0 \\ l_y & m_y & n_y & 0 & 0 & 0 \\ l_z & m_z & n_z & 0 & 0 & 0 \\ 0 & 0 & 0 & l_x & m_x & n_x \\ 0 & 0 & 0 & l_y & m_y & n_y \\ 0 & 0 & 0 & l_z & m_z & n_z \end{bmatrix} \quad (13)$$

bağıntısıyla belirlenmektedir.

Sistem eksen takımında eleman rijitlik matrisi

$$[K]_{ij}^s = [T]^T \cdot [K]_{ij}^e \cdot [T] \quad (14)$$

bağıntısıyla hesaplanmaktadır.

Sistem eksen takımında eleman rijitlik matrisleri bulunduktan sonra tüm sistemin sistem rijitlik matrisi

$$[K] = \sum_{L=1}^n [K]_{ij}^s \quad (15)$$

bağıntısıyla belirlenmektedir.

Sistemdeki düğüm noktalarının yerdeğiřtirmeleri,  $\{F\}$  dış yük vektörünü ve  $\{U\}$  yerdeğiřtirmeleri göstermek üzere

$$\{U\} = [K]^{-1} \cdot \{F\} \quad (16)$$

bağıntısıyla belirlenmektedir. Bulunan yerdeğiřtirmeler sistem eksen takımındadır. Eleman çubuk kuvvetleri

$$\{f\}_{ij}^e = [K]_{ij}^e \cdot [T] \cdot \{U\} \quad (17)$$

bağıntısıyla hesaplanmaktadır.

#### 2.4. Genetik Algoritmada Optimizasyon Süreci

Yapının toplam ağırlığını minimuma indirmek için yapı elemanlarının olabildiğince en küçük kesit alanına sahip çelik profillerden oluşturulması hedeflenmektedir. Bu amaç doğrultusunda,  $\rho$  birim hacim ağırlığı,  $L$  çubuk boyunu,  $A$  çubuk enkesit alanını ve  $W$  yapı ağırlığını göstermek üzere minimum yapı ağırlığı

$$W_{\min} = \sum_{i=1}^n \rho(L.A)_i \quad (18)$$

bağıntısıyla verilmektedir. Minimum ağırlığı elde etmek için sınırlayıcıların ihlal edilmemesi gerekmektedir.  $\delta$  düğüm noktasının yerdeğiřtirmesini,  $\delta_u$  müsaade edilen maksimum yerdeğiřtirmeyi,  $\sigma$  elemanda oluşan gerilmeyi,  $\sigma_u$  müsaade edilen maksimum gerilmeyi göstermek üzere sınırlayıcılar

$$\delta \leq \delta_u \quad i = 1, \dots, k \quad (19)$$

$$\sigma \leq \sigma_u \quad j = 1, \dots, n \quad (20)$$

bağıntılarıyla dikkate alınmaktadır.

Burada 'k' yerdeğiştirme yapabilen nokta, 'n' ise eleman sayısıdır. Sınırlayıcıların ihlal edilmesi durumunda ceza fonksiyonunun belirlenebilmesi için sınırlayıcıların normalize edilmiş formları aşağıdaki gibi verilmektedir.

Gerilme sınırlayıcıları çekme çubukları için,  $\sigma_{em}$  çekme emniyet gerilmesini göstermek üzere

$$g_i(x) = \frac{\sigma_i}{\sigma_{em}} - 1 \leq 0 \quad i = 1, \dots, k \quad (21)$$

şeklinde, basınç çubukları için,  $\sigma_{bem}$  basınç emniyet gerilmesini göstermek üzere

$$g_i(x) = \frac{\sigma_i}{\sigma_{bem}} - 1 \leq 0 \quad i = 1, \dots, n \quad (22)$$

şeklinde yazılabilmektedir. Yerdeğiştirme sınırlayıcıları ise

$$g_i(x) = \frac{\delta_i}{\delta_u} - 1 \leq 0 \quad i = 1, \dots, k \quad (23)$$

şeklinde yazılabilmektedir. Sınırlayıcılar yukarıda belirtilen şekilde dönüştürüldükten sonra sınırlayıcıların ihlal edilme katsayıları

$$c_i = g_i(x) \quad g_i(x) > 0 \quad \text{ise} \quad (24)$$

$$c_i = 0 \quad g_i(x) \leq 0 \quad \text{ise} \quad (25)$$

bağıntılarıyla hesaplanmaktadır. Bir bireyin ceza fonksiyonu, C, o bireye ait ihlal edilme katsayılarının toplanmasıyla, m sınırlayıcı sayısını göstermek üzere

$$C = \sum_i^m c_i \quad (26)$$

bağıntısıyla belirlenmektedir.

Yapı ağırlığı ve ceza fonksiyonuna bağlı olarak cezalandırılmış amaç fonksiyonu, P sınırlayıcıların veya yapı toplam ağırlığının önemini artıran, problemde problemde farklı değerler alabilen bir katsayıyı göstermek üzere

$$\phi(x) = W(x) \cdot [1 + P.C] \quad (27)$$

bağıntısıyla belirlenmektedir.

Cezalandırılmış amaç fonksiyonundan yararlanarak bir bireyin kopya sayısını bulmak için bireylerin uyum dereceleri

$$F_i = (\phi(x)_{\max} + \phi(x)_{\min}) - \phi(x)_i \quad (28)$$

bağıntısıyla hesaplanmaktadır.

Bireylerin uyum dereceleri bulduktan sonra o bireyden eşleşme havuzuna kaç adet alınacağı uyum faktörlerinin,  $F_c$ , alacağı değere bakılarak belirlenir.  $F_i$ ,  $i$ . bireyin uyum derecesini ve  $n$  toplam birey sayısını göstermek üzere, bireylerin ortalama uyum dereceleri,  $F_{ort}$  ve uyum faktörü,  $F_c$  sırasıyla,

$$F_{ort} = \sum F_i / n \quad (29)$$

$$F_c = F_i / F_{ort} \quad (30)$$

bağıntılarıyla belirlenir. Bu formüllerle uyum faktörleri bulunup elde edilen sayı en yakın tamsayı değerine çevrildiğinde elde edilen tamsayı ilgili bireyin kopya sayısı olmaktadır.

Uyum faktörü 0.5 ve daha yukarı olan bireyler eşleşme havuzuna alınırken bu değerden küçük olanlar alınmazlar. Yani, uyum faktörü 0.5'den küçük olan bireyler toplumdaki dışlanırlar.

Yukarıdaki adımlar her birey için tekrarlanarak genetik algoritma süreci devam ettirilir. Bu süreç sonunda, başlangıçta kullanıcı tarafından belirlenen yakınsama elde edildiğinde ve aynı zamanda bulunan çözümler sınırlayıcıları ihlal etmemişse bu çözüm optimum çözüm olarak kabul edilir.

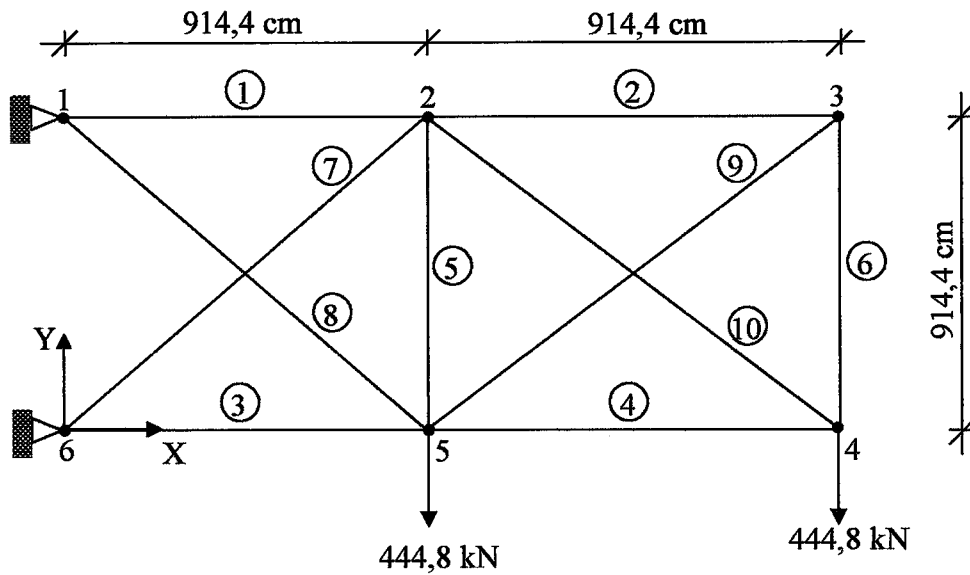


## 2.5. Kafes Sistemlerin Optimizasyon Örnekleri

Bu çalışmada Fortran programlama dilinde bir genetik algoritma programı kodlanarak düzlem ve uzay kafes sistemlerin optimum tasarımı gerçekleştirilmiştir. Optimum tasarımı gerçekleştirilen örnek sistemler literatürden seçilmiştir. Bu amaçla 9 çubuklu düzlem kafes sistem [28], 10 çubuklu düzlem kafes sistem [4, 20, 29, 30], 25 çubuklu uzay kafes sistem [1, 14, 31, 32], 52 çubuklu uzay kafes sistem [33, 34] ve 72 çubuklu uzay kafes sistem [35] örnek olarak alınmıştır. Bu çalışmada programlama aşamasında literatürdekilerden farklı olarak ikilik kodlama sistemi yerine değer kodlaması kullanıldığı için elde edilen tasarım sonuçlarının doğruluğunu kanıtlamak amacıyla sonuçlar literatürdeki çalışmalarla karşılaştırılmıştır.

### 2.5.1. On Çubuklu Düzlem Kafes Sistem

Şekil 10'da geometrik özellikleri verilen on çubuklu sistem literatürde en yaygın rastlanan örnektir. Bu sistemin AISC-ASD (American Institute of Steel Construction-Allowable Stress Design)'ye, TS-648'e göre burkulmalı ve burkulmasız olmak üzere üç farklı tasarımı gerçekleştirilmiştir. Genetik algoritmada optimizasyon sürecini göstermek üzere on çubuklu bu sistemin AISC-ASD'ye göre tasarımı detaylı olarak ele alınmaktadır.



Şekil 10. On çubuklu düzlem kafes sistem ve yükleme durumu

### 2.5.1.1. AISC-ASD'ye Göre On Çubuklu Sistemin Burkulmalı Tasarımı

Eksenel stabilite göz önünde tutularak tasarımı yapılan bu sistem, Groenwold [30] tarafından da eksenel stabilite dikkate alınarak tasarlandığı için bu araştırmacı tarafından alınan değerler kullanılmıştır. Basınç çubukları için emniyet gerilmesi hesapları AISC-ASD'ye [27] göre yapılmıştır. Tasarım için kullanılan çelik profili enkesit alan değerleri ve atalet yarıçapları Tablo 8'de verilmektedir.

Tablo 8. On çubuklu sisteme ait enkesit alanları ve atalet yarıçapları

Enkesit sıra no	Alan (cm <sup>2</sup> )	Atalet yarıçapı (cm)	Enkesit sıra no	Alan (cm <sup>2</sup> )	Atalet yarıçapı (cm)
1	10,45	2,01	17	87,10	4,55
2	13,74	1,99	18	90,97	5,28
3	14,26	1,20	19	99,35	5,64
4	15,16	1,08	20	110,32	5,33
5	17,10	1,22	21	123,23	7,67
6	18,79	1,46	22	136,13	7,72
7	20,52	1,36	23	143,87	6,63
8	24,71	2,54	24	149,68	7,75
9	30,19	3,23	25	158,71	7,47
10	35,74	3,25	26	165,16	7,80
11	39,74	3,20	27	170,97	9,40
12	45,68	4,09	28	181,94	7,85
13	53,23	4,11	29	187,74	9,42
14	58,90	5,13	30	197,42	7,39
15	66,45	5,16	31	206,45	9,47
16	75,48	5,18	32	216,13	6,15

Bu sistemde müsaade edilen maksimum yerdeğiştirme, bütün düğüm noktaları için her iki doğrultuda 5,08 cm (2.0 inç) olup, uygulanan dış yükler 444,8 kN (100 kips)'dur. Elastisite modülü  $E=206850$  Mpa ( $30 \times 10^6$  psi), malzeme birim hacim ağırlığı  $\rho=0,076$  N/cm<sup>3</sup> ( $0,28$  lb/in<sup>3</sup>), çekme çubukları için emniyet gerilmesi  $\sigma = 172,375$  Mpa (25 ksi) olarak dikkate alınmıştır (1 kip=4,448 kN, 1 lb=4,448 N, 1 ksi=6,895 Mpa, 1 psi=0,006895 Mpa, 1 Mpa=N/mm<sup>2</sup>, 1 inç=2,54 cm, 1 pa=N/m<sup>2</sup>, 1 N=0,1 kgf).

