

**KARADENİZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

İNŞAAT MÜHENDİSLİĞİ ANABİLİM DALI

**GENETİK ALGORİTMA İLE ÇELİK ÇERÇEVELERİN
OPTİMUM BOYUTLANDIRILMASI**

139198

İnş. Müh. Serkan BEKİROĞLU

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

138138

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. YÜKSEKÖĞRETİM KURULU
DOKÜMANTASYON MERKEZİ
DOKÜMANIZASYON MERKEZİ**

ÖNSÖZ

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

Optimizasyon yöntemleri arasında yer alan genetik algorithmada farklı kodlama çeşitleri ile çok noktalı çaprazlama ve akıllı mutasyon kullanılarak çelik çerçeve sistemlerin minimum ağırlıklı olarak boyutlandırıldığı bu çalışma süresince sürekli destek olan hocam Sayın Prof. Dr. Yusuf AYVAZ'a teşekkürlerimi sunarım.

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

Çalışmam boyunca yardımlaştığım Arş. Gör. Tayfun DEDE ve Arş. Gör. Vedat TOĞAN'a; hocalarıma ve tüm diğer arkadaşlarıma teşekkür ederim.

Hayatım boyunca hep yanımda olan ve desteklerini esirgemeyen aileme teşekkürümü bir borç bilirim.

Serkan BEKİROĞLU
Trabzon, 2003

İÇİNDEKİLER

Sayfa No

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. Yapısal Tasarımın Karakteristikleri	2
1.3. Tasarım Optimizasyonu ve Karşılaşılan Problemler.....	3
1.4. Uygulanan Optimizasyon Teknikleri	4
1.4.1. Matematik Programlama Yöntemi	4
1.4.2. Optimumluk Kriteri Yöntemi.....	6
1.4.3. Yapay Zeka ve Evrimsel Optimizasyon Teknikleri	6
1.4.3.1. Yapay Sinir Ağları	7
1.4.3.2. Bulanık Mantık.....	9
1.4.3.3. Uzman Sistemler	11
1.4.3.4. Ardışık Optimizasyon (Hill Climbing, HC) Yöntemi.....	12
1.4.3.5. Benzetilmiş Tavlama (Simulated Annealing)	13
1.4.3.6. Genetik Algoritma.....	14
1.5. Çalışmanın Amaç ve Kapsamı	16
2. YAPILAN ÇALIŞMALAR, BULGULAR VE İRDELEME	17
2.1. Çelik Çerçevelerin AISC-ASD ve TS 648 Göre Tasarımı.....	17
2.1.1. Emniyet Gerilmeleri.....	18
2.1.1.1. Eksenel Stabilite (Burkulma)	18
2.1.1.2. Yanal Burkulma	23
2.1.1.3. Kritik Burkulma Gerilmesi	25
2.1.2. Eksenel Basınç ve Eğilmeye Çalışan Yapı Elemanlarında Gerilme Denetimi	25

2.1.3.	Eksenel Çekme ve Eğilmeye Çalışan Yapı Elemanlarında Gerilme Denetimi	26
2.1.4.	Kayma Gerilmesi Denetimi.....	27
2.1.5.	Makaslama Denetimi.....	27
2.2.	Genetik Algoritma ve Optimizasyon.....	27
2.2.1.	Doğa ile Genetik Algoritmanın Benzerlikleri	29
2.2.2.	Doğal Seçilim ve Genetik Algoritma	29
2.2.3.	Genetik Algoritmada Yapılan Temel Kabüller	30
2.3.	Genetik Algoritma ile Yapıların Optimum Tasarımı	31
2.3.1.	Genetik Algoritmada Çözümlerin Kodlanması.....	31
2.3.2.	Başlangıç Stratejisi.....	34
2.4.	Yapısal Optimizasyon Problemi.....	36
2.4.1.	Sınırlayıcılar	36
2.4.2.	İhlaller ve Ceza Fonksiyonu.....	37
2.4.3.	Amaç Fonksiyonu	40
2.4.4.	Uygunluk Değerinin Hesaplanması	41
2.4.5.	Genetik Operatörlerin Uygulanması	42
2.4.5.1.	Seçim Stratejisi ve Kopyalama Operatörü	43
2.4.5.2.	Çaprazlama Operatörü.....	45
2.4.5.3.	Mutasyon Operatörü.....	48
2.4.6.	Yeni Kuşağın Oluşması ve Döngünün Durdurulması.....	52
2.5.	Genetik Algoritmanın Avantaj ve Dezavantajları.....	52
2.6.	Çerçeve Sistemlerin Tasarım Örnekleri	53
2.6.1.	Altı Katlı İki Açıklıklı Çerçeve.....	53
2.6.2.	Diyagonal Kafes Elemanlı Altı Katlı İki Açıklıklı Çerçeve	58
2.6.3.	Bir Katlı Bir Açıklıklı Çerçeve	64
3.	SONUÇLAR VE ÖNERİLER	66
4.	KAYNAKLAR.....	69
	ÖZGEÇMİŞ	72

ÖZET

Genetik algoritma (GA), bir sistem veya problemin çözümü olabilecek mümkün alternatiflerden istenilen kriterler doğrultusunda ideal sonuç arayan bir optimizasyon yöntemidir. Yöntem gereğince, problemin olası çözüm kümeleri hayatta kalmak için mücadele eden canlı topluluklarına benzetilmektedir. Yaşam süresince ortaya çıkan engeller, toplum içindeki yaşam gücü fazla olan bireylerin toplum içinde sivrilmesini, az olanların ise toplumdaki elimine edilmesine neden olmaktadır. Toplulukların tümü, bir problemde mümkün olan çözüm kümelerini temsil edecek olursa, GA, bu çözüm kümelerinin herhangi birinden başlayarak toplumlar arasında yaşam gücü en yüksek olan bireyi bulmaya çalışmaktadır.

Bu çalışmada, çok noktalı çaprazlama ve akıllı mutasyon operatörleri ile farklı kodlama çeşitleri kullanılarak GA ile çelik çerçeve sistemlerin minimum ağırlıklı tasarımı yapılmaktadır. Bu amaçla gerilme, yerdeğiştirme, stabilite, sehim ve geometrik şartları sınırlayıcı olarak içeren, analizinde Matris Deplasman Yöntemi kullanılan ve FORTRAN dilinde kodlanmış bir program ile çalışılmıştır.

Bu çalışma içinde dört bölüm yer almaktadır. Birinci bölümde optimizasyon teknikleri ve çalışmanın amacı hakkında bilgiler verilmektedir. İkinci bölümde literatürden alınan birkaç çelik çerçeve sistem örneği genetik algoritma ile optimize edilmekte, sonuçlar literatürdeki sonuçlarla karşılaştırılmakta ve irdelenmektedir. Üçüncü bölümde ise elde edilen sonuçlar sunulmakta ve bazı öneriler verilmektedir. Bu bölümü kaynaklar listesi izlemektedir.

Sonuç olarak, çok noktalı çaprazlama ve akıllı mutasyon operatörleri ile farklı kodlama çeşitlerinin GA'nın performansını arttırdığı görülebilmektedir.

Anahtar Kelimeler: Minimum Ağırlıklı Boyutlandırma, Genetik Algoritma, Kodlama Çeşitleri, Çok Noktalı Çaprazlama, Akıllı Mutasyon, Çelik Çerçeve Sistemler

SUMMARY

Optimum Design of Steel Frame with Genetic Algorithm

Genetic algorithm (GA) is an optimization method that obtains an ideal result among available choices that is solution of problem or system by ensuring desired criteria. According to the method, a set of possible solutions for problem is simulated real populations struggling for survival in nature. Difficulties occurring during life provide the survival of the fittest while the others are eliminated from these populations. If all of the populations represent a set of available solutions for problem, GA tries to find the best individual among these populations by starting with any of them.

In this study, the minimum weight design of steel framework is done by using GA with different encoding, intelligent mutation and flexible crossover operators. For this purpose, a program is coded in FORTRAN. In the program, stress, stability, slope and geometrical conditions are included as constraint. In the analysis of the frame systems, Matrix Analysis of Structures is used.

This study consists of four chapters. In the first chapter, some information about optimization technique and the purpose of the study are given. In the second chapter, some steel framework examples taken from literature are optimized with GA, and the design results obtained in this study are compared with the design results taken from literature. In the third chapter, the conclusions drawn from this study are presented and some recommendations are given. In the fourth chapter, the references are presented.

In conclusion, it appears that varieties of different encoding with intelligent mutation and flexible crossover operators increase performance of GA.