Genetik algoritma programı ile bu sistemin çözümü için 32 adet çelik profil enkesit alanı kullanılmış ve toplam 30 adet birey oluşturulmuştur. Amaç fonksiyonuna ait ceza katsayısı 4.5 alınıp, optimum tasarım için yakınsama oranı %80 olarak kabul edilmiştir. Program sürecinde en fazla 200 iterasyona kadar izin verilmiştir. Bu sistemin çözümünde 94. iterasyonda optimum sonuçlara ulaşılmıştır. Aşağıda genetik algoritmada tasarım değişkenlerinin kodlanması, uyumluluklarının hesaplanması, kopyalanması, çaprazlanması mutasyona uğramaları ve optimum tasarımın elde edilmesi anlatılmaktadır.

Tablo 9'da genetik algoritmada başlangıç nesli bilgisayar tarafından rasgele oluşturulmuştur. Bu tabloda bireylerin ağırlıkları ve elemanlarda oluşan maksimum gerilme değerleri de verilmektedir. Daha öncede belirtildiği gibi başlangıç nesli yakınsamanın sağlanabilmesi için oldukça önemlidir. Eğer ilk nesli oluşturan bireylerin uyum dereceleri çok düşük olursa yakınsama oranını yakalayabilmek için gereken iterasyon sayısı fazla olacaktır. Ancak genetik algoritmanın verimli çalışabilmesi için diğer genetik operatörlerde de olduğu gibi bireylerin oluşturulmasında da rasgelelikten vazgeçilmemelidir. Bireylerin, rasgele değil de, kullanıcı tarafından belirlenmesi durumunda belki genetik algoritma daha kısa zamanda sonuç üretebilir ama erken yakınsama olabileceği için sonuçlar aldatıcı olabilir.

Bu sistem için 30 adet birey oluşturulmuştur. Örnek olarak 15. bireyi ele aldığımızda Tablo 9'da başlangıç neslinde 15. satırda bulunan 16-12-10-16-15-12-3-1-31-14 sayıları on çubuklu sistemdeki 1., 2., 3., 4., 5., 6., 7., 8., 9. ve 10. elemanların sırasıyla 16, 12, 10, 16, 15, 12, 3, 1, 31, ve 14 nolu tasarım değişkenlerinden oluştuğunu ifade eder. Bu sayılar tasarım değişkenleri enkesit alanları değerlerinin küçükten büyüğe doğru sıralanması ile oluşan sıra numaralarıdır. İkilik kodlamada tasarım değişkenlerinin sıra nosunu elde etmek için başlangıçta verilen şifreler (sayı dizileri) çözülür ama bu çalışmada değer kodlaması kullanıldığı için tasarım değişkenleri noları doğrudan kullanılmaktadır.

On çubuktan oluşan bu sistem için her bir bireyin ağırlığı başlangıç neslinde kuşkusuz birbirinden farklı olacaktır. İterasyon sayısı arttıkça genetik algoritma, bu rasgele belirlenmiş olan bireyleri adeta sömürerek en iyi bireyi bulacaktır. Nesillerde, başlangıçta belirlenen yakınsama oranı yakalanınca genetik algoritma optimum sonuca ulaşmış olmaktadır. Ancak optimum sonuç, sınırlayıcıları ihlal etmemelidir.

Tablo 9. On çubuklu sistem için genetik algorithmada başlangıç nesli

Birey no	Başlangıç nesli										Ağırlık (kN)	Maksimum gerilme (Mpa)
1	25	25	10	8	24	1	18	20	5	5	60,531	212,57
2	26	1	4	14	18	23	12	19	26	13	69,387	-329,88
3	4	9	27	4	16	12	19	30	8	23	70,241	321,71
4	32	28	4	6	22	12	4	17	32	30	93,364	-317,24
5	3	6	11	4	26	4	31	2	31	10	64,098	348,04
6	27	12	19	22	14	2	29	12	7	2	62,776	205,94
7	3	29	27	14	28	7	16	14	6	9	62,120	358,64
8	1	29	4	8	7	7	25	27	17	19	70,121	713,59
9	21	17	3	21	4	11	12	4	10	19	47,242	-646,72
10	4	23	27	14	21	21	18	21	7	1	68,249	335,06
11	10	15	22	24	16	27	8	3	11	26	68,250	348,86
12	2	20	1	20	2	24	21	8	19	16	60,092	-928,39
13	22	5	4	16	18	15	7	18	18	14	53,573	-329,94
14	30	13	4	25	16	27	31	3	11	16	79,643	-656,25
15	16	12	10	16	15	12	3	1	31	14	52,448	-549,44
16	12	12	18	15	11	14	25	13	25	5	62,251	148,45
17	32	15	24	6	7	16	30	21	31	14	95,621	-138,14
18	13	4	8	14	7	1	3	11	4	23	33,651	-288,69
19	10	23	24	3	5	23	8	29	16	17	71,919	221,16
20	11	23	18	5	31	22	32	15	20	23	96,831	-128,83
21	24	10	28	31	6	22	10	21	5	4	69,449	236,31
22	21	7	32	20	17	9	17	7	28	14	75,073	-126,15
23	4	26	22	2	11	29	6	13	26	15	68,613	370,46
24	8	4	2	21	31	29	14	9	22	1	62,845	-624,57
25	30	23	6	20	5	21	14	3	20	22	73,866	-518,54
26	30	29	7	13	29	15	20	16	7	20	80,675	-306,75
27	17	32	22	19	12	2	23	18	21	4	78,247	89,70
28	11	4	1	25	24	9	2	24	23	1	59,298	-455,57
29	3	1	14	6	26	18	6	9	18	24	53,410	574,64
30	9	1	21	13	17	10	27	7	4	7	45,950	238,41

Tablo 9’da, 4. kolondan görüleceği üzere birey elemanlarında oluşan gerilmeler müsaade edilen gerilme sınırını aşmışlardır. Bu bireyler daha önceden tanımlanmış olan bir takım ceza fonksiyonlarında değerlendirilerek uyum dereceleri belirlenecektir.

Uyum derecesi yüksek olan bireyler, uyum derecesi düşük olan bireylerin yerine kopyalanacaktır. Yani, daha öncede belirtildiği gibi, Darwin’in evrim teorisine göre güçlü bireyler yaşamaya devam ederken zayıf bireyler ölmektedirler. Zayıf bireylerin ölmesi genetik algoritmada uyum derecesi düşük olan bireylerin o andaki nesilden yani topluluktan dışarı atılmasıyla olmaktadır.

Tablo 10’da düğüm noktalarında oluşan maksimum yerdeğiştirme, bireylerin uyum faktörü değerleri ve kopya sayıları verilmektedir. Ayrıca bu tabloda bireylerin kopya sayılarına göre eşleşme havuzuna alınmaları gösterilmektedir. Sistemdeki tüm düğüm noktaları için müsaade edilen yerdeğiştirme değeri 5,08 cm iken başlangıç neslindeki bireylerin çoğu bu sınırlayıcıyı ihlal etmektedirler. Bunun nedeni başlangıç neslinin rasgele belirlenmesidir.

Bu çalışmada sınırlayıcı olarak yerdeğiştirme, gerilme ve burkulma kullanılmaktadır. Bireylerin bu sınırlayıcıları ihlal edip etmeme durumları genetik algoritma tarafından araştırılıp, uyum faktörleri hesaplanarak, bireylere yaşamlarını devam ettirip ettiremeyeceklerini belirleyen anlamlı sayılar atanmaktadır. Uyum faktörü denen bu sayılar eğer 0.5’den küçükse bireyin yaşamı sona erdirilmekte yani, topluluktan dışarı atılmaktadır. Örnek olarak, Tablo 10’da 2. kolonda uyum faktörleri 0.5 den küçük olan 9., 14. ve 25. bireyin 3. kolonda kopya sayıları sıfırdır. Bu bireyler yerine uyum faktörleri yüksek olan 7., 18. ve 23. bireylerden birer adet kopyalanarak eşleşme havuzundaki yerlerine gönderilmektedirler.

Kopyalama sayesinde eşleşme havuzunda aynı bireyden birden fazla bulunabilmektedir. Kopyalanan bireyler en iyi bireyler olduğu için nesil sayısı arttıkça topluluk içerisinde iyi bireylerin baskınlığı artmaktadır. Bu durum genetik algoritmanın, en uygun tasarımı bulması için başlangıçta gelişigüzel verilen topluluğu sömürmesi anlamına gelmektedir.

Tablo 11’de bireylerin çaprazlama noktaları, mutasyona uğrayan grup ve bu operatörlerin ardından oluşturulan yeni nesil gösterilmektedir.

Tablo 10. On çubuklu sistem için yerdeğişirmeler, bireylerin uyum faktörleri, kopya sayıları ve eşleşme havuzu

Mak. yer-değişirme (cm)	Uyum faktörü	Kopya sayısı	Eşleşme havuzu									
-4,535	1,041	1	25	25	10	8	24	1	18	20	5	5
-3,783	0,960	1	26	1	4	14	18	23	12	19	26	13
-4,210	1,189	1	4	9	27	4	16	12	19	30	8	23
-3,789	0,791	1	32	28	4	6	22	12	4	17	32	30
-6,069	1,031	1	3	6	11	4	26	4	31	2	31	10
-3,492	1,183	1	27	12	19	22	14	2	29	12	7	2
-5,160	1,244	2	3	29	27	14	28	7	16	14	6	9
-10,352	0,623	1	3	29	27	14	28	7	16	14	6	9
-7,872	0,417	0	1	29	4	8	7	7	25	27	17	19
-5,302	1,153	1	4	23	27	14	21	21	18	21	7	1
-5,146	0,866	1	10	15	22	24	16	27	8	3	11	26
-11,671	0,994	1	2	20	1	20	2	24	21	8	19	16
-4,138	1,029	1	22	5	4	16	18	15	7	18	18	14
-6,925	0,206	0	16	12	10	16	15	12	3	1	31	14
-7,422	1,028	1	12	12	18	15	11	14	25	13	25	5
-2,932	1,174	1	32	15	24	6	7	16	30	21	31	14
-1,721	1,169	1	13	4	8	14	7	1	3	11	4	23
-7,307	1,219	2	13	4	8	14	7	1	3	11	4	23
-4,049	1,138	1	10	23	24	3	5	23	8	29	16	17
-2,710	1,184	1	11	23	18	<u>5</u>	<u>31</u>	<u>22</u>	<u>32</u>	<u>15</u>	20	23
-3,471	0,997	1	24	10	28	31	6	22	10	21	5	4
-2,194	1,110	1	21	7	32	20	17	9	17	7	28	14
-6,914	1,208	2	4	26	22	2	11	29	6	13	26	15
-8,559	1,053	1	4	26	22	2	11	29	6	13	26	15
-6,163	0,453	0	8	4	2	21	31	29	14	9	22	1
-3,847	1,051	1	30	<u>29</u>	7	13	29	15	20	16	7	20
-2,151	1,161	1	17	32	22	19	12	2	23	18	21	4
-6,118	1,150	1	11	4	1	25	24	9	2	24	23	1
-8,352	1,177	1	3	1	14	<u>6</u>	<u>26</u>	<u>18</u>	<u>6</u>	<u>9</u>	18	24
-4,575	1,127	1	9	1	21	13	17	10	27	7	4	7



Eşleşme havuzunda bireyler kendilerine rasgele eş seçerler. Seçilen bu eşler arasında yine rasgele 2 adet çaprazlama noktası seçilir. Bu çaprazlama noktaları arasında kalan dizi parçaları eş bireyler arasında karşılıklı olarak değiştirilir. Örneğin, eşleşme havuzunda 20 nolu birey 29 nolu bireyle eşleşip, çaprazlama noktaları Tablo 11'de görüldüğü gibi 3 ve 8 olarak rasgele seçilmiştir.

Eşleşme havuzunda 20 nolu bireyin 3 ile 8 arasında kalan 4., 5., 6., 7. ve 8. grupları (altı çizili sayılar) yani 5-31-22-32-15 şeklinde olan dizi parçası, 29 nolu bireyin yine aynı noktalar arasında kalan 6-26-18-6-9 şeklindeki dizi parçasıyla karşılıklı olarak değiştirilmiştir. Böylece eşleşme havuzunda 11-23-18-5-31-22-32-15-20-23 şeklinde olan 20 nolu birey eşleşme ve çaprazlanmadan sonra Tablo 11'de 6. kolondan görüldüğü gibi 11-23-18-6-26-18-6-9-20-23 (altı çizili sayılar) haline dönüşerek yeni nesildeki yerini almaktadır.

Mutasyon, daha öncede belirtildiği gibi herhangi bir bireyin herhangi bir grubunun rasgele değiştirilmesiyle yapılmaktadır. Literatürde genellikle mutasyon oranının düşük alınması vurgulandığı için on çubuklu bu sistemde, 30 bireyde sadece tek bir bireyin yine tek bir elemanını değiştirmek suretiyle mutasyon gerçekleştirilmiştir.

Eşleşme havuzunda 26. bireyin 2. grubundaki eleman (29) mutasyona uğradıktan sonra Tablo 11'de görüldüğü gibi 13 olmuştur. Mutasyon operatörü çaprazlamadan sonra yapıldığı için aynı bireyin bazı elemanları iki defa değişmiş olabilmektedir. Bu nedenle mutasyona uğramış bireyin mutasyondan önceki durumu çaprazlamadan önce görülemeyebilmektedir. Mutasyon çaprazlanmış bireyler üzerinde gerçekleşmektedir.

Tablo 12'de optimum tasarımın elde edildiği topluluk bulunmaktadır. Optimum tasarım 94. nesilde oluşmuştur. Bu son nesildeki bireylerin hemen hemen hepsi aynı tür bireyden oluşmaktadır. Başlangıçta optimum tasarım için %80'lik bir yakınsama kabul edildiği için 30 bireyden oluşan topluluğun son nesilde 24 bireyinin aynı bireyler olması beklenmektedir. Yine bu tablodan görüldüğü gibi 24 adet bireyin 12-5-22-14-11-7-17-21-4-23 sıra nolu elemanlardan oluştuğu görülür. Bu da genetik algoritmanın başlangıç neslinden itibaren 94 nesil boyunca topluluğu sömürerek optimum tasarımı bulduğunu ifade etmektedir. Optimum tasarım olarak kabul edilen bu bireyin gerilme ve yerdeğiştirme sınırlayıcılarını ihlal etmediği de yine bu tablonun 4. ve 5. sütunlarından anlaşılmaktadır. Sistem için müsaade edilen maksimum gerilme 172 Mpa ve yerdeğiştirme ise 5,08 cm iken optimum tasarımı veren bu bireyde oluşan en büyük gerilme 168,25 Mpa ve en büyük yerdeğiştirme 3,72 cm dir.

Tablo 11. On çubuklu sistem için çaprazlama, mutasyon ve yeni nesil oluşumu

Birey no	Eş birey	Çaprazlama noktaları		Mutasyon	Yeni nesil									
		1.	2.											
1	17	5	6		25	25	10	8	24	1	18	20	5	5
2	24	3	5		26	1	4	2	11	23	12	19	26	13
3	21	3	9		4	9	27	31	6	22	10	21	5	23
4	13	5	5		32	28	4	6	22	12	4	17	32	30
5	23	1	6		3	26	22	2	11	29	31	2	31	10
6	26	1	6		27	29	7	13	29	15	29	12	7	2
7	16	8	10		3	29	27	14	28	7	16	14	31	14
8	9	2	6		3	29	4	8	7	7	16	14	6	9
9	8	2	6		1	29	27	14	28	7	25	27	17	19
10	27	3	6		4	23	27	19	12	2	18	21	7	1
11	12	5	7		10	15	22	24	16	24	21	3	11	26
12	11	5	7		2	20	1	20	2	27	8	8	19	16
13	4	5	5		22	5	4	16	18	15	7	18	18	14
14	28	6	7		16	12	10	16	15	12	2	1	31	14
15	30	6	6		12	12	18	15	11	14	25	13	25	5
16	7	8	10		32	15	24	6	7	16	30	21	6	9
17	1	5	6		13	4	8	14	7	1	3	11	4	23
18	22	6	7		13	4	8	14	7	1	17	11	4	23
19	25	1	10		10	4	2	21	31	29	14	9	22	1
<u>20</u>	<u>29</u>	3	8		<u>11</u>	<u>23</u>	<u>18</u>	<u>6</u>	<u>26</u>	<u>18</u>	<u>6</u>	<u>9</u>	<u>20</u>	<u>23</u>
21	3	3	9		24	10	28	4	16	12	19	30	8	4
22	18	6	7		21	7	32	20	17	9	3	7	28	14
23	5	1	6		4	6	11	4	26	4	6	13	26	15
24	2	3	5		4	26	22	14	18	29	6	13	26	15
25	19	1	10		8	23	24	3	5	23	8	29	16	17
26	6	1	6	2.Grup	30	<u>13</u>	19	22	14	2	20	16	7	20
27	10	3	6		17	32	22	14	21	21	23	18	21	4
28	14	6	7		11	4	1	25	24	9	3	24	23	1
29	20	3	8		3	1	14	5	31	22	32	15	18	24
30	15	6	6		9	1	21	13	17	10	27	7	4	7

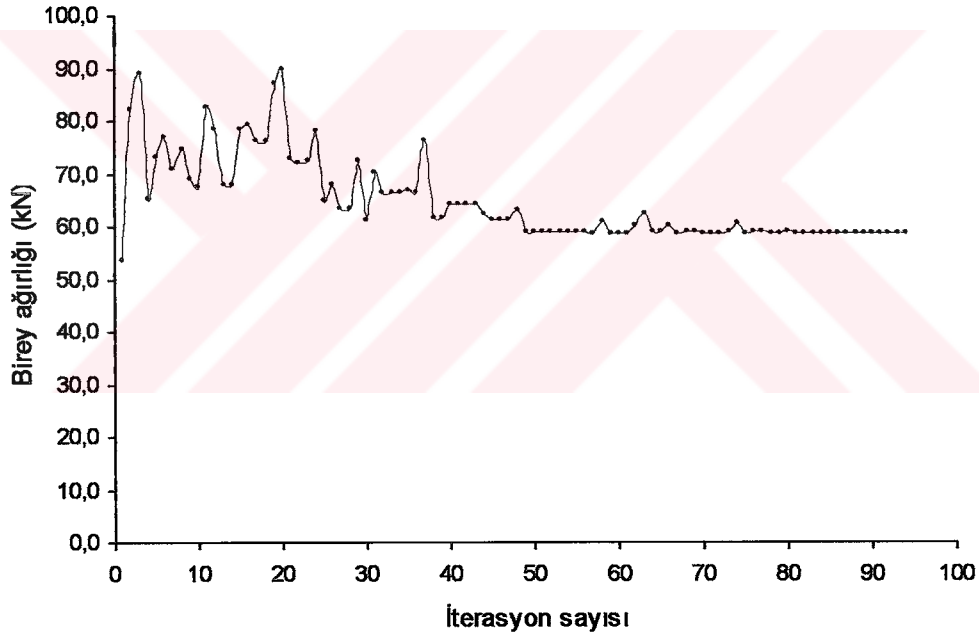


Tablo 12. On çubuklu sistem için optimum tasarım sonuçları

Birey no	Son nesil										W (kN)	$\sigma$ (Mpa)	U (cm)	C	$\Phi(x)$
1	12	5	22	14	11	7	17	21	4	23	58,40	168,25	3,72	0,0	58,40
2	12	5	22	14	11	7	17	21	4	23	58,40	168,25	3,72	0,0	58,40
3	12	6	22	14	11	7	17	21	4	23	58,53	164,12	3,68	0,0	58,53
4	12	5	22	14	11	7	17	21	4	23	58,40	168,25	3,72	0,0	58,40
5	12	5	22	14	11	7	17	21	4	23	58,40	168,25	3,72	0,0	58,40
6	12	5	22	14	11	7	17	21	4	23	58,40	168,25	3,72	0,0	58,40
7	12	5	22	14	11	7	17	21	4	23	58,40	168,25	3,72	0,0	58,40
8	12	5	22	14	11	7	17	21	4	23	58,40	168,25	3,72	0,0	58,40
9	12	5	22	14	11	7	17	21	4	23	58,79	167,75	3,72	0,0	58,79
10	12	5	22	14	11	7	18	21	4	23	58,40	168,25	3,72	0,0	58,40
11	12	5	22	14	11	7	17	21	4	23	58,40	168,25	3,72	0,0	58,40
12	12	5	22	14	11	7	17	21	4	23	58,40	168,25	3,72	0,0	58,40
13	12	6	22	14	11	7	17	21	4	23	58,53	164,11	3,68	0,0	58,53
14	12	5	22	14	11	7	17	21	4	23	58,40	168,25	3,72	0,0	58,40
15	12	5	22	14	11	7	17	21	4	23	58,40	168,25	3,72	0,0	58,40
16	12	5	22	14	11	7	17	21	4	23	58,40	168,25	3,72	0,0	58,40
17	12	5	22	15	11	7	17	21	4	23	58,93	169,18	3,72	0,0	58,93
18	12	5	22	14	11	7	17	21	4	23	58,40	168,25	3,72	0,0	58,40
19	12	5	22	14	11	7	17	21	4	23	58,40	168,25	3,72	0,0	58,40
20	12	6	22	14	11	7	17	21	4	23	58,53	164,11	3,68	0,0	58,53
21	12	5	22	14	11	7	17	21	4	23	58,40	168,25	3,72	0,0	58,40
22	12	5	22	14	11	7	17	21	4	23	58,40	168,25	3,72	0,0	58,40
23	12	5	22	14	11	7	17	21	4	23	58,40	168,25	3,72	0,0	58,40
24	12	5	22	14	11	7	17	21	4	23	58,40	168,25	3,72	0,0	58,40
25	12	5	22	14	11	8	17	21	4	23	58,68	163,88	3,68	0,0	58,68
26	12	5	22	14	11	7	17	21	4	23	58,40	168,25	3,72	0,0	58,40
27	12	5	22	14	11	7	17	21	4	23	58,40	168,25	3,72	0,0	58,40
28	12	5	22	14	11	7	17	21	4	23	58,40	168,25	3,72	0,0	58,40
29	12	5	22	14	11	7	17	21	4	23	58,40	168,25	3,72	0,0	58,40
30	12	5	22	14	11	7	17	21	4	23	58,40	168,25	3,72	0,0	58,40

Yine bu tablonun 6. sütunundan görüldüğü gibi, bireylere ait ceza katsayıları sıfır olup hiçbir birey ceza almamıştır. Çünkü bunlar nesiller boyunca hep iyi bireylerin kopyalanması sonucunda oluşmuştur. Cezalandırılmış amaç fonksiyonu (7.sütun), birey toplam ağırlığının ve ceza katsayısının bir fonksiyonu olduğundan ve son nesilde hiçbir birey ceza olmadığından dolayı doğrudan birey ağırlığına (3. sütun) eşit olmaktadır.