Keywords: Minimum Weight Design, Genetic Algorithm, Variety of Different Encoding, Flexible Crossover, Intelligent Mutation, Steel Framework

ŞEKİLLER DİZİNİ

Sayfa No

Şekil 1.	(a) Gerçek ve (b) lineerleştirilmiş optimizasyon problemi	5
Şekil 2.	Yanal ötelemesi (a) önlenmemiş (b) önlenmiş sistemler için burkulma boyu katsayıları	21
Şekil 3.	Yanal ötelenmesi önlenmemiş sistemler için gerçek ve hesaplanan burkulma boyu katsayısı	22
Şekil 4.	Yanal ötelenmesi önlenmiş sistemler için gerçek ve hesaplanan burkulma boyu katsayısı	22
Şekil 5.	Uç moment oranlarının, C_b , işaretleri.....	24
Şekil 6.	Birey topluluğu (popülasyon).....	30
Şekil 7.	Sekiz elemanlı düzlem çelik çerçeve.....	32
Şekil 8.	Altı katlı iki açıklıklı çerçeve	54
Şekil 9.	Altı katlı iki açıklıklı çerçevenin 5. tasarıma göre kolonlardaki burkulma boyu katsayısı	56
Şekil 10.	Altı katlı iki açıklıklı çerçevenin 5. tasarıma göre kirişlerdeki burkulma boyu katsayısı	56
Şekil 11.	Diyagonal kafes elemanlı altı katlı iki açıklıklı çerçeve	59
Şekil 12.	Optimum tasarıma sahip 20. bireyin iterasyon sayısına bağlı olarak cezalandırılmış amaç fonksiyonu değişimi	61
Şekil 13.	Ortalama cezalandırılmış amaç fonksiyonunun iterasyon sayısına bağlı olarak değişimi	62
Şekil 14.	Optimum tasarıma sahip 20. bireyin iterasyon sayısına bağlı olarak ağırlık değişimi.....	63
Şekil 15.	Optimum tasarıma sahip 20. bireyin mutasyona uğrayan tasarım değişkeni sayısının iterasyon sayısına bağlı olarak değişimi	63
Şekil 16.	Bir katlı bir açıklıklı çerçeve.....	64

TABLULAR DİZİNİ

Sayfa No

Tablo 1.	Genetik algoritma ile doğanın karşılaştırılması.....	29
Tablo 2.	Bireyin fenotip ve genotip gösterimi.....	31
Tablo 3.	Enkesit alan kodlamalarının veri tabanındaki sayı ve değer karşılığı.....	33
Tablo 4.	Birey ve tasarım değişkenleri arasındaki ilişki.....	33
Tablo 5.	Birey uygunluklarının bulunuşu.....	44
Tablo 6.	Çaprazlama yapılacak tasarım değişkeni sayısının belirlenmesi	48
Tablo 7.	Mutasyona uğrayacak tasarım değişkeni sayısının belirlenmesi.....	50
Tablo 8.	Altı katlı iki açıklıklı çerçevenin eleman grup numaraları.....	55
Tablo 9.	Altı katlı iki açıklıklı çerçevenin tasarım sonuçları.....	57
Tablo 10.	Diyagonal kafes elemanlı altı katlı iki açıklıklı çerçevenin eleman grup numaraları	60
Tablo 11.	Diyagonal kafes elemanlı altı katlı iki açıklıklı çerçevenin tasarım sonuçları.....	60
Tablo 12.	Bir katlı bir açıklıklı çerçevenin tasarım sonuçları	65

SEMBOLLER DİZİNİ

A_b	: Basınç başlığının enkesit alanı
A_i	: i. çubuğun alanı
A_k	: k. gruba ait eleman alanı
C	: Toplam ihlal değeri
C_b	: Uç moment oranlarına göre hesaplanan bir katsayı
c_i	: Her bir sınırlayıcının ihlal edilme değeri
C_{mx}	: x doğrultusundaki mesnetlenmeyi ifade eden katsayı
C_{my}	: y doğrultusundaki mesnetlenmeyi ifade eden katsayı
d	: İki başlık dış yüzeyleri arasındaki mesafe
δ_0	: Eğilme yükünden meydana gelebilecek maksimum yerdeğiştirme
δ_j	: j. düğüm noktasında hesaplanan yer değiştirme miktarı
δ_k	: k. düğüm noktasında hesaplanan sehim miktarı
δ_{uj}	: j. düğüm noktası için belirlenen yer değiştirme miktarı üst sınırı
δ_{uk}	: k. düğüm noktasında izin verilen sehim miktarı
E	: Elastisite modülü
EI	: Çubuk rijitliği
f	: Birey uygunluğu
f'	: Çaprazlanacak ebeveynlerden en büyük uygunluğu olanın uygunluğu
\bar{f}	: Ortalama uygunluk
\bar{F}	: Uyum derecesi ortalaması
$f(x)$: Uygunluk fonksiyonu
$F(x)$: Amaç fonksiyonu
$\Phi(x)$: Cezalandırılmış amaç fonksiyonu
$\Phi(x)_{max}$: Popülasyon içinde en büyük değerdeki cezalandırılmış amaç fonksiyonu değeri
$\Phi(x)_{min}$: Popülasyon içinde en küçük değerdeki cezalandırılmış amaç fonksiyonu değeri
$F_{c,i}$: Her bir bireyin uyum faktörü
F_i	: Uyum derecesi
f_{max}	: Maksimum uygunluk
$G_1(x)$: Sınır fonksiyonu

$G_2(x)$: Sınır fonksiyonu
G_A	: Çubuğun birinci ucunun rijitlik oranı
G_B	: Çubuğun ikinci ucunun rijitlik oranı
g_f	: f. giriş için makaslama gerilmesi ihlali
g_i	: i. basınç elemanı için gerilme ihlali
g_j	: j. düğüm noktasındaki ihlal değeri
g_p	: p. çekme elemanı için gerilme ihlali
g_t	: t. basınç elemanı için gerilme ihlali
g_u	: u. giriş için kayma gerilmesi ihlali
I_B	: Burkulma düzlemindeki kolon atalet momenti
i_{by}	: Basınç başlığı ve gövdenin basınç bölgesinin 1/3'ünün gövde eksenine göre atalet yarıçapı
I_C	: Burkulma düzlemindeki kolon atalet momenti
i_x	: Çubuğun x-x eksenine göre atalet yarıçapı
i_y	: Çubuğun y-y eksenine göre atalet yarıçapı
k	: Burkulma boyu katsayısı
$K_{1,.., K_5}$: 1. eşin karakteri
$M_{1,.., M_5}$: 2. eşin karakteri
\bar{K}, \bar{M}	: Ebeveyn karakterleri
λ	: Narinlik oranı
L_B	: Girişin akstan aksa uzunluğu
L_C	: Kolonun akstan aksa uzunluğu
L_i	: i. çubuğun boyu
λ_p	: Plastik narinlik sınırı
λ_x	: x-x eksenine göre narinlik oranı
λ_y	: y-y eksenine göre narinlik oranı
M_0	: Açıklıktaki maksimum moment
M_1	: Yapı elemanının uç momentlerinin küçüğü
M_2	: Yapı elemanının uç momentlerinin büyüğü
N	: Emniyet katsayısı
ng	: Problemdaki toplam grup sayısı
nm	: Sistemdeki çubuk sayısı
P	: Ceza katsayısı

p	: Yer deęiřtirmesi sınırlandırılmıř nokta sayısı
p_c	: Birleřim noktalarının özellikleri
P_c	: aprazlama oranı
PC	: Ceza fonksiyonu
p_m	: Malzeme özellikleri
P_m	: Mutasyon oranı
p_s	: Sistemin yapısal karakteristikleri
r	: Sehimi sınırlandırılmıř cubuk sayısı
ρ_i	: i. ubuęun yoęunluęu
s	: ubuęun basın bařlıęında dnmeye ve burkulmaya karřı mesnetleri arasındaki mesafe
σ_a	: Akma gerilmesi
σ_B	: Eęilme emniyet gerilmesi
σ_{bem}	: Basın emniyet gerilmesi
σ_{Bx}	: x-x eksenine gre eęilme emniyet gerilmesi
σ_{bx}	: x-x eksenine gre eęilme gerilmesi
σ_{By}	: y-y eksenine gre eęilme emniyet gerilmesi
σ_{by}	: y-y eksenine gre eęilme gerilmesi
s_e	: Elemanın boyu
σ_e	: Kritik burkulma (Euler) gerilmesi
$\sigma_{e,i}$: i. ubuk iin hesaplanan gerilme deęeri
σ_{ex}	: x-x eksenine gre kritik burkulma gerilmesi
σ_{ey}	: y-y eksenine gre kritik burkulma gerilmesi
$\sigma_{k,u}$: u. kiriř iin kayma gerilmesi
S_{k_x}	: ubuęun x-x eksenine dik dzlemdeki burkulma boyu
S_{k_y}	: ubuęun y-y eksenine dik dzlemdeki burkulma boyu
σ_v	: Makaslama gerilmesi
x_1, x_2	: Tasarım deęiřkenleri
ψ	: Mesnetlemeye, yklemeye ve ubuęun rijitlięi ile boyuna baęlı bir katsayı
AI	: Yapay zeka
GA	: Genetik algoritma

- EY : Esas yüklemeler
EIY : Esas ve ilave yüklemeler
HC : Ardışık optimizasyon (Hill climbing) metodu
LP : Lineer programlama
NLP : Lineer olmayan programlama
OC : Optimum kriterler metodu
SA : Benzetilmiş tavlama (Simulated annealing)
YSA : Yapay sinir ağıları



1. GENEL BİLGİLER

1.1. Giriş

Yapı tasarımı, yapıya gelecek etkilere bağlı olarak bünyesinde oluşacak tepkilerin, izin verilen değerlere eşitlenmesidir. Eşitliğin kurulabilmesi etkiyi ve tepkiyi belirleyen kriterlerin açık bir şekilde ortaya konmasına bağlıdır. Bu nedenle yapı problemlerinin neler olduğu tespit edilmeye çalışılmış ve yapı elemanlarının boyutlandırılmasında etkin olan kriterler ortaya çıkarılmıştır. Ayrıca yapının maruz kalacağı dış kuvvetlerin karakteristikleri ve ne tür etkiler yaratabilecekleri belirlenmiştir. Bunların yanı sıra malzeme biliminde de büyük ilerlemeler sağlanmıştır. Taşıyıcı elemanlar önceleri taş ve topraktan oluşturulurken, artık çelik, betonarme ve kompozit elemanlardan meydana gelebilmektedir. Yapılar tasarlanırken malzeme, yapı elemanı ve dış etkilerin keşfedilen karakteristiklerindeki gelişmelere bağlı olarak, dikkate alınan unsurların sayısı artmakta ve karar verme işlemi zorlaşmaktadır. Mühendislikte problemleri basite indirgeme içgüdüsünden hareketle, karar verme mekanizmasını daha etkin kılabilmek amacıyla yapı tasarımında yeniliğe gidilmektedir. Her şeyin sınırlı miktarlarda var olduğu dünyada amacına hizmet etmekle görevli olan yapının tasarımı için, sistemin, taşıyıcı elemanlarında kullanılacak malzemenin ve yapıya etkiyecek dış etkenlerin etkin bir biçimde belirlenmesi gereklidir. Bu nedenle optimum tasarıma bir yöneliş vardır. Optimum tasarım, mevcut olabilecek çözümler arasından istenilen kriterlere en uygun olanıdır. En uygunun bulunması, elle hesabı imkansız bir döngüyle gerçekleştirilmektedir. Örneğin yapı elemanlarının boyutlandırılması, belirlenen yükler ve bunları dikkate alacak yüklemeler altında; elemanların konumundan etkilenen ve sistemin davranışını belirleyen bir önemi haizdir. Tasarımın ilk adımı olan yüklerin belirlenmesinde dahi, eleman boyutlarına ihtiyaç duyulmaktadır. Eleman sayısı fazla olan sistemler düşünüldüğünde ön boyutlandırma işlemi, karar vermesi güç bir safhadır. Çünkü verilecek ön boyutlara göre sistemlerin yükü belirlenmeli, yapısal çözümlene yapılmalı ve eleman ile sistem bazında kriterlerin sağlanıp sağlanmadıkları kontrol edilmelidir. Eğer bir yapı elemanının boyutu değişirse tüm bu işlemler tekrarlanmalıdır. Böylesine büyük bir işlem hacminin elle hesabı kolay olmadığı gibi, bilgisayar programlarıyla bile her defasında veri girişi yapılması büyük güçlükler neden olmaktadır. Ayrıca çözüm olarak sunulan boyutlandırmanın, bütün

çözümler dikkate alınmadığından, gerçekten arzu edilen kriterleri sağlayan bir en iyinin olup olmadığı bilinmemektedir. Fakat bilgisayarların gelişimi, hem zaman hem de optimum tasarımın yapılmasını mümkün kıldığından yeterli imkanları sağlamaktadır.

Optimum tasarım alanındaki araştırmalar; yapısal tasarımın vazgeçilemez bir unsur olması, tasarım sürecinin kısaltılmaya çalışılması, bilgisayar yardımıyla tasarımın cazibeli hale getirilmesi ve tasarım sürecinin ekonomi ile zaman açısından önemli oluşu gibi birçok sebepten dolayı son kırk yıl içerisinde büyük ivme kazanarak etkin bir biçimde sürdürülmektedir. Bununla birlikte, tüm yapısal tasarım problemlerine çözüm olabilecek etkili, güvenilir ve doğru sonuçlar sağlayan bir yöntem bilimi veya teknik mevcut değildir (Chen, 1997).

Geliştirilen yöntemler ve onlar sayesinde pratiklik sağlayan bilgisayar programlarının optimum tasarım sürecine katkısı çok büyük olmakla beraber en azından şu an için sınırlıdır. Bir yapı tasarımında kullanılacak sistem için çoğunlukla, eleman boyutlarındaki değişikliklerin beraberinde getirdiği farklı çözümler arasında arama yapılmaktadır. Fakat, sistemin ve/veya malzemenin de değiştirilmesinden meydana gelecek çözümler dikkate alınmamaktadır. Malzeme ve sistem çeşitliliğinden ileri gelen seçenekleri dikkate almak, kullanılan bilgisayar programlarının kapsamını aşmaktadır. Dolayısıyla bu iki faktörün seçimi mühendislik yargısı ile yapılmaktadır. Günümüz itibarıyla de geliştirilen yöntem ve programlar ancak mühendislik yargısının etkinliğini artırmaktadır.

1.2. Yapısal Tasarımın Karakteristikleri

Çelik çerçeve tasarımında bir mühendisin eleman tipine göre kullanılacak uygun enkesitleri belirleyebilmesi amacıyla dikkate alması gerektiği bazı kriterler mevcuttur. Örneğin, kirişlerin, döşeme dolayısıyla konacak ikincil kirişlerden bağlantı düzenine göre destek alıp almayacağına bağlı olarak yanal burkulma etkisinin azaltılabileceğinin tasarlanması gerekmektedir (Cao, 1996). Ayrıca yapının yanal ötelenme yapıp yapmamasına, sisteme yerleştirilecek betonarme perde, diyagonal eleman gibi destekleyicilerle veya yanal hareketi önleyen başka sınırlayıcılarla karar verilmeli ve en olumsuz durumu yaratacak yük kombinasyonları öngörülmelidir (Cao, 1996).

Yukarıda belirtilenlerin dışında, yapı sisteminin belirlenmesinin ayrı bir önemi vardır. Yapı sistemini oluşturan yapı elemanlarının konumu ve dağılımı yapının etkiyecek

yükler altında davranışını belirlemektedir. Yapının performansı elemanların boyutları ile doğrudan ilgilidir. Analize ön boyutlandırma ile başlanmaktadır. Ön boyutlandırmanın uygunluğu tasarım döngüsünün kısalmasında önemli etkenlerden biridir. Daha sonra her düğüm noktasının yerdeğiřtirmesi ve elemanlardaki enkesit etkileri bulunmakta ve izin verilen deęerlerin saęlanıp saęlanmadığı kontrol edilmektedir (Cao, 1996).

Rölatif yerdeğiřtirmeler izin verilen deęerin üstünde ise yapının stabilitesinin yetersiz olduęu kanısına varılmakta ve eleman boyutları deęiřtirilmektedir. Ayrıca her eleman için gerilme denetimleri yapılmaktadır. Yapı elemanları bu denetimleri saęlamaz ise yine ön boyutlar deęiřtirilmektedir. Tüm istenenleri saęlayan tasarımlar uygulanabilir niteliktedir. Ancak tasarımın kalitesine karar vermek için başka uygulanabilir çözümlerin de bulunması gerekmektedir. Bu boyutlandırma döngüsü her bir çözüm için tekrar icra edilmelidir. Böylelikle optimum çözüme gidilebilmektedir (Cao, 1996).

1.3. Tasarım Optimizasyonu ve Karşılaşılan Problemler

Tasarım optimizasyonu, yapı elemanlarının geometrik parametreleri ve mukavemet özelliklerinin seçimi ile ilişkili olup keyfi çözümlerin olduęu bir araştırma havuzunda yapılmaktadır. Bu araştırma havuzu tasarımcının sezgi veya özel yeteneğine göre deęil nesnel ve rasyonel şekilde oluşturulmaktadır. Sonrasında tasarım optimizasyonu boyut seçme ve sınırların kontrolü ile gerçekleřtirmektedir.

Tasarımların bilgisayar yardımı ile gerçekleştirilmesi geleneksel tasarımlara kıyasla daha güvenilir ve doęru analizlerin yapılmasına imkan saęladığı gibi hesap hacmi ve süresini de kısaltmaktadır. Ayrıca bilgisayar, tasarım süresini kısalttığından ve daha fazla tasarımın dikkate alınmasına olanak saęladığından en ideal yani optimum çözüme ulaşma şansını artırmaktadır.