Daha önceden de belirtildiği gibi, genetik algoritmalar optimum tasarımı bulacaklarını garanti edemeyip, diğer optimizasyon tekniklerine göre daha kısa zamanda daha pratik çözümler verebilmekte ancak bazen de sonuç veremeyebilmektedir. Zira belirli bir yakınsama oranı içerisinde bulunduğu çözümü optimum tasarım olarak vermektedir. Şekil 11'de %80 yakınsama oranı kabul edilen bu sistem için herhangi bir bireyin ağırlığının iterasyon sayısı ile değişimi gösterilmektedir.



Şekil 11. Birey ağırlığının iterasyon sayısı ile değişimi

Bu şekilden görüldüğü gibi, ilk iterasyonlarda değişken olan birey ağırlığı, iterasyon sayısı arttıkça sabitleşmekte ve 50. iterasyondan sonra pratik olarak değişmemekte olup mutasyon etkisinden dolayı grafik çok az dalgalanmaktadır. Bu da genetik algoritmanın ne kadar etkin çalıştığını göstermektedir.

Böylece sabitleşen eğrinin karşılık geldiği değer (  $W=58,40$  kN ) optimum tasarıma sahip olan bireyin toplam ağırlığı olarak alınmaktadır. Tablo 13’de, 10 çubuklu bu sistem için yapılan burkulmalı tasarım sonuçlarının literatürdeki bir çalışmayla karşılaştırılması verilmektedir.

Tablo 13. On çubuklu sistemin burkulmalı tasarım sonuçlarının karşılaştırılması

Eleman no	Enkesit alanları (cm <sup>2</sup> )	
	Bu çalışma	Groenwold vd., [30]
1	45,68	53,23
2	17,10	30,19
3	136,13	149,68
4	58,90	90,97
5	39,74	30,19
6	20,52	30,19
7	87,10	66,45
8	123,23	149,68
9	15,15	58,90
10	143,87	123,23
Toplam ağırlık (kN)	58,404	65,859

Bu tablodan görüldüğü gibi, bu çalışmada değer kodlaması yapılarak belirlenen toplam ağırlık Groenwold vd., [30]’un çalışmasında verilenlerden daha küçüktür. Yani en az ağırlıklı tasarımı arzulayan amaç fonksiyonu için daha ideal bir çözüm bulunmuştur.

### 2.5.1.2. On Çubuklu Sistemin Burkulmasız Tasarımı

On çubuklu sistemin burkulmasız tasarımını gerçekleştirirken sistemle ilgili genel veriler burkulmalı tasarımdakinin aynısıdır. Farklı olarak çelik elemanların birim hacim ağırlığı  $\rho = 0,02714 \text{ N/cm}^3$  (  $0,1 \text{ lb/in}^3$  ) ve elastisite modülü  $E=68950 \text{ Mpa}$  ( $10^4 \text{ ksi}$ ) alınmış olup kullanılan tasarım değişkenleri enkesit alanları Tablo 14’de verilmektedir. Bu çalışmada değer kodlaması kullanıldığından tasarım değişkenleri toplam sayısının ikinin katları şeklinde olması gerekmektedir.

Tablo 14. On çubuklu sistemin burkulmasız tasarımı için kullanılan enkesit alanları

Enkesit sıra no	Alan (cm <sup>2</sup> )	Enkesit sıra no	Alan (cm <sup>2</sup> )
1	10,45	22	30,96
2	11,61	23	32,06
3	12,84	24	33,03
4	13,74	25	37,03
5	15,35	26	46,58
6	16,90	27	51,42
7	18,58	28	74,19
8	18,90	29	87,10
9	19,94	30	89,68
10	20,19	31	91,61
11	21,81	32	100,00
12	22,39	33	103,23
13	22,90	34	109,03
14	23,41	35	121,29
15	24,77	36	128,39
16	24,97	37	141,94
17	25,03	38	147,74
18	26,97	39	170,97
19	27,23	40	193,55
20	28,97	41	216,13
21	29,61	—	—

On çubuklu sistemin burkulmasız tasarımı için genetik programlamada 30 adet birey rasgele oluşturulup %80 yakınsama dikkate alınarak en fazla 200 iterasyona izin verilmiştir. Cezalandırılmış amaç fonksiyonundaki ceza katsayısı 3.5 olarak alınmıştır. Program 148. iterasyonda %80 yakınsamaya ulaşarak sistem ağırlığını  $W=24,849$  kN olarak belirlemiştir. Ayrıca sistemde düğüm noktalarının yerdeğiştirilmesi en fazla 5,06 cm olup bu değer müsaade edilen yerdeğiştirme değerinden küçüktür. Sistem için bulunan sonuçlar literatürde yapılan bir çalışmayla karşılaştırılmalı olarak Tablo 15’de verilmektedir.

Tablo 15. On çubuklu sistemin burkulmasız tasarım sonuçlarının karşılaştırılması

Eleman no	Enkesit alanları (cm <sup>2</sup> )	
	Bu çalışma	Rajeev vd., [29]
1	216,13	216,13
2	13,74	10,45
3	147,74	141,90
4	91,61	100,00
5	10,45	10,45
6	15,35	11,61
7	51,42	91,61
8	141,94	128,39
9	147,74	128,39
10	16,90	16,90
Toplam ağırlık (kN)	24,849	24,998

Burkulmasız tasarımdan bulunan sonuçlarda birey ağırlığı daha azdır. Bunun nedeni burkulmalı tasarımda basınç çubuklarının narinliğinden dolayı daha büyük enkesit alanlarına ihtiyaç duyulmasıdır.

Bu tablodan görüldüğü gibi bu çalışmadan elde edilen toplam ağırlık, diğerlerinde verilenlerden daha azdır.

### 2.5.1.3. On Çubuklu Sistemin TS-648'e Göre Burkulmalı Tasarımı

On çubuklu sistemin son olarak TS-648'e [36] göre burkulmalı tasarımı gerçekleştirilmiştir. Bunun için yine ilk durumda AISC-ASD'ye göre tasarım yapılırken kullanılan veriler alınmıştır. Yine 30 adet birey oluşturulup, en fazla 200 iterasyona izin verilmiştir. Ceza katsayısı her probleme göre farklı değerler alabilmektedir. Bu çalışmada ceza fonksiyonunun alacağı değerler deneme yanılma yoluyla belirlendiğinden cezalandırılmış amaç fonksiyonu için ceza katsayısı 10.0 olarak alınmıştır.

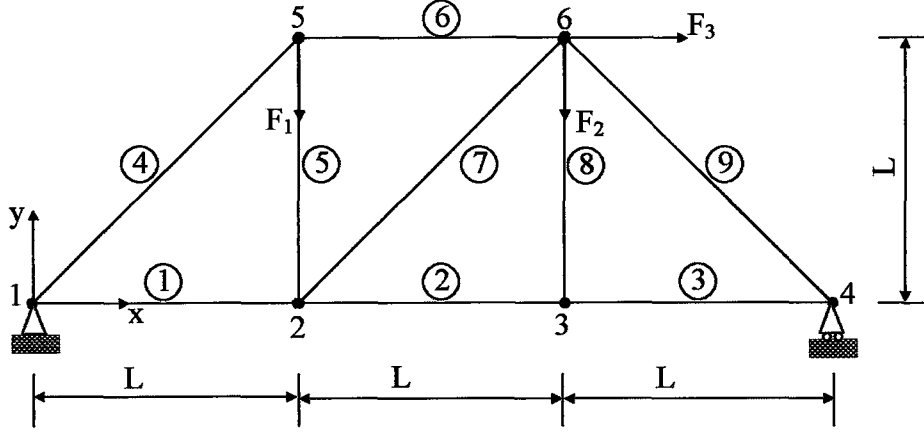
Genetik algoritma programı, 117. iterasyonda %80 lik yakınsamayla,  $W=62,559$  kN ağırlıklı bir tasarım elde etmiştir. Optimum tasarım sayılabilecek bu tasarımda gerilme ve yerdeğiştirme sınırlayıcıları ihlal edilmemektedir. Düğüm noktalarında oluşan en büyük yerdeğiştirme  $3,413$  cm'dir. Tablo 16'da on çubuklu bu sistemin TS-648'e göre burkulmalı tasarımından elde edilen sonuçlar yine literatürdeki bir çalışmayla karşılaştırmalı olarak sunulmuştur.

Tablo 16. On çubuklu sistemin TS-648'e göre burkulmalı tasarım sonuçlarının karşılaştırılması

Eleman no	Enkesit alanları (cm <sup>2</sup> )	
	Bu çalışma	Ayvaz vd., [37]
1	45,68	53,23
2	17,10	30,19
3	187,74	149,68
4	75,48	90,97
5	17,10	35,74
6	24,71	10,45
7	87,10	53,23
8	136,13	181,94
9	17,10	35,74
10	136,13	123,23
Toplam ağırlık (kN)	62,559	64,439

### 2.5.2. Dokuz Çubuklu Düzlem Kafes Sistem

Bu tasarıma konu olan 9 çubuklu düzlem kafes sistem Şekil 12’de verilmektedir.



Şekil 12. Dokuz çubuklu düzlem kafes sistem

Bu sistem, 5 nolu düğüm noktasında negatif y yönünde  $F_1=22,44$  kN (5000 lbs), 6 nolu düğüm noktasında ise negatif y yönünde  $F_2=8,896$  kN (2000 lbs), ve pozitif x yönünde  $F_3=31,136$  kN (7000 lbs) değerinde kuvvetlere maruzdur.

İzin verilen emniyet gerilmesi çekme ve basınç çubuklarında  $\sigma=68,95$  Mpa (10 kpsi) dir. Düğüm noktalarında müsaade edilen maksimum yerdeğiştirme 0,127 cm (0,05 in) dir. Elastisite modülü  $E=206850$  Mpa ( $30 \times 10^6$  psi), çubuk elemanlar için birim hacim ağırlık  $\rho=0,07682$  N/cm<sup>3</sup> (0,283 lb/in<sup>3</sup>) ve  $L=25,4$  cm’dir.

Bu sistemde çok fazla sayıda eleman olmadığından grup sayısı eleman sayısına eşit alındığı için bir bireyi oluşturan dizi 9 adet sayıdan oluşmaktadır. Tasarım değişkeni olarak 50 adet enkesit alanı kullanılmaktadır. Bu sistem için burkulmasız olarak tasarım yapıp, sürekli değişkenlerin ayrık değişken olarak düşünülmesiyle, literatürde 0,1 in<sup>2</sup> den 5,0 in<sup>2</sup> arasında değişen enkesit alan değerleriyle oluşturulmuştur. Tablo 17’de bu sistem için kullanılan enkesit alanları verilmektedir.

Bu sistemin optimum tasarımını genetik algoritma ile gerçekleştirirken 20 adet birey oluşturulmuştur. Maksimum 200 iterasyona izin verilip %85’lik bir yakınsama oranı yeterli görülmüştür. Cezalandırılmış amaç fonksiyonu için ceza katsayısı 5,0 olarak alınmıştır.

Programın kořturulmasıyla genetik algoritma 68. iterasyonda %85 yakınsama oranına ulařarak,  $W=0,134$  kN'luk bir tasarım elde edilmiřtir.