Tasarımda, önceden yapılmış yapılardan edinilen tecrübeler, laboratuarlarda yapılan testler ve araştırma sonuçlarının sentezinden yararlanılmaktadır. Tasarımın ekonomik olmasının yanı sıra gereksinimleri karşılamasına (fonksiyonel olması) ve güvenilir olmasına dikkat edilmelidir. Bu dengeyi saęlamak kolay olmamaktadır.

Geleneksel tasarım yapılırken karşılaşılan bazı zorluklar,

- mühendislerin farklı tasarım yapmaları,
- boyutlandırma yapılırken kompleks yapıların yinelenen analizleri,
- mümkün olan tüm yük durumlarının dikkate alınamaması,

- yapının kullanım şeklinin ekonomik tasarımda güçlükler neden olması,
 - mevcut tasarımlardan ve analiz tekniklerinden en uygununun seçimi,
- şeklinde sıralanabilir (Mahfouz, 1999).

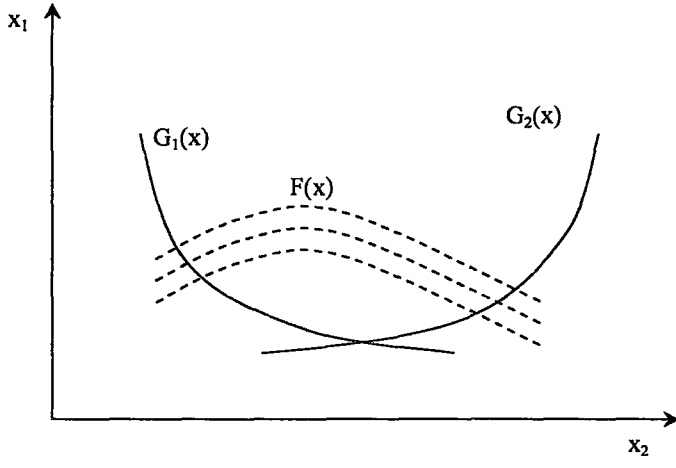
1.4. Uygulanan Optimizasyon Teknikleri

Literatürde çelik çerçevelerin optimizasyonu ile ilgili birçok kaynak mevcuttur. Bu kaynakların bazıları sürekli bazıları da ayrık çözümler ile çalışmaktadır. Ayrık optimizasyon teknikleri pratikte mevcut olan ayrık çözümleri kullandığı için son zamanlarda büyük ilgi çekmektedir (Choi vd., 1991). Genelde yapı mühendisliğinde yapılan tasarımlarda kullanılan optimizasyon teknikleri 1) matematik programlama yöntemi 2) optimumluk kriteri yöntemi, 3) yapay zeka ve evrimsel optimizasyon teknikleri şeklinde üçe ayrılmaktadır.

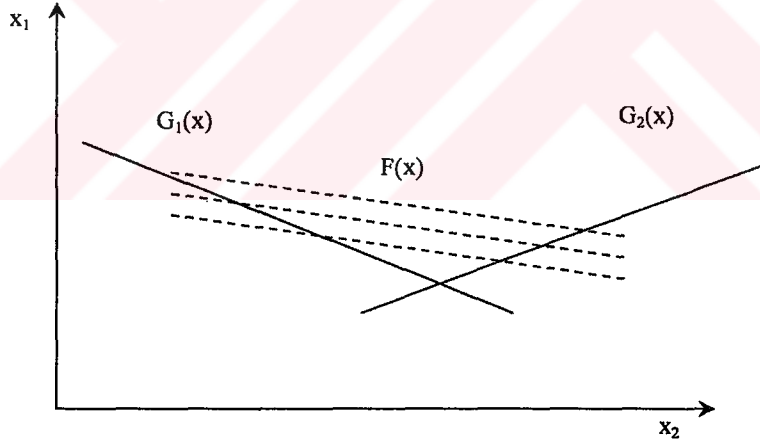
1.4.1. Matematik Programlama Yöntemi

Matematik programlama lineer programlama (LP) ve lineer olmayan programlama (NLP) olmak üzere ikiye ayrılmaktadır. LP yönteminin en büyük karakteristiği; amaç fonksiyonunun, ilgili sınırların ve tasarım değişkenlerinin lineer bir kombinasyonundan oluşmasıdır. Yapıların optimizasyonunda LP yöntemini uygulamak için amaç fonksiyonu, sınırlar ile lineerleştirilmelidir. Bu lineer ilişki, lineer olmayan yapıları optimize etmek için kullanıldığında, Şekil 1'deki gerçek ve lineerleştirilmiş optimizasyona bakılarak anlaşılacağı gibi, hata yapmak kaçınılmaz olmaktadır. Bu şekilde $F(x)$ amaç fonksiyonunu, x_1 ve x_2 tasarım değişkenlerini, $G_1(x)$ ve $G_2(x)$ de ilgili sınırları temsil etmektedir (Mahfouz, 1999).

NLP yöntemi lineer olmayan sınırlandırılmamış optimizasyon problemleri için geliştirilmiştir. Kuhn-Tucker şartları (Kuhn-Tucker, 1951) optimum çözüm için gerekenleri sağlamaktadır. Kısmi türevler vektörünün ve ilişik lineer olmayan denklemlerin çözümü çoğu tasarım problemi için bu şartların doğrudan uygulamasını zorlaştırmaktadır. LP ve NLP yöntemleri çelik çerçevelerin minimum ağırlıklı olarak tasarlanması için kullanılmıştır (Mahfouz, 1999).



(a) Gerçek optimizasyon problemi



(b) Lineerleştirilmiş optimizasyon problemi

Şekil 1. (a) Gerçek ve (b) lineerleştirilmiş optimizasyon problemi

1.4.2. Optimumluk Kriteri Yöntemi

Optimumluk kriteri yöntemi (OC), Langraj çarpanları ile sentezlenen lineer olmayan programlamanın Kuhn-Tucker şartlarının doğrudan uygulanmaması ile gerçekleştirilmektedir. Langraj çarpanları ilgili sınırları içerisinde barındırır. OC yöntemi sürekli tasarım değişkenleri ile işlemlerini yürütmekte olup tasarım değişkeni olarak sadece yapısal elemanın enkesit alanını kullanmaktadır. Tüm diğer enkesit özellikleri bu tasarım değişkeninin fonksiyonu olarak yazılmaktadır. Bu yöntemi kullanarak ayırık değişkenli çözümler bulunmak istendiğinde önce optimizasyon problemi sürekli tasarım değişkenlerine göre çözülmekte ve sonra bu çözümlere en yakın ayırık tasarım değişkenleri bulunmaya çalışılmaktadır. Genelde bu dönüşüm esnasında iki problem ile karşılaşmaktadır. Birincisi, profillerin enkesit alanları ile diğer enkesit özellikleri arasında doğrusal bir orantı olmamasıdır. Enkesit alanı bir başka profile göre büyük iken kullanım yönündeki talet momenti küçük olabilir. İkincisi ise sürekli çözümlerin yerine koyulan ayırık çözümler, sınırların tümünü sağlamayan bambaşka bir yapı sistemi ortaya çıkarabilmektedir (Mahfouz, 1999).

1.4.3. Yapay Zeka ve Evrimsel Optimizasyon Teknikleri

Yapay zeka (Artificial intelligence, AI), insanın düşünme yapısını anlamak ve bunun benzerini ortaya çıkaracak bilgisayar işlemlerini geliştirmeye çalışmak olarak tanımlanmaktadır (URL-6, 2003). Yani programlanmış bir bilgisayarın düşünme girişimidir. Daha geniş bir tanıma göre ise, yapay zeka, bilgi edinme, algılama, görme, düşünme, karar verme, öğrenme, gerekçeleme, problem çözme, yabancı bir dili algılama v.b. gibi insan zekasına özgü davranışları gösterebilen sistemlerle ilgilenen bir bilgisayar bilimidir (URL-5, 2003). Yapay zeka konusundaki ilk çalışma McCulloch ve Pitts tarafından yapılmıştır. Bu araştırmacıların önerdiği, yapay sinir hücrelerini kullanan hesaplama modeli, önermeler mantığı, fizyoloji ve Turing'in hesaplama kuramına dayanmaktadır. 1950'lerde Shannon ve Turing bilgisayarlar için satranç programları yazdırmıştı. İlk yapay sinir ağı temelli bilgisayar SNARC, MIT'de Minsky ve Edmonds tarafından 1951'de yapıldı. Çalışmalarını Princeton Üniversitesi'nde sürdüren Mc Carthy, Minsky, Shannon ve Rochester'le birlikte 1956 yılında Dartmouth'da iki aylık bir toplantı düzenledi. Bu toplantıda bir çok çalışmanın temelleri atılmakla birlikte, toplantının en

önemli özelliği Mc Carthy tarafından önerilen *Yapay zeka* adının konmasıdır. Daha sonra Newell ve Simon, “insan gibi düşünme” yaklaşımına göre üretilmiş ilk program olan General Problem Solver (Genel problem çözücü)’ı geliştirmişlerdir. Simon, daha sonra fiziksel simge varsayımını ortaya atmış ve bu kuram, insandan bağımsız zeki sistemler yapma çalışmalarıyla uğraşanların hareket noktasını oluşturmuştur (URL-6, 2003).

AI'nın ana amacı, insanların davranışlarının ve sezgisel yeteneklerinin bilgisayar üzerinde benzetimidir. İnsanoğlu esas olarak bilgiyi kullanmakta ve onu işlemektedir. Bu yüzden bilgi ve bilginin kullanımı AI'nın anahtar karakteristiğidir. AI'nın diğer bir özelliği, zeka, rakam ya da veriler yerine bilgiye dayalı mantıksal bir süreçte, bilgi ve bilginin işlenmesi ile zeki davranışların ortaya çıkarılabilmesidir. Geleneksel programlamaya göre öğrenbilme, tecrübe kazanabilme ve bu tecrübeyi kullanarak yeni problemleri çözebilme gibi üstünlükleri vardır. Ayrıca eksik veri ile problemler çözebilmekte ve belirli bir algoritma yerine sezgisel yöntemler kullanabilmektedirler. En büyük dezavantajı yanlış yapabilme olasılıklarının her zaman varolmasıdır (URL-5, 2003). AI, birçok sistem ve algoritmayı araç olarak kullanabilir. Örneğin bunlardan birkaçı; uzman sistemler ,yapay sinir ağları, bulanık mantık, genetik algoritma (GA), benzetilmiş tavlama (SA) şeklinde sıralanabilir(URL-6, 2003).

1.4.3.1. Yapay Sinir Ağları

Yapay sinir ağları (YSA) genel olarak insan beyninin ya da merkezi sinir sisteminin çalışma prensiplerini taklit eden bilgi işleme sistemleridir. YSA, genel olarak birbirleri ile bağlantılı işlemci birimlerden veya diğer bir ifade ile işlemci elemanlardan (sinir hücreleri) oluşmaktadır. Sinir hücreleri arasındaki bağlantıların yapısı, ağın yapısını belirlemektedir. İstenilen hedefe ulaşmak için bağlantıların nasıl değiştirileceği, öğrenme algoritması tarafından belirlenmektedir. Kullanılan bir öğrenme kuralına göre, hatayı sıfıra indirecek şekilde, ağın ağırlıkları değiştirilmektedir (URL-7, 2003).

YSA, bir öğrenme sürecinden sonra bilgiyi toplama, hücreler arasındaki bağlantı ağırlıkları ile bu bilgiyi saklama ve genelleme yeteneğine sahip paralel dağılmış ve çok hızlı çalışan bir işlemci şeklinde de tanımlanabilmektedir. Her bir işlem elemanı birbiri ile bağlantılıdır. Bu yüzden YSA bazen, paralel dağıtılmış işleme sistemleri (Parallel Distributed Processing Systems) ya da bağlantıcı sistemler (Connectionist systems) olarak da adlandırılmaktadır. Öğrenme süreci, arzu edilen amaca ulaşmak için YSA ağırlıklarının

yenilenmesini sağlayan öğrenme algoritmalarını ihtiva etmektedir. YSA' nın hesaplama ve bilgi işleme gücünü, paralel dağılmış yapısından, öğrenebilme ve genelleme yeteneğinden aldığı söylenebilir. Öğrenme, ilgili problemdeki veri-sonuç ilişkisini en güzel tanımlayacak optimum ağırlıkların bulunmasıdır. Problemden alınan örneklerden faydalanılarak ilgili probleme farklı çözümler sağlayabilmek amacıyla kendisine uygulanan örneklerden öğrenmeye çalışmakta ve uzman sistemler gibi bilgiyi kurallar halinde istememektedir. YSA, ilgilendiği problemi öğrendikten sonra eğitim sırasında karşılaşmadığı test örnekleri için de arzu edilen çözümü sunabilmektedir. Eğitilmiş bir ağa, veri, sadece bir kısmı ya da eksik, bozuk veya daha önce hiç karşılaşmadığı şekilde verilse bile, ağ hafızadan bu girişe en yakınına seçerek tam bir giriş verisi alıyormuş gibi kabul eder ve buna uygun bir sonuç üretir. Ancak YSA çıkardıkları sonuçları nasıl ve neden çıkardığını açıklayamamaktadır (Bu nedenle kapalı kutu olarak nitelendirilir.). Bu özellik ağın genelleştirme özelliğidir. YSA, ilgilendiği problemdeki değişikliklere göre ağırlıklarını ayarlamaktadır. Yani, belirli bir problemi çözmek amacıyla eğitilen YSA, problemdeki değişimlere göre tekrar eğitilebilmekte ve değişimler devamlı ise gerçek zamanda da eğitime devam edilebilmektedir. YSA'nın eğitilmesi oldukça zaman alıcı ve zor bir işlemdir (URL-7, 2003).

YSA, çok sayıda işlemci elemanların (sinir hücresi) bağlantısıyla paralel dağılmış bir yapıya sahip olup ağın sahip olduğu bilgi, ağdaki tüm bağlantılara gönderilmektedir. Bu nedenle, eğitilmiş bir YSA'nın bazı bağlantılarının hatta bazı hücrelerinin etkisiz hale gelmesi, ağın doğru bilgi üretmesini önemli ölçüde etkilememektedir. Bu nedenle, geleneksel yöntemlere göre hatayı giderme yetenekleri son derece yüksektir (URL-7, 2003).

YSA'nın en önemli özelliklerinden biri de bilgiyi depolamalarıdır. Sinirsel hesaplamalarda bilgi ağırlıklar üzerine dağıtılmaktadır. Bağlantıların ağırlıkları sinirsel ağın hafıza birimidir. Bu ağırlıklar ağın o andaki sahip olduğu bilgiyi veya uygulanan örneklerden öğrenmiş olduğu davranışı vermektedir. Bu bilgiler, ağdaki birçok ağırlıklar üzerine (hafıza birimine) dağıtılmaktadır (URL-7, 2003).

YSA'nın temel işlem elemanı olan hücrenin yapısı ve modeli, bütün YSA yapılarında yaklaşık aynıdır. Dolayısıyla, YSA' nın farklı uygulama alanlarındaki yapıları da standart yapıdaki bu hücrelerden oluşmaktadır. Bu nedenle, farklı uygulama alanlarında kullanılan YSA, benzer öğrenme algoritmalarını ve teorilerini paylaşabilirler. Bu özellik, problemlerin YSA ile çözümünde önemli bir kolaylık getirmektedir. Ayrıca YSA'nın

temel işlem elemanı olan hücre doğrusal değildir. Dolayısıyla hücrelerin birleşmesinden meydana gelen YSA da doğrusal değildir ve bu özellik bütün ağa yayılmış durumdadır. Bu özelliği ile YSA, doğrusal olmayan karmaşık problemlerin çözümünde en önemli araç olmaktadır (URL-7, 2003).

Son yıllarda YSA, özellikle günümüze kadar çözümü güç, karmaşık veya ekonomik olmayan çok farklı alanlardaki problemlerin çözümü için uygulanmış ve genellikle başarılı sonuçlar alınmıştır. Robotik uygulamaları, proses kontrolü, ürün tasarımı, işlem planlama kalite kontrolü, gerçek zamanlı modelleme, adaptif kontrolü, görüntü tanıma, borsa endeksi, enflasyon ve kur tahmini, v.b. daha birçok alanda uygulanabilirliği mevcuttur (URL-6, 2003). YSA, kimya mühendisliği, inşaat ve yapı mühendisliği, elektrik ve elektronik mühendisliği, imalat ve makine mühendisliği, sistem ve kontrol mühendisliği alanlarında da önemli birçok mühendislik problemlerinin çözümünde kullanılmışlardır. Örneğin, İnşaat ve yapı mühendisliğinde; konstrüksiyon projelerinde kaynak seviyelerini belirleme, bir rezervuardan çıkış kontrolü, biyolojik bilgiler yardımıyla nehir suyu kalitesinin sınıflandırılması, nehirlerin akışının tahmin edilmesi, sonlu-eleman-temelli yapısal analiz işleminin modellenmesi, yapı malzemelerinin iç yapılarındaki çatlakların tespit edilmesi ve depreme maruz betonarme çerçevelerde emniyetli yatay taşıyıcı tahmininde uygulanmıştır (URL-7, 2003).