Tablo 17. Dokuz çubuklu sistemde tasarım için kullanılan enkesit alanları

Enkesit sıra no	Alan (cm <sup>2</sup> )	Enkesit sıra no	Alan (cm <sup>2</sup> )	Enkesit sıra no	Alan (cm <sup>2</sup> )
1	0,65	18	11,63	35	22,58
2	1,29	19	12,26	36	23,23
3	1,94	20	12,91	37	23,87
4	2,58	21	13,55	38	24,52
5	3,23	22	14,19	39	25,16
6	3,87	23	14,84	40	25,81
7	4,52	24	15,48	41	26,45
8	5,16	25	16,13	42	27,09
9	5,81	26	16,78	43	27,74
10	6,45	27	17,42	44	28,39
11	7,10	28	18,06	45	29,03
12	7,74	29	18,71	46	29,68
13	8,38	30	19,35	47	30,32
14	9,03	31	20,00	48	30,97
15	9,68	32	20,65	49	31,61
16	10,32	33	21,29	50	32,26
17	10,97	34	21,94		

Bulunan tasarım sonuçları, literatürde yapılan bir çalıřmayla karřılařtırılması amacıyla Tablo 18'de verilmektedir. Bu tablodan görüldüğü gibi, bu çalıřmadan bulunan eleman enkesit alanları ve yapı toplam ağırlığı literatürde verilen tasarım sonuçlarından daha küçüktür.

Literatürde yapılan çalıřmalarda burkulmasız tasarımlar için sürekli deęiřkenler ayrıık deęiřken olarak kullanılarak tasarım deęiřkenleri oluşturulmuřtur. Bundan ötürü yukarıdaki tabloda mevcut olan tasarım sonuçları enkesit alanları pratikte mevcut olmayabilir.

Genetik algoritmayı cezalandırılmıř amaç fonksiyonu yönlendirmektedir. Topluluktaki bir bireyin topluma uyum derecesini belirlediği ve gerektiğinde de bireyin topluluktan atılmasında rol oynadıđı için bu fonksiyon genetik algoritmanın bařlangıç neslini sömürerek en iyi bireyi yani optimum tasarımı bulmasını sađlamaktadır.



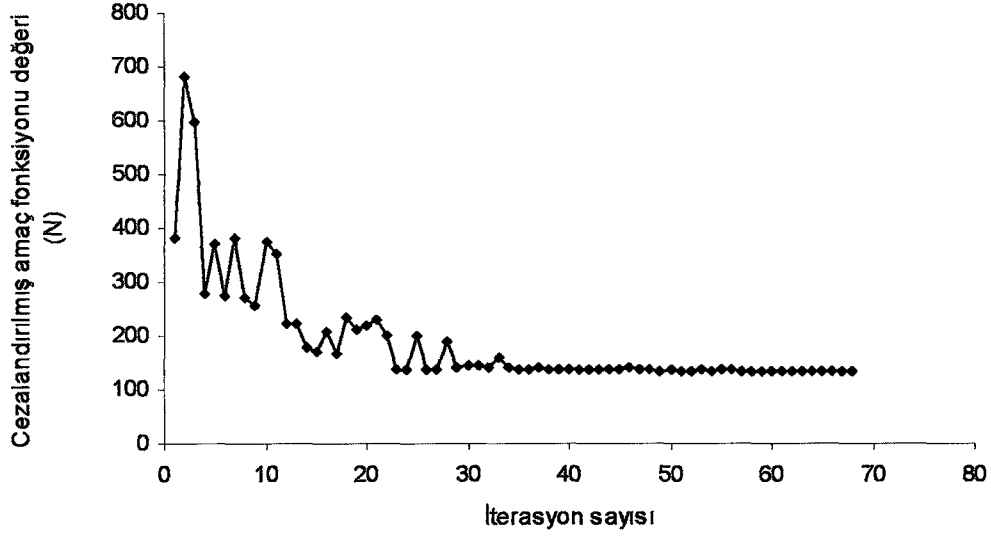
Tablo 18. Dokuz çubuklu düzlem kafes sistemin tasarım sonuçlarının karşılaştırılması

Eleman no	Bu çalışma	Arun [28]
1	6,45	19,03
2	1,29	8,77
3	0,64	16,97
4	9,68	9,80
5	10,32	12,90
6	5,80	5,74
7	10,96	0,65
8	1,29	0,65
9	9,67	6,77
W (kN)	0,134	0,172

Genetik algoritmada optimum tasarıma yaklaştıkça artık bireyler çoğunlukla iyi uyumlu olacağı için cezalandırılmış amaç fonksiyonunun giderek azalması ve sonuçta sabit bir değerde kalması beklenmektedir. Bu sabit değer optimum tasarımı veren bireyin toplam ağırlığıdır. Çünkü optimum tasarım hiçbir sınırlayıcıyı ihlal etmediğinden ceza faktörü sıfır olmaktadır. Cezalandırılmış amaç fonksiyonu (27) nolu bağıntıdan da görüldüğü gibi bireyin toplam ağırlığı ile o bireyin almış olduğu cezaların bileşik bir fonksiyonu olduğu için ceza almamış bir bireyin cezalandırılmış amaç fonksiyonu değeri o bireyin ağırlığına eşit olmaktadır.

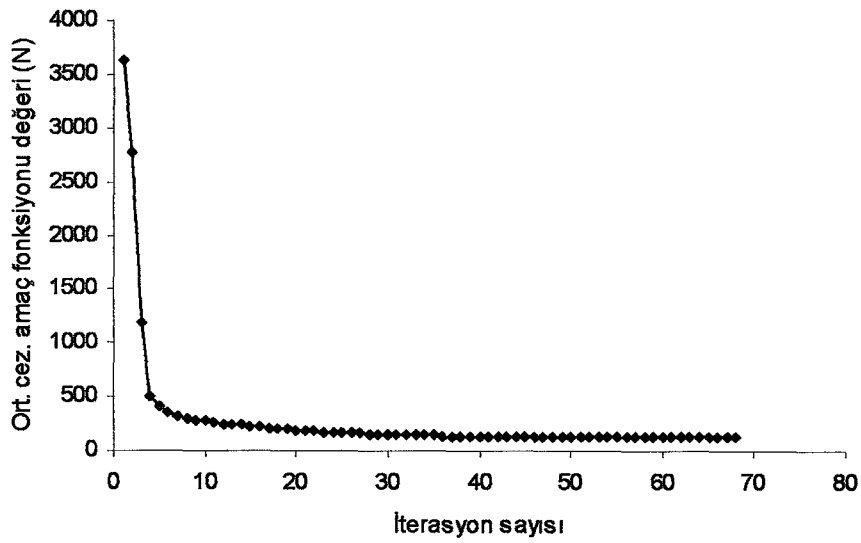
Bir toplulukta herhangi bir birey başlangıçta ne kadar uyumsuz olursa olsun topluluktan dışarı atılmadığı sürece genetik operatörler sayesinde son nesile doğru yaklaştıkça uyumlu bir birey haline gelebilir. Dolayısıyla yukarıda da bahsedildiği gibi ilk nesillerde cezalandırılmış amaç fonksiyonu değeri büyük olsa bile optimizasyon süreci sonunda azalarak bir değerde sabitleşmektedir. Eğer bu birey optimum tasarımı veren bireylerden biri olursa cezalandırılmış amaç fonksiyonunun değeri birey toplam ağırlığına eşit olmaktadır. Aksi halde bu fonksiyonun değeri en iyi bireyin toplam ağırlığına yakın bir değer almaktadır.

Şekil 13'de optimum tasarıma sahip 7. bireyin iterasyona bağlı olarak cezalandırılmış amaç fonksiyonunun değişimi görülmektedir. Bu şekilden görüldüğü gibi başlangıç neslinde  $W=381$  N değerinde olan cezalandırılmış amaç fonksiyonu değeri otuz iterasyon boyunca dalgalı bir eğri oluşturmaktadır. 36. iterasyondan itibaren 7. bireyin cezalandırılmış amaç fonksiyonu değeri  $W=134$  N değerinde sabitleşmiştir. Bu değer optimum tasarıma sahip bireyin toplam ağırlığıdır.



Şekil 13. Optimum tasarıma sahip 7. bireyin iterasyona bağlı olarak cezalandırılmış amaç fonksiyonun değeri

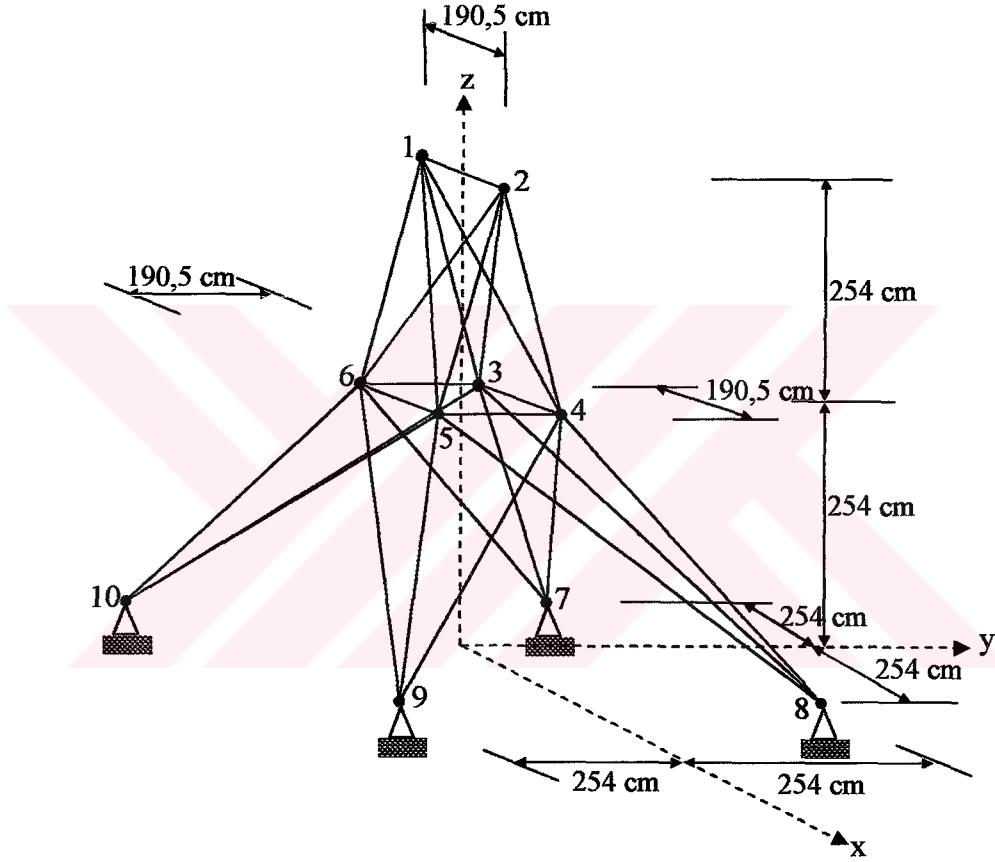
Şekil 14'de ortalama cezalandırılmış amaç fonksiyonunun iterasyona bağlı olarak değişimi gösterilmektedir. Her nesilde bireylerin cezalandırılmış amaç fonksiyonu değerleri ortalaması alındığı için ilk iterasyonlarda çok büyük değerler çıkabilmektedir. Başlangıç nesli bilgisayar tarafından rasgele oluşturulduğundan uyumsuz bireyler ortalamayı oldukça fazla etkilemektedirler. Ancak iterasyon sayısı arttıkça uyumsuz bireyler topluluktan dışlandığı için ortalama belli bir değerde sabitleşmeye başlar.



Şekil 14. Ortalama cezalandırılmış amaç fonksiyonunun iterasyon sayısına bağlı olarak değişimi

### 2.5.3. Yirmibeş Çubuklu Sistem

Bu tasarıma konu olan sistemle ilgili geometrik bilgiler Şekil 15’de verilmektedir. Burkulmasız olarak tasarlanan bu sistem, literatürde iki farklı yükleme durumu altında ele alınmaktadır. Bu çalışmada sadece Tablo 19’da verilen yükleme durumu göz önüne alınarak tasarım yapılmaktadır.



Şekil 15. Yirmibeş çubuklu uzay kafes sistem

Tablo 19. Yirmibeş çubuklu sisteme uygulanan kuvvetler

Yüklü nokta	$F_x$ (kN)	$F_y$ (kN)	$F_z$ (kN)
1	4,448	-44,480	-44,480
2	0,000	-44,480	-44,480
3	2,224	0,000	0,000
6	2,669	0,000	0,000

Bu sistemin çözümünde gruplandırma yapılmış olup oluşturulan 8 adet grubun hangi elemanlardan meydana geldiği Tablo 20'de verilmektedir.

Tablo 20. Yirmibeş çubuklu sistem için gruplandırma sonuçları

Eleman no	Grup no	Eleman uç noktaları		Eleman no	Grup no	Eleman uç noktaları	
		1. nokta	2. nokta			1. nokta	2. nokta
1	1	1	2	14	6	3	10
2	2	1	4	15	6	6	7
3	2	2	3	16	6	4	9
4	2	1	5	17	6	5	8
5	2	2	6	18	7	4	7
6	3	2	4	19	7	3	8
7	3	2	5	20	7	5	10
8	3	1	3	21	7	6	9
9	3	1	6	22	8	6	10
10	4	6	3	23	8	3	7
11	4	5	4	24	8	4	8
12	5	3	4	25	8	5	9
13	5	6	5				

Sistemin yapısal çözümlemesiyle ilgili olarak elastisite modülü  $E=68950 \text{ Mpa}$  ( $10^7 \text{ psi}$ ) ve çubuk elemanların birim hacim ağırlığı  $\rho=0,02714 \text{ N/cm}^3$  ( $0,11\text{lb/in}^3$ ) olarak alınmıştır. Sistemde 1 ve 2 nolu düğüm noktalarında x ve y doğrultularında  $0.889 \text{ cm}$  ( $0.35 \text{ in}$ ) yerdeğiştirmeye müsaade edilmektedir. Çekme ve basınç çubuklarında müsaade edilen emniyet gerilmesi ise Tablo 21'de verilmektedir.

Tablo 21. Yirmi çubuklu sistem için emniyet gerilmeleri

Grup no	Eleman no	Basınç (Mpa)	Çekme (Mpa)
1	1	241,96 (35,092 ksi)	241,33 (35,0 ksi)
2	2, 3, 4, 5	79,91 (11,59 ksi)	241,33 (35,0 ksi)
3	6, 7, 8, 9	119,35 (17,31 ksi)	241,33 (35,0 ksi)
4	10, 11	241,96 (35,09 ksi)	241,33 (35,0 ksi)
5	12, 13	241,96 (35,09 ksi)	241,33 (35,0 ksi)
6	14, 15, 16, 17	46,85 (6,759 ksi)	241,33 (35,0 ksi)
7	18, 19, 20, 21	46,85 (6,759 ksi)	241,33 (35,0 ksi)
8	22, 23, 24, 25	76,40 (11,08 ksi)	241,33 (35,0 ksi)

Bu sistemin tasarımında kullanılan 30 adet enkesit alanı ise Tablo 22’de verilmektedir. Bu sistem için 30 adet birey oluşturulmuştur. Genetik algoritma sürecinde en fazla 200 iterasyona izin verilip, yakınsama oranı %80 ve cezalandırılmış amaç fonksiyonu için ceza katsayısı 5,0 olarak alınmıştır. Program 144. iterasyonda  $W=2,334$  kN’luk bir birey ağırlığı bulmuştur. Tasarım sonuçları Tablo 23’de literatürden alınan bir çalışmanın tasarım sonuçlarıyla karşılaştırmalı olarak sunulmaktadır.

Tablo 22. Yirmibeş çubuklu sistemin tasarımında kullanılan enkesit alanları

Enkesit sıra no	Alan (cm <sup>2</sup> )	Enkesit sıra no	Alan (cm <sup>2</sup> )
1	0,65	16	10,32
2	1,29	17	10,96
3	1,94	18	11,61
4	2,58	19	12,26
5	3,23	20	12,90
6	3,87	21	13,55
7	4,52	22	14,19
8	5,16	23	14,84
9	5,80	24	15,48
10	6,45	25	16,13
11	7,10	26	16,77
12	7,74	27	18,06
13	8,39	28	19,35
14	9,03	29	20,65
15	9,68	30	21,94

Tablo 23. Yirmibeş çubuklu sistemin tasarım sonuçlarının karşılaştırılması

Grup no	Elemanlar	Enkesit alanları (cm <sup>2</sup> )	
		Bu çalışma	Adeli vd., [31]
1	1	0,65	0,65
2	2, 3, 4, 5	12,90	11,61
3	6, 7, 8, 9	16,13	14,84
4	10, 11	0,65	1,29
5	12, 13	1,29	0,65
6	14, 15, 16, 17	4,52	5,16
7	18, 19, 20, 21	7,74	11,61
8	22, 23, 24, 25	20,65	19,35
Toplam ağırlık (kN)		2,334	2,428

#### 2.5.4. Yetmişiki Çubuklu Uzak Kafes Sistem

Yetmişiki çubuklu bu sistemin geometrik özellikleri Şekil 16'da verilmektedir. Şekil üzerinde daire içindeki sayılar eleman nolarını, diğer sayılar ise düğüm noktalarını göstermektedir.

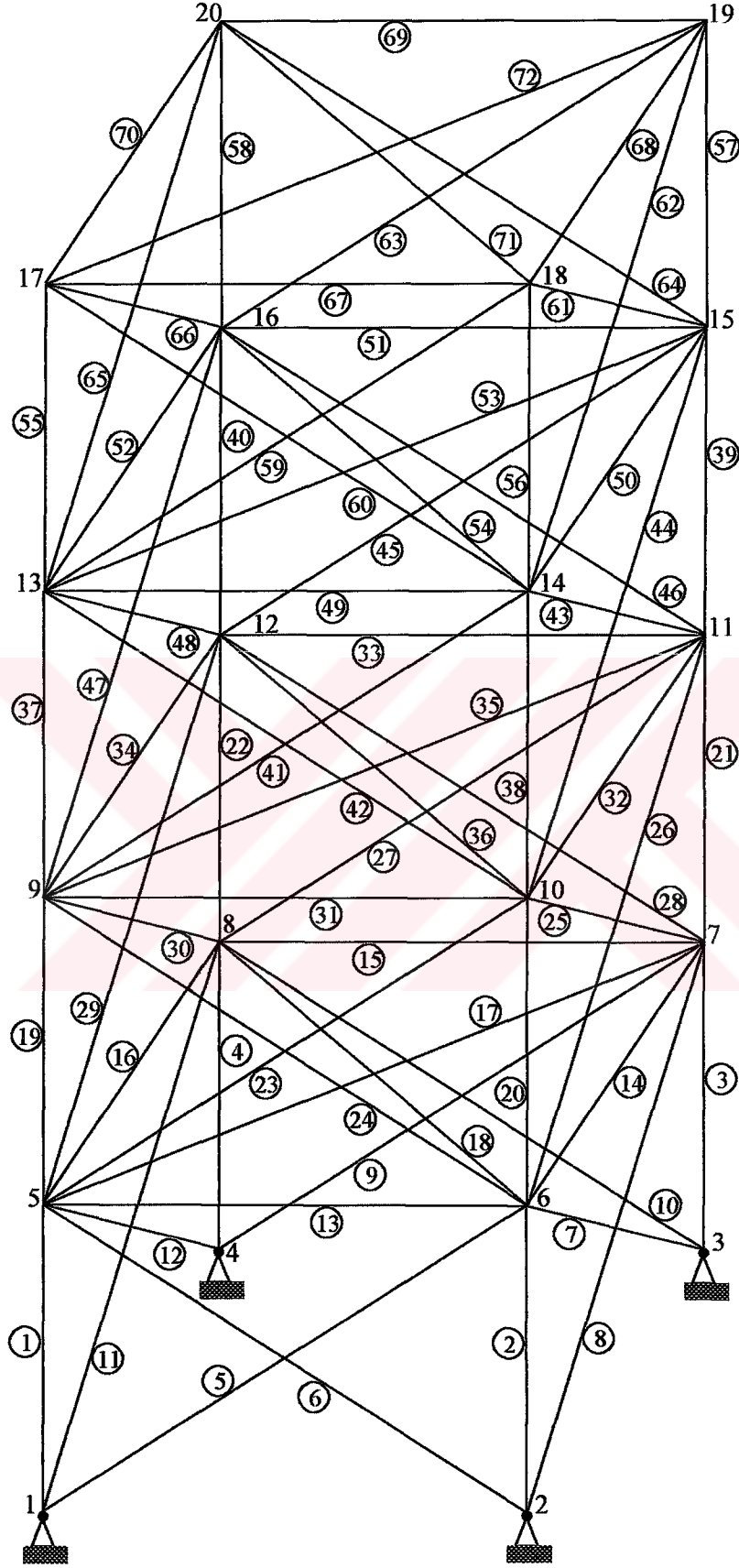
Yetmişiki çubuklu uzak kafes sistem için elastisite modülü  $E=68950$  Mpa (10000 ksi), malzeme birim hacim ağırlığı  $\rho=0,02714$  N/cm<sup>3</sup> (0,1 lb/in<sup>3</sup>), burkulmasız tasarım yapıldığından dolayı çelik çubuklarda basınç ve çekme emniyet gerilmesi birbirine eşit olarak  $\sigma=172,375$  Mpa ( $\mp 25$  ksi) alınmaktadır. Sistemde 17, 18, 19, 20 nolu düğüm noktalarında en fazla 0,635 cm (0,25 in) yerdeğiştirmeye izin verilmektedir.

Tasarım değişkeni olarak 32 adet enkesit alanı kullanılmıştır. Başlangıç neslinde 30 adet birey oluşturulup, programın çalıştırılması sırasında %80 yakınsamayla en fazla 200 iterasyona izin verilmiştir. Cezalandırılmış amaç fonksiyonu için ceza katsayısı 3,484 olarak deneme yanılma yoluyla belirlenmiştir.

İki farklı yüklemeye maruz sistemin yükleme durumları Tablo 24'de verilmektedir. Genetik algoritma programı her iki yükleme durumu için de koşturularak elverişsiz duruma göre tasarım sonuçları kaydedilmiştir.

Bireyin dizi uzunluğunun fazla olmaması için gruplandırmaya gidilip, 72 çubuk eleman için 16 grup oluşturulmuştur. Böylece bir birey, 72 yerine 16 adet sayının yan yana dizilmesiyle meydana gelmektedir. Oluşturulan gruplarda hangi elemanların bulunduğu Tablo 25'de verilmektedir. Gruplandırma yapılırken genellikle sistemin simetriklik özelliğinden yararlanılmaktadır.

Sistemin genetik algoritma ile her iki yükleme durumuna göre optimize edilmesi sonunda ilk durum daha elverişsiz olduğu için bu ilk durum dikkate alınarak 189. iterasyonda %80 yakınsama ile  $W=1.812$  kN'luk bir toplam birey ağırlığı elde edilmiştir. Sistemdeki düğüm noktalarında meydana gelen en büyük yerdeğiştirme 0.635 cm olup bu değer müsaade edilen değerden büyük değildir. Elde edilen tasarım sonuçları, Tablo 26'da belirtilen enkesit alanlarını kullanmak süretiyle, literatürde yapılan bir çalışmayla karşılaştırmalı olarak Tablo 27'de sunulmaktadır.