1.4.3.2. Bulanık Mantık

Bulanık mantık (Fuzzy logic) kavramı ilk kez 1965 yılında California Berkeley Üniversitesinden Prof. Lotfi A.Zadeh'in bu konu üzerinde ilk makalelerini yayınlamasıyla ortaya çıkmıştır. O tarihten sonra önemi gittikçe artarak günümüze kadar gelen bulanık mantık, insanların her gün kullandığı ve davranışlarının yorumlandığı yapıya ulaşılmasını sağlayan, belirsizliklerin anlatımı ve belirsizliklerle çalışılabilmesi için kurulmuş katı bir matematiksel disiplin olarak tanımlanabilir (URL-6, 2003). Geleneksel mantık sistemi yalnızca 1 ve 0 üzerine kuruludur. Doğru veya yanlış vardır. Bu ikisinin arası yoktur. Belirsiz bir problemin çözümü güçtür. Bulanık mantık sisteminde de 1 ve 0 değerleri vardır. Bununla birlikte 0 ile 1 arasındaki değerler de kullanılmaktadır. Doğru ya da yanlışın ne kadar doğru ya da ne kadar yanlış olduğu belirlenebilmektedir (URL-5, 2003). Bulanık mantık istatistik ve olasılık kuramından farklı olarak kesinliklerle değil

belirsizliklerle çalışmaktadır. Olasılıkta problemin kendisi tanımlıdır. Bulanık mantık ise problemin tanımının tam olmasına gereksinim duymamaktadır (URL-6,2003).

Bulanık kuramının merkez kavramı bulanık kümeleridir. Küme kavramı kulağa biraz matematiksel gelebilir ama anlaşılması kolaydır. Örneğin “orta yaş” kavramı incelenecek olursa, bu kavramın sınırlarının kişiden kişiye değişiklik gösterdiği görülmektedir. Kesin sınırlar söz konusu olmadığı için orta yaş kavramı matematiksel olarak da kolayca formüle edilememektedir. Ama genel olarak 35 ile 55 yaşları orta yaşlılık sınırları olarak düşünülebilir. Bu kavrama ait çizilen eğriye “aitlik eğrisi” adı verilir ve kavram içinde hangi değer hangi ağırlıkta olduğunu göstermektedir. Bir bulanık kümesi kendi aitlik fonksiyonu ile açık olarak temsil edilebilir. Aitlik fonksiyonu 0 ile 1 arasındaki her değeri alabilir. Böyle bir aitlik fonksiyonu ile “kesinlikle ait” veya “kesinlikle ait değil” arasında istenilen incelikte ayarlama yapmak mümkündür (URL-6, 2003).

Bulanık mantıkta bulanık kümeleri kadar önemli bir diğer kavramda linguistik değişken kavramıdır. Linguistik değişken “sıcak” veya “soğuk” gibi kelimeler ve ifadelerle tanımlanabilen değişkenlerdir. Bir linguistik değişkenin değerleri bulanık kümeleri ile ifade edilmektedir. Örneğin, oda sıcaklığına, linguistik değişken olarak “sıcak”, “soğuk” ve “çok sıcak” gibi günlük hayatta; tam olarak tanımlanmamış ve sayısal olmayan dilsel niteleyiciler kullanarak kararlar verilebilir. Bu üç ifadenin her biri ayrı ayrı bulanık kümeleri ile modellenmektedir (URL-6, 2003).

Bulanık mantık ile matematik arasındaki temel fark bilinen anlamda matematiğin sadece aşırı uç değerlerine izin vermesidir. Klasik matematiksel yöntemlerle karmaşık sistemleri modellemek ve kontrol etmek bu yüzden zor olmaktadır. Çünkü veriler tam olmalıdır. Bulanık mantık kişiyi bu zorunluluktan kurtarmakta ve daha niteliksel bir tanımlama olanağı sağlamaktadır. Bir kişi için 38,5 yaşında demektense sadece orta yaşlı demek bir çok uygulama için yeterli bir veridir. Böylece azımsanamayacak ölçüde bir bilgi indirgenmesi söz konusu olacak ve matematiksel bir tanımlama yerine daha kolay anlaşılabilen niteliksel bir tanımlama yapılabilecektir (URL-6, 2003).

Bulanık mantığın sağladığı en büyük fayda ise “insana özgü tecrübe ile öğrenme” olayının kolayca modellenebilmesi ve belirsiz kavramların bile matematiksel olarak ifade edilebilmesine olanak tanınmasıdır. Bu nedenle lineer olmayan sistemlere yaklaşım yapabilmek için özellikle uygundur (URL-6, 2003). Bulanık mantık sistemleri ne öğrenebilmekte ne de öğretebilmektedir. Ayrıca bulanık mantığın uygulama alanları çok

geniştir. Örneğin elektrikli ev aletleri, oto elektroniği ve fren sistemleri, elektronik denetim sistemleri, karar verme, proses planlamada kullanılabilir (URL-5, 2003).

1.4.3.3. Uzman Sistemler

Uzman sistemler, belirli bir alanda, bir uzmanın önerdiği çözümleri üretebilen, o alanın bilgileri ile donatılmış, gerekçeleme yöntemleri ile olayları süzebilen programlardır (URL-8, 2003). Bilgi Tabanlı Sistemler (Knowledge Based Systems) olarak da bilinmektedirler (URL-5, 2003). Uzman sistem tabiri kullanılmasının sebebi, sistemin bir veya daha fazla uzmanın bilgilerine sahip olarak onun veya onların yerini almaya yönelmesinden dolayıdır (URL-8, 2003). Uzman sistemler, YSA'nın aksine, çıkardıkları sonuçları nasıl ve neden çıkardığını açıklayabilmekte ve kurallar üretilerek rahatlıkla değiştirebilmektedirler. En büyük dezavantajları, Uzman sisteme bilgiyi verecek uzman bulmanın her zaman kolay olmaması ve uzmanların bilgilerini kurallar halinde belirtememesi olasılığının varolmasıdır (URL-5, 2003). Bir uzman sistem, geliştirme çevresi ve görüşme çevresi olmak üzere, iki ana parçanın birleşiminden oluşmaktadır. Geliştirme çevresi sistemin bileşenlerini kurmak ve uzman insan bilgilerini bilgi tabanına girmek için uzman sistemi kuranlar tarafından kullanılmaktadır. Görüşme çevresi ise uzman bilgi ve nasihatlerine ulaşabilmek için uzman olmayanlar tarafından kullanılmaktadır. Bir uzman sistem: 1. Bilgi kazanma, 2. Bilgi tabanı, 3. Çıkarım mekanizması, 4. Çalışma alanı, 5. Kullanıcı arabirimi 6. Açıklama, 7. Düşünme kapasitesini iyileştirme şeklindeki alt birimlerden oluşmaktadır (URL-8, 2003).

Bilgi kazanma, bazı bilgi kaynaklarından bir bilgisayar programına problem çözümü için bilgi aktarma ve dönüştürme işlemlerinden oluşmaktadır. Potansiyel bilgi kaynakları uzman insanlar, kitaplar, veri tabanları, özel araştırma raporları ve kullanıcının kendi deneyimleri olabilmektedir (URL-8, 2003).

Bilgi tabanı, bilgi tabanı problemlerin anlaşılması, formülasyonu ve çözümü için gerekli olan tüm bilgileri içermektedir. Örneğin olaylar ve durumlar hakkında bilgi ve bunlar arasındaki mantıksal ilişki yapılarını ihtiva etmektedir. Ayrıca standart çözüm ve karar alma modellerini de içermektedir (URL-8, 2003).

Çıkarım mekanizması, uzman sistemin beynidir. Bilgi tabanı ve çalışma alanında bulunan bilgiler üzerine düşünmek için bir yöntem bilimi sunan ve sonuçları biçimlendiren bir bilgisayar programıdır. Bir başka deyişle problemlere çözümler üreten bir

mekanizmadır. Burada sistem bilgisinin nasıl kullanılacağı hakkında kararlar alınmaktadır (URL-8, 2003).

Çalışma alanı, giriş verileri tarafından belirlenmiş problem tanımları için hafızanın bir köşesinde bulunan çalışma alanıdır. Bu alan işlemlerin ara seviyelerindeki sonuçları kaydetmek için de kullanılmaktadır (URL-8, 2003).

Kullanıcı arabirimi, uzman sistemler için, kullanıcı ile bilgisayar arasında probleme yönelik iletişimin sağlanması için bir dil işleyicisidir. Bu iletişim, en sağlıklı doğal dil ile yapılmaktadır. Kısaca kullanıcı ara birimi kullanıcı ile bilgisayar arasında bir çevirmen rolünü üstlenmiştir (URL-8, 2003).

Açıklama modülünden kasıt, kullanıcıya çeşitli yardımların verilmesi ve soruların açıklanması olduğu kadar, uzman sistemin çıkardığı sonucu nasıl ve neden çıkardığını açıklayabilmesidir. Burada uzman sistem karşılıklı soru cevap şeklinde davranışlarını açıklamaktadır. Uzman sistemleri diğer sistemlerden farklı yapan bir özelliği de açıklama modülüdür (URL-8, 2003).

Düşünme kapasitesini iyileştirme, bir uzman insanın kendi performansını analiz edebilmesini, öğrenebilmesini ve gelecekteki kullanım için onu iyileştirebilmesini baz alan bir mekanizmadır. Sistemin kendini iyileştirmesi öğrenme ile ilgili bir konudur. Sistemlerin bir uzman insan gibi öğrenebilmelerine yönelik çalışmalar sinirsel ağlar üzerinde sürdürülen araştırmalarla devam etmektedir (URL-8, 2003).