Şekil 16. Yetmişiki çubuklu uzay kafes sistem

Tablo 24. Yetmişiki çubuklu sistem için yükleme durumları

Birinci yükleme durumu			
Yüklü nokta	Yük değeri (kN)		
	$F_x$	$F_y$	$F_z$
17	22,24	22,24	-22,24
İkinci yükleme durumu			
Yüklü nokta	Yük değeri (kN)		
	$F_x$	$F_y$	$F_z$
17	0,0	0,0	-22,24
18	0,0	0,0	-22,24
19	0,0	0,0	-22,24
20	0,0	0,0	-22,24

Tablo 25. Yetmişiki çubuklu sistem için gruplandırma sonuçları

Grup no	Elemanlar
1	1, 2, 3, 4
2	5, 6, 7, 8, 9, 10, 11, 12
3	13, 14, 15, 16
4	17, 18
5	19, 20, 21, 22
6	23, 24, 25, 26, 27, 28, 29, 30
7	31, 32, 33, 34
8	35, 36
9	37, 38, 39, 40
10	41, 42, 43, 44, 45, 46, 47, 48
11	49, 50, 51, 52
12	53, 54
13	55, 56, 57, 58
14	59, 60, 61, 62, 63, 64, 65, 66
15	67, 68, 69, 70
16	71, 72



Tablo 26. Yetmişiki çubuklu sistemin tasarımında kullanılan enkesit alanları

Enkesit sıra no	Alan (cm <sup>2</sup> )	Enkesit sıra no	Alan (cm <sup>2</sup> )
1	1,12	17	5,86
2	1,42	18	6,31
3	1,45	19	6,56
4	1,74	20	6,91
5	1,85	21	8,24
6	2,26	22	8,70
7	2,67	23	9,03
8	2,78	24	9,40
9	3,08	25	10,10
10	3,28	26	11,00
11	3,79	27	11,50
12	3,87	28	11,90
13	4,30	29	12,30
14	4,48	30	13,20
15	4,80	31	14,10
16	5,69	32	14,30

Tablo 27. Yetmişiki çubuklu sistemin tasarım sonuçlarının karşılaştırılması

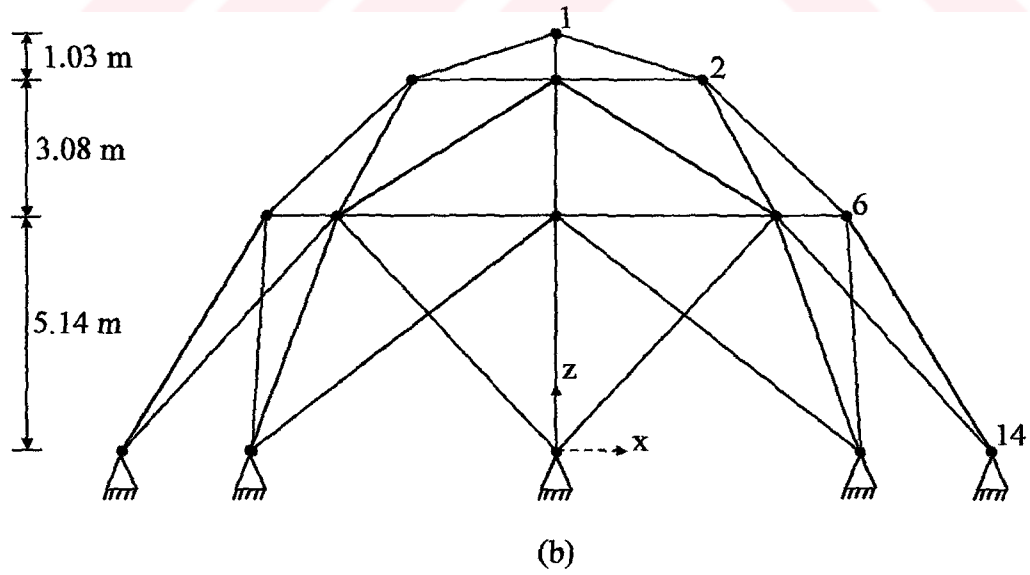
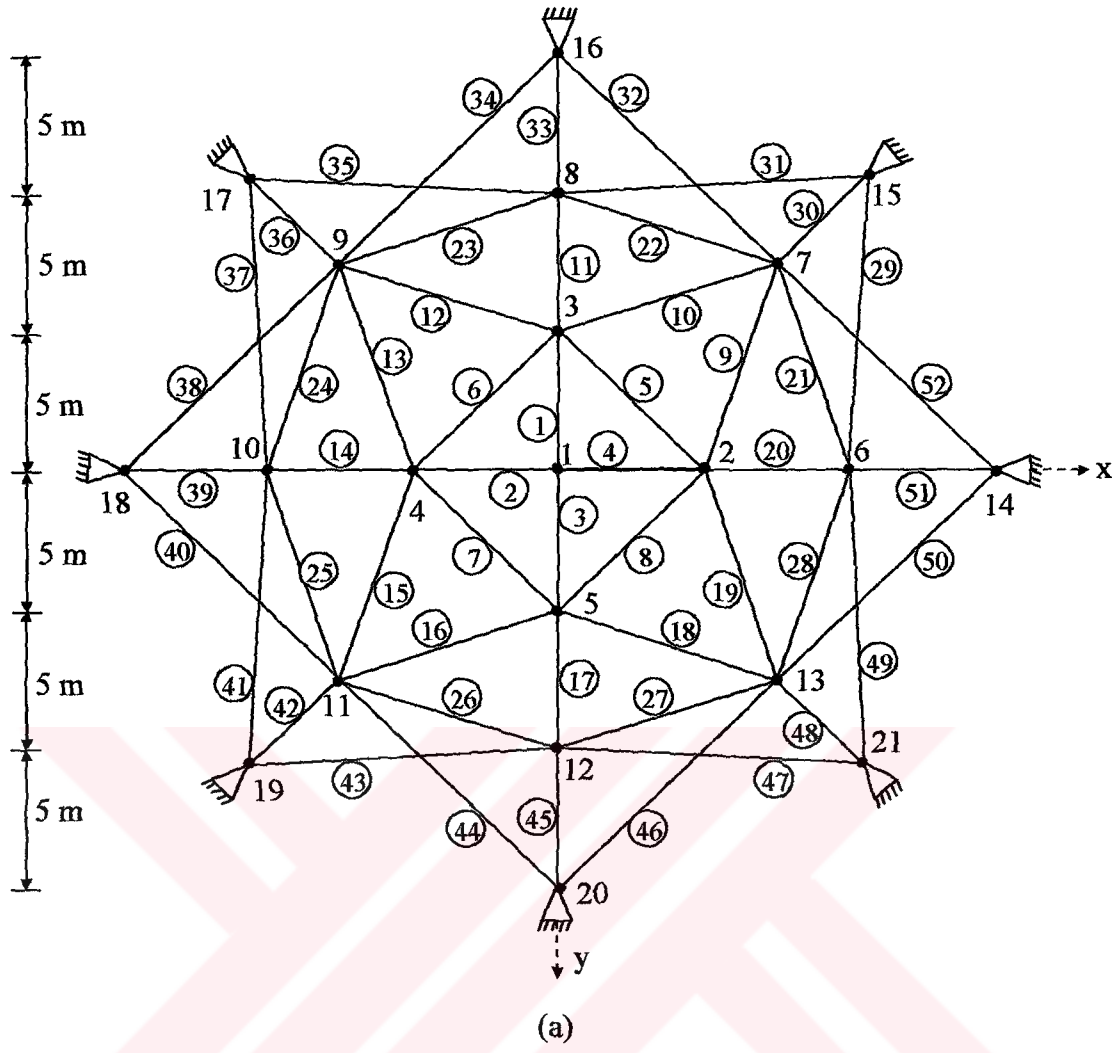
Grup no	Elemanlar	Enkesit alanları (cm <sup>2</sup> )	
		Bu çalışma	Adeli vd., [31]
1	1-4	13,20	13,07
2	5-12	3,08	3,44
3	13-16	1,12	0,65
4	17-18	1,12	0,65
5	19-22	9,40	7,46
6	23-30	3,28	3,67
7	31-34	1,12	0,65
8	35-36	1,85	0,65
9	37-40	2,78	3,31
10	41-48	3,28	3,09
11	49-52	1,42	0,65
12	53-54	1,42	0,65
13	55-58	1,12	1,02
14	59-66	3,79	3,55
15	67-70	2,78	2,23
16	71-72	2,78	3,22
Toplam ağırlık (kN)		1,812	1,688

### 2.5.5. Elliiki ubuklu Uzay Kafes Sistem

Tasarıma konu olan Elliiki ubuklu sistem Őekil 17’de verilmektedir. Bu sistem TS-648’e gre burkulmalı olarak tasarlanmıŐtır. Bu sistemin zm iin kullanılan enkesit alanları TS-648’deki eŐit kollu L profillerinden 42 adet seilmiŐtir. Bu alanlar Tablo 28’de verilmektedir. Kullanılan malzeme F37 eliĐi olup elastisite modl  $E=210000$  Mpa, birim hacim aĐırlıĐı  $\rho=0,0785$  N/cm<sup>3</sup> ve emniyet gerilmelerinin hesabı iin  $\sigma_a=235$  Mpa olarak dikkate alınmıŐtır.

Sistem 6, 7, 8, 9, 10, 11, 12 ve 13 nolu dĐm noktalarından negatif z ynnde  $F=150$  kN’luk kuvvetlerle basınca alıŐtırılıp dĐm noktalarında en fazla 10 mm’lik bir yerdeĐiŐtirmeye izin verilmektedir. Grup sayısı 8 olan bu sistemin zm iin baŐlangı neslinde 50 adet birey oluŐturulmuŐtur. Tablo 29’da gruplarda mevcut olan elemanlar gsterilmektedir.

Sistem 14, 15, 16, 17, 18, 19, 20, 21 nolu noktalardan sabit mesnetlidir. Genetik algoritmada ceza katsayısı 5,0 alınıp 110. iterasyon sonunda %80 yakınsama ile  $W=140,161$  kN aĐırlıĐında bir sistem tasarımı elde edilmiŐtir. Tasarım sonuları Tablo 30’da gsterilmektedir.



Şekil 17. Elliiki çubuklu uzay kafes sistemin (a) üstten, (b) yandan görünüşü

Tablo 28. Elliiki çubuklu sistemin tasarımında kullanılan tasarım değişkenleri

Enkesit sıra no	Profil türü	Enkesit alanı (cm <sup>2</sup> )	Atalet yarıçapı (cm)
1	L 50.50.5	4,80	1,51
2	L 50.50.6	5,69	1,50
3	L 45.45.7	5,86	1,33
4	L 55.55.6	6,31	1,66
5	L 50.50.7	6,56	1,49
6	L 60.60.6	6,91	1,82
7	L 55.55.8	8,23	1,64
8	L 50.50.9	8,24	1,47
9	L 65.65.7	8,70	1,96
10	L 60.60.8	9,03	1,80
11	L 70.70.7	9,40	2,12
12	L 55.55.10	10,10	1,62
13	L 65.65.9	11,00	1,94
14	L 60.60.10	11,10	1,78
15	L 75.75.8	11,50	2,26
16	L 70.70.9	11,90	2,10
17	L 80.80.8	12,30	2,42
18	L 65.65.11	13,20	1,91
19	L 75.75.10	14,10	2,25
20	L 70.70.11	14,30	2,08
21	L 80.80.10	15,10	2,41
22	L 90.90.9	15,50	2,74
23	L 75.75.12	16,70	2,22
24	L 80.80.12	17,90	2,39
25	L 90.90.11	18,70	2,72
26	L 100.100.10	19,20	3,04
27	L 80.80.14	20,60	2,36
28	L 90.90.13	21,80	2,69
29	L 100.100.12	22,70	3,02
30	L 100.100.14	26,20	3,00
31	L 100.100.16	29,60	2,97
32	L 120.120.15	33,90	3,63
33	L 130.130.14	34,70	3,94
34	L 130.130.16	39,90	3,92
35	L 140.140.15	40,00	4,25
36	L 150.150.16	45,70	4,56
37	L 150.150.18	51,10	4,54
38	L 160.160.19	57,50	4,84
39	L 180.180.18	61,90	5,49
40	L 180.180.20	68,40	5,47
41	L 200.200.18	69,10	6,13
42	L 200.200.20	76,40	6,11

Tablo 29. Elliiki çubuklu sistemin gruplandırma sonuçları

Grup no	Elemanlar
1	1, 2, 3, 4
2	5, 6, 7, 8
3	11, 14, 17, 20
4	9, 10, 12, 13, 15, 16, 18, 19
5	21, 22, 23, 24, 25, 26, 27, 28
6	30, 33, 36, 39, 42, 45, 48, 51
7	29, 31, 35, 37, 41, 43, 47, 49
8	32, 34, 38, 40, 44, 46, 50, 52

Tablo 30. Elliiki çubuklu sistemin tasarım sonuçları

Grup no	Elemanlar	Enkesit alanı (cm <sup>2</sup> )
1	1- 4	6,56
2	5- 8	4,80
3	11- 20	5,86
4	9- 19	6,91
5	21- 28	40,0
6	30 - 51	39,3
7	29- 49	68,4
8	32- 52	61,9
Toplam ağırlık (kN)		140,161

### 3. SONUÇLAR VE ÖNERİLER

Düzlem kafes ve uzay kafes sistemlerinin, genetik algoritmada değer kodlaması kullanarak, minimum ağırlıklı olarak boyutlandırıldığı bu çalışmada 2 adet düzlem kafes ve 3 adet uzay kafes sistem örnekleri ele alınmıştır. Genetik algoritma programı Fortran programlama dilinde yazılıp kişisel bilgisayarda koşturulmuştur. Elde edilen sonuçlar ve öneriler aşağıda maddeler halinde verilmiştir.