Uzman sistemlerin kullanım alanlarından bazıları: 1. Proses tasarımı ve seçimi, 2. Ürün tasarımı, 3. İşlem planlama, 4. Medikal Tedavi, 5. Kalite Kontrol, 6. Çizelgeleme, 7. Ses işleme, 8. Görüntü tanıma, 9. Robotik uygulamaları ve 10. Hata düzeltmeleri şeklinde sıralanabilir (URL-5, 2003).

1.4.3.4. Ardışık Optimizasyon (Hill Climbing, HC) Yöntemi

Ardışık Optimizasyon (Hill Climbing) yöntemi(HC), bir deneme yanılma yöntemi olup fonksiyonu maksimize etmeyi amaçlamakta ve yerel (lokal) çözüm bulmayı garanti etmektedir. Yakınsama keyfi bir noktadan ardışık bir noktaya geçerek sağlanmakta yani her hangi bir X noktası etrafında bir fonksiyon verildiğinde, fonksiyonu X+1 ve X-1 noktaları içinde denemektedir. Bu üç değer en iyisini X kabul edip tekrar aynı işlemleri tekrar etmektedir. Bundan dolayı HC'nin iraksaması, yakınsama işleminin uzun zaman almasına neden olabilmektedir.

Ardışık optimizasyon yöntemi türev gibi bir yardımcı bilgiye ihtiyaç duymamaktadır. Ayrıca ardışık optimizasyon yöntemiyle araştırma uzayı taranırken ya çok geniş bir aralıkla ya da çok dar bir aralıkla arama yapıldığından global çözümler atlanabilmekte ve yerel (lokal) çözümlerin bulunma şansı daha fazla olmaktadır (Kılıç, 1997).

1.4.3.5. Benzetilmiş Tavlama (Simulated Annealing)

SA, metallere ısı işlemlerin uygulanmasından etkilenen bir tekniktir. Yüksek sıcaklıklarda bir maddenin atomları serbestçe hareket etmektedir. Likit haldeki bu madde kademeli olarak soğutulursa atomlar serbestçe hareket etme yeteneklerini kaybetmekte ve kendilerine has, homojen (kusursuz) kristalleşme meydana gelmektedir. Bu olay SA ile çözülecek problem için global optimum çözümü nitelendirir. Eğer hızlı bir soğutma yapılırsa belli bir yapıya sahip olmayan heterojen bir kristalleşme olmaktadır. Buda homojen kristallere göre daha yüksek enerji seviyeli yerel optimum çözümler oluşturmaktadır. Dikkat edilmesi gereken nokta, atomlara homojen ve kusursuz dizilime (kristalleşmeye) izin verecek soğutma işleminin hassas olarak gerçekleştirilmesidir (URL-2, 2003). SA'nın doğası gereği yavaş soğutma, maddenin minimum enerji seviyesine ulaşmasına imkan vermektedir (URL-3, 2003).

SA yönteminin özü, sistemin ısıtılıp soğutulmasıyla bulunduğu enerji seviyesinden daha düşük enerji seviyelerine geçme fırsatı yakalamaya çalışmasından ibarettir. (URL-1, 2003). Böylelikle yerel minimum enerji seviyelerinden ziyade daha düşük enerji seviyelerine doğru bir geçiş sağlanmaktadır.

1970 yılının ortalarında Kirkpatrick ve arkadaşları çizilmiş bir elektrik devresinin optimum şeklini bulmak için SA yöntemini kullanmış olup elde bulunan çözümden hareketle keyfi olarak başka bir çözüm bulmuşlardır. Fizik terminolojisini kullanarak sistemin enerjisini minimum ederlerken biyoloji terminolojisini kullanarak sistemin uygunluğunu yani etkinliğini maksimum etmek için uğraşmışlardır. SA'da, çözüme gidilmesi istenen problem için çözüm olabilecek x değerinin uygunluğu $f(x)$ fonksiyonu olduğunda, metod gereği $f(x_{i+1}) < f(x_i)$ olduğu kabul edilmektedir. Eğer $f(x_i) < f(x_{i+1})$ ise bu durum x değerindeki değişimin (daima pozitif) sıcaklığa oranının negatif işaretlisinin eksponansiyeli olasılığında kabul edilebilmektedir (URL-4, 2003). Bu olasılık tarifi daha sonra araştırmacılara göre değişmiştir. Ancak temelde aynı mantık mevcuttur. Bir x_i tekil değerinden başka bir x_{i+1} değerine geçerken olasılık bir ise yeni çözüm eskisinden iyi, sıfır

ile bir arasında ise kötüdür (Rasheed, 1998). Başlangıçta SA'yı kötü sonuçlara yönelten olasılık değeri araştırmanın başlangıç safhasını genişletmekte ve araştırma sürecini kısaltmaktadır (URL-4, 2003). Kirkpatrick ve arkadaşları da başlangıçta olasılık değerini küçülten sıcaklık faktörünün araştırma havuzunun kapsamlı olarak taranmasına vesile olduğunu ve bu sayede etkili sonuçlara ulaşıldığını bulmuşlardır (URL-4, 2003). Araştırma uzayının kapsamlı şekilde taranması, bulunan çözümün etkinliğini, diğer bir deyişle (global) optimum olma şansını artırmaktadır. Sıcaklık faktörü azalırken yani olasılık değeri küçülürken x değerleri yani araştırma uzayındaki azalma adımının (hızının) yaratabileceği atlamalar, araştırma uzayının kapsamlı taranamamasına dolayısıyla bulunacak çözümün (yerel) optimum olmasına neden olabilmektedir. Anlaşılabilirliği gibi gereğinden hızlı soğuma, sonucu, globalden yerel optimuma götürebilmektedir. Ayrıca SA yöntemi, ayrık değişkenli optimizasyonda kullanılabilirliği gibi sürekli değişken e sahip optimizasyonda da kullanılabilirliği.

1.4.3.6. Genetik Algoritma

Genetik sistemler mühendislik amaçlı optimizasyon olarak düşünülmeden önce biyologlar, 1950'li yılların başlarında genetik sistemlerin benzeştirildiği bilgisayarlar kullanmışlardır. 1962'de de Fraser genetik araştırmanın, fonksiyonların optimizasyonu için uygulanabilir olduğunu göstermiştir (Chen, 1997). Aynı tarihte John Holland evrim sürecinin bir bilgisayar yardımıyla kullanılarak, bilgisayara anlayamadığı çözüm yöntemlerinin öğretilerilebileceğini düşünmüştür. GA, John Holland'ın bu düşüncesinin bir sonucu olarak bulunmuştur (URL-9, 2003). Daha sonra ilk olarak Bagle'nin tezinde kullanılmıştır (Goldberg, 1989). Ayrıca De Jong, GA'nın fonksiyonların minimize edilmesinde kullanıldığını resmen bulan ilk kişi olmuştur (Chen, 1997).

Günümüzün karmaşık ve zor koşulları, problemlere hızlı ve kolay çözüm veren yeni çözüm yöntemleri arayışına neden olmuştur. Evrimsel yaklaşımlardan biri olan GA da, bu arayışlar içinde önemli bir yer tutmaya başlamıştır (Emel ve Taşkın, 2002). GA, doğal seçim ilkelerine dayanan bir mekanizma yardımıyla arama yapan bir optimizasyon yöntemi olup, yapay zekanın kolu olan evrimsel hesaplama tekniğinin bir parçasıdır (URL-9, 2003). GA'nın temel özelliği en iyinin hayatta kalması ve adaptasyondur. Bir GA'da potansiyel çözümler ile ilgili bilgiler, diziler (string) şeklindeki sayılarla ifade edilmektedirler. Dizilere kromozom da denmektedir. Kromozomlar ise genlerin

sıralanmasından oluşmaktadır. GA ile optimizasyonun ilk adımında, çoğunlukla rasgele sayı üretilerek bir grup çözüm oluşturulmaktadır. Başlangıçta iyi bir yaklaşık çözüm için, herhangi bir bilgiye ihtiyaç duyulmamaktadır. Başlangıç topluluğunun üretilmesinin ardından, her çözümün uygunluğu veya iyiliği, seçilen bir uygunluk fonksiyonu kullanılarak değerlendirilmektedir. Uygunluk fonksiyonu, eldeki veriler dikkate alınarak belirlendiği için, bunun seçimi probleme bağlıdır. Uygunluk işlemi, daha iyiyi seçmek için elde edilen iyi çözümlerden daha iyi çözümlere ulaşıncaya kadar devam ettirilmektedir. Seçme, çaprazlama ve mutasyon gibi genetik operatörler, elde edilen iyi çözümlerden denenmiş yeni çözümler üretmek için kullanılmaktadırlar. Bu iyileştirme işlemi, daha önceden belirlenen bir iterasyon sayısına veya tatmin edici bir sonuca ulaşıncaya kadar devam ettirilmektedir (URL-7, 2003).