- Genetik algoritmalar ayırık tasarım değişkenleriyle çalıştığı için bulunan tasarım sonuçları değiştirilmeden kullanılabilir.
- Genetik algoritmalar bir problemi çözmek için çok fazla matematiksel ifadelerin oluşturulmasını gerektirmezler.
- Genetik algoritmalar optimum çözümü her zaman bulacağını garanti edememesine rağmen diğer yöntemlere göre daha kısa zamanda pratik sonuçlar üretmektedirler.
- Bu çalışmada kullanılan değer kodlaması yöntemiyle, ikilik kodlama yönteminde bulunan tasarım sonuçlarından daha az ağırlıklı tasarımlar elde edilebilmiştir. Bu durum değer kodlamasının bir avantajı olarak düşünülebilir.
- Değer kodlamasında bir bireyin kodlanması daha kısa uzunluklu dizilerden oluştuğu için hem zaman açısından hem de işlem açısından bilgisayarda kolaylıklar sağlamaktadır.
- Değer kodlamasında bir bireyin dizi uzunluğu eleman sayısına yada grup sayısına eşittir. Oysa ikilik kodlamada dizi uzunluğu tasarım değişkenlerine bağlı olarak  $2^n$ 'nin kuvvetlerinden birine eşit olmaktadır. Böylece tasarım değişken sayısının çok olduğu durumlarda ikilik kodlamada bir bireyin dizi uzunluğu artarken aynı birey için dizi uzunluğu değer kodlaması yönteminde sabit kalmaktadır.
- Çaprazlama operatöründe önce tek, sonra iki noktalı çaprazlama yapılarak iki noktalı çaprazlamanın daha etkin olduğu görülerek bu çalışmada iki noktalı çaprazlama yapılmıştır. Literatürde de önerildiği gibi bu çalışmada mutasyon oranı düşük alınmıştır. Bilindiği üzere mutasyon operatörü topluma yeni bireyler kazandırdığı için bu oran ve çaprazlama noktası sayısı artırılarak bir çalışma yapılabilir.

- Genetik algoritmada başlangıç neslinin rasgele oluşturulması nedeniyle iterasyon sayısı artabileceği için, başlangıç neslini belirli kriterlere bağlı olarak seçebilen bir alt program kullanılması uygun olabilmektedir.
- Genetik algoritmanın daha etkin çalışmasını sağlamak amacıyla literatürde bahsedilen baskınlık, göç ve paylaşım gibi diğer genetik operatörleri de kullanmakta fayda bulunmaktadır.



#### 4. KAYNAKLAR

1. Chen, D., Least Weight of 2-D and 3-D Geometrically Nonlinear Framed Structures Using a Genetic Algorithm, Ph. D. Dissertation, The University of Memphis, 1997.
2. Rasheed, K. M., A Genetic Algorithm for Continuous Design Optimization, Ph. D. Dissertation, The University of New Jersey, 1998.
3. Chen, S. Y., Using Genetic Algorithm for the Optimal Design of Structural Systems, Ph. D. Dissertation, Arizona State University, 1997.
4. Aydın, Z., Uzak Kafes Sistemlerin Genetik Algoritma ile Optimum Tasarımı, Yüksek Lisans Tezi, K.T.Ü., Fen Bilimleri Enstitüsü, Trabzon, 2000.
5. Morley, G. D. ve Grover, W. D., Tabu Search optimization of Optical Ring Transport Networks, Proceeding IEEE Globecom 2001, San Antonio, Texas, Nov., 2001, 25-29.
6. <http://www.cs.brown.edu/people/pvh/trwh.pdf>, A Simple Tabu Search for Warehouse Location, 21 Ocak 2003.
7. <http://www.cs.sandia.gov/opt/survey/ts.html>, Tabu Search, 17 Ocak 2003.
8. <http://emlab.berkeley.edu/Software/abstracts/goffe895.html>, Simulated Annealing - Global Optimization Method That Distinguishes Between Different Local Optima, 13 Ağustos 2002.
9. [http://www.isletme.istanbul.edu.tr/akpinar/content/Yapay\\_sinir\\_aglar.pdf](http://www.isletme.istanbul.edu.tr/akpinar/content/Yapay_sinir_aglar.pdf), 05 Ağustos 2002.
10. Bungay, D. S., Optimization of Transition State Structures Using Genetic Algorithms, M. Sc. Thesis, Memorial University of Newfoundland, 2000.
11. [http://www.mmo.org.tr/muhendismakina/arsiv/2001/ekim/Genetik\\_Algoritma.html](http://www.mmo.org.tr/muhendismakina/arsiv/2001/ekim/Genetik_Algoritma.html), 16 Ocak 2003.
12. Goldberg, D. E., Genetic Algorithms in Search, Optimization, and Machine Learning, 20th printing, Addison-Wesley Publishing Company Inc., New York, 1999.
13. <http://www.cs.felk.cvut.cz/~xobitkolya/intro.html>, Genetic Algorithms, 16 Ocak 2003.
14. Cao, G., Optimized Design of Framed Structures Using a Genetic Algorithm, Ph. D. Dissertation, The University of Memphis, 1996.
15. Saka, M. P., Daloğlu, A. ve Malhas, F., Optimum Spacing Design of Grillage Systems Using Genetic Algorithm, Civil-Comp., 1997, 137-148.



16. Dalođlu, A. ve Armutçu, M., Genetik Algoritma ile Düzlem Çelik Çerçeveselerin Optimum Tasarımı, İMO Teknik Dergi, 1998, 1601-1615.
17. Dalođlu, A. ve Aydın, Z., Kafes Sistemlerin Uygulamaya Yönelik Optimum Tasarımı, Mühendislik Bilimleri Dergisi, 5 (1), 1999, 948-954.
18. Park, Y., Genetic Algorithm Approach for Bandwidth Allocation in ATM Networks, Ph. D. Dissertation, Polytechnic University, 1999.
19. Joshi, B. D., Dynamic Skale Genetic Algorithm: An Enhanced Genetic Search for Discrete Optimization, Ph. D. Dissertation, Old Dominion University, 1996.
20. Rajeev, S. ve Krishnamoorthy, C. S., Discrete Optimization of Structures Using Genetic Algorithms, Journal of Structural Engineering, 118 (5), 1992, 1233-1251.
21. Dalođlu, A. ve Armutçu, M., Kafes Sistemlerin Genetik Algoritma ile TS 648'e Uygun Olarak Optimum Tasarımı, İnşaat Mühendisliğinde Gelişmeler, III. Teknik Kongre, ODTÜ, Ankara, 1997, 459-467.
22. Hawat, R. N., Genetic Algorithm Approach to Curve and Surface Intersection, Ph. D. Dissertation, University of South Florida, 1998.
23. <http://www.maru.cs.ritsumei.ac.jp/~taku/study/RINKO.TXT>, Handbook of Genetic Algorithms, 16 Ocak 2003.
24. Sudarshan, R., Genetic Algorithms and Application to the Optimization of Space Trusses, A Project Report, Indian Institute of Technology, Madras, 2000.
25. Brierley, P. D., Some Practical Applications of Neural Networks in the Electricity Industry, EngD Thesis, Cranfield University, 1998.
26. İnan, M., Cisimlerin Mukavemeti, 7. Baskı, İ.T.Ü. Vakfı Yayınları, İstanbul, 1996.
27. Manuel of Steel Construction – Allowable Steel Design, 1989, 9<sup>th</sup> Ed., Am. Inst. of Steel Constr., Chicago, III.
28. Arun, C. S., A Modular Genetic Algorithm and Its Implementation in a Distrubuted Environment for Structural Optimization, M. Sc. Thesis, The University of Texas, 1996.
29. Rajeev, S. ve Krishnamoorthy, C. S., Genetic Algorithms-Based Methodologies for Design Optimization of Trusses, Journal of Structural Engineering, 123 (3), 1997, 350-358.
30. Groenwold, A. A., Stander, N. ve Snyman, J. A., A Regional Genetic Algorithm for the Discrete Optimal Design of Truss Structures, International Journal for Numerical Methods in Engineering, 44, 1999, 749-766.

31. Adeli, H. ve Kamal, O., Efficient Optimization of Space Trusses, Computers & Structures, 24 (3), 1986, 501-511.
32. Fleury, C. ve Geradin, M., Optimality Criteria and Mathematical Programming in Structural Weight Optimization, Computers & Structures, 8, 1978, 7-17.
33. Saka, M. P. ve Ulker, M., Optimum Design of Geometrically Nonlinear Space Trusses, Computers & Structures, 41 (6), 1991, 1387-1396.
34. Lipson, S. L. ve Gwin, L. B., The Complex Method Applied to Optimal Truss Configuration, Computers & Structures, 7, 1997, 461-468.
35. Sarma, K. C., Fuzzy Discrete Multicriteria Cost Optimization of Steel Structures Using Genetic Algorithm, Ph. D. Dissertation, The Ohio State University, 2001.
36. TS 648, Çelik Yapıların Hesap ve Yapım Kuralları, T.S.E., Ankara, I. Baskı, Nisan 1982.
37. Ayvaz, Y. ve Aydın, Z., "Optimum Design of Trusses Using a Genetic Algorithm", Proceedings of the Second International Conference on Engineering Using Metaphors from Nature, Leuven, Belgium, pp:159-168, 6-8 September 2000 .

## ÖZGEÇMİŞ

Tayfun DEDE 1979 yılında Trabzon'da doğdu. İlk ve orta öğrenimini Gölçayır Köyü İlköğretim okulu'nda, lise öğrenimini Trabzon Affan Kitapcıoğlu Lisesi'nde tamamladı. 1996-1997 öğretim yılında Karadeniz Teknik Üniversitesi Mühendislik-Mimarlık Fakültesi İnşaat Mühendisliği Bölümü'nde lisans öğrenimine başladı. Lisans öğrenimi süresinde Fakülte Dekanlığı'ndan onur ve yüksek onur belgeleri aldı. İnşaat Mühendisliği Bölümü'nden 30 Haziran 2000 tarihinde bölüm üçüncüsü olarak mezun olup aynı yılda bu bölümde yüksek lisans eğitimine başladı. Yüksek lisans programında bir yıl İngilizce eğitimi aldıktan sonra, 21 Aralık 2002 tarihinde Karadeniz Teknik Üniversitesi Fen Bilimleri Enstitüsü'nde araştırma görevlisi olarak atandı. İngilizce bilen Tayfun DEDE halen bu görevini sürdürmektedir.