GA'yı diğer evrimsel olmayan tekniklerden ayıran temel özellikleri şöyle sıralanabilmektedir (Chen, 1997):

1. GA, problemin gerçek parametrelerinden ziyade kodlanmış tasarım değişkenleri ile çalışmaktadır.
2. GA, araştırma uzayı içerisinde nihai çözümü tek bir aday çözümden ardışık adımlarla değil, farklı çözümlerin oluşturduğu topluluktan hareket ile aramaktadır.
3. GA, türev, integral gibi kompleks matematiksel bilgiden ziyade sadece uygunluk ve amaç fonksiyonunun değerine ihtiyaç duymaktadır.
4. GA, çözüme deterministik kurallar yerine probabilistik kurallar (genetik operatörler) ile gitmektedir.

GA ile SA arasında bazı benzerlik ve farklılıklar vardır. Her iki yöntem de yerel (local) optimizasyondan ziyade "global" optimizasyona meyillidir (Rasheed, 1998). Ayrıca GA noktalar kümesinden çözüme ulaşırken SA tek bir noktadan hareket ederek sonuca varmaktadır. Öte yandan SA araştırma alanının bir kısmından diğer kısmına odaklanması GA'ya göre daha hızlıdır. Hangi yöntemin daha etkin olduğu konusunda tartışmalar sürmektedir. Yapı mühendisliği açısından, GA ,tek bir çözümle değil, çözümler topluluğu ile araştırma yapması ve her iki yöntemde de eski çözümden yeni çözüme geçerken mutasyon operatörünün yanı sıra GA'nın ayrıca çaprazlama operatörünü de kullanması onun daha çok kabul görmesine neden olmaktadır (URL-4, 2003).

GA diğer evrimsel algoritmalar ve sistemler ile birlikte kullanılabilir. Örneğin GA ve SA ,GA ve bulanık mantık birlikte kullanılarak melez sistemler geliştirilmektedir (Chen, 1997). Ayrıca GA'nın, fonksiyon optimizasyonu, çizelgeleme, mekanik öğrenme,

tasarım, hücresele üretim gibi alanlarda başarılı uygulamaları bulunmaktadır (Emel ve Taşkın, 2002).

1.5. Çalışmanın Amaç ve Kapsamı

Bu çalışmanın amacı, ayırık tasarım değişkenleri kullanılarak geliştirilen bir GA ile çelik çerçevelerin optimizasyonu gerçekleştirmektedir. Kullanılan teknik ve terminoloji ile ilgili detaylı tanımlamalar yapılmaktadır. Ayrıca GA'nın temel karakteristikleri de belirtilmektedir.

TS 648 ve AISC-ASD (American Institute of Steel Structure-Allowable Stress Design, 1984) standartlarına göre çelik yapıdaki elemanların gerçeklemeleri yapılmıştır. Bu amaç doğrultusunda FORTRAN dilinde bir bilgisayar programı hazırlanmıştır. Programın analiz kısmı için Zaduryan (1999) tarafından geliştirilen program kodundan, çerçeve sistemlerin rijitlik matrislerinin oluşturulmasında ise Tezcan (1970)'dan yararlanılmıştır. Programın analiz kısmı matris deplasman yöntemi kullanılarak geliştirilirken sınırlayıcı olarak yerdeğiştirme, gerilme, stabilite, sehim ve geometrik uygunluklar kullanılmıştır. Ayrıca etkili boy faktörünün sistemin ötelenmiş olup olmamasına göre karar verip belirlenmesini sağlayan mekanizma ile stokastik operatörlerin uygulanma tarzları dikkate alınarak hazırlanmış olan akıllı mutasyon ve çok noktalı çaprazlama operatörleri kullanılmıştır.

2. YAPILAN ÇALIŞMALAR, BULGULAR VE İRDELEME

2.1. Çelik Çerçevelerin AISC-ASD ve TS 648 Göre Tasarımı

Yapı tasarımı yapılırken ilk önce inşası yapılacak yapı için uygun bir sistem seçimi yapılmakta ve hangi yük durumlarının dikkate alınması gerektiği saptanmaktadır. Daha sonra yapının kullanım amacına bağlı olarak kritik yük durumları belirlenmekte ve bu kriterleri sağlayan güvenli bir sistem seçilmektedir. (Mahfouz, 1999)

Tasarımda ikinci önem ekonomiye verilmektedir. Ne yazık ki, ilgili yapı için doğru yüklerin (özellikle hareketli, rüzgar ve deprem yükleri) saptanmasındaki belirsizlikler, malzemedeki belirsizlikler, inşaat sırasındaki geçici yüklemeler ve tahmin edilemeyen etkilerden dolayı ekonomik bir tasarımın elde edilmesi oldukça zordur. Yapı maliyetine etkileyen; eleman boyutları, birleşim detayları, yapının ömrü süresince oluşacak bakım masrafları, gibi bir çok parametre mevcuttur. Bu parametrelerin maliyete olan etkilerini azaltmak bir bakıma sistem seçimi ile orantılıdır. Yapının sistemi, yapı elemanlarında meydana gelecek kesit etkilerini doğrudan etkilediğinden yapı elemanlarının boyutlandırılmasında en etkin faktördür. Değınilen bu parametrelerin yanı sıra Cao (1996),

- geleneksel olmayan boyutlarda enkesitlerin kullanılması (hazır enkesitler dışında özel yapım enkesitlerin kullanılması),
- farklı enkesit boyutlarına sahip profiller kullanılarak komplike birleşimlerin oluşturulması,
- yapının bütünü düşünülmeden, yapı elemanları için ayrı ayrı uygun enkesit boyutlarının belirlenmesi,
- büyük yapılarda inşaat alanında oluşturulamayan ve yalnızca fabrika üretimi yapılabilen yapı kısımlarının inşa alanına getirilmesinde ilgili kısmın büyüklüğünün mevcut yol alt yapısına ve taşıma kapasitesine göre dikkate alınması

gibi faktörleri sıralamaktadır.

Yapı bütününe dikkate alınmadan yapı elemanlarının ferdi olarak boyutlandırılması, komşu elemanların boyutlarının farklı olmasına neden olduğundan birleşimlerinde maliyet artışı olmaktadır. Bu çalışmada kullanılacak optimizasyon yöntemi, elemandan ziyade

yapının tümünü kapsayan boyutlandırmayı gerçekleştirdiği için bu tür sakıncaları baştan elemektedir.

Tasarım, iki temel aşamada gerçekleştirilmektedir. Birincisi sistemin analizi yapıldıktan sonra elemanlardaki kesit etkilerinin belirlenmesi, ikincisi ise yapı elemanlarının standartlara göre boyutlandırılmasıdır (Mahfouz, 1999). Bu çalışmada sistem çözümü için matris deplasman yöntemi seçilmiştir. Boyutlandırmada ise kesit tayini veya kesit denetimi olmak üzere, iki yöntem bulunmaktadır. Optimizasyonda kullanılacak yöntem gereği, kesit denetimi ile boyutlandırma yapılmaktadır. Kesit denetimi ister emniyet gerilmeleri yöntemi ile isterse de taşıma gücü yöntemi ile sağlanabilmektedir. Emniyet gerilmeleri yöntemi kullanılmış olup hem AISC-ASD hem de TS 648 (T.S.E., 1980) göre işlem yapılmıştır. Bu bölümde iki standartın benzer formülleri, aralarındaki farklılıklar vurgulanarak aktarılmaktadır. Bu çalışmada TS 648 için k gf-cm, AISC-ASD için de kip-in kullanılmaktadır.

2.1.1. Emniyet Gerilmeleri

Eksenel basınca çalışan elemanlarda gerilme problemlerinin yanı sıra bir de stabilite problemleri mevcuttur. Bunlar 1) Eksenel Stabilite (Burkulma), 2) Yanal Burkulma şeklinde sıralanabilir. Bu problemler emniyet gerilmelerinin belirlenmesinde dikkate alınmaktadır.

Eksenel burkulma, kuvvet düzleminde ve sadece eksenel kuvvetten dolayı meydana gelmektedir. Yanal burkulma ise kuvvet düzlemine dik düzlemde meydana gelmektedir. Bu problem eksenel kuvvet ve/veya eğilme etkisi mevcut iken oluşmaktadır.

2.1.1.1. Eksenel Stabilite (Burkulma)

Eksenel basınca çalışan elemanlarda stabilite problemi gözönünde tutularak her iki standartta emniyet gerilmesi belirlenmiştir. İlgili formülasyon aşağıda adım adım verilmektedir. λ_p , plastik narinlik sınırı, E elastisite modülünü, σ_a akma gerilmesini göstermek üzere

$$\lambda_p = \sqrt{\frac{2\pi^2 E}{\sigma_a}} \quad (1)$$

bağıntısıyla verilmekte ve burkulma emniyet gerilmesi, σ_{bem} , elastik burkulma için ($\lambda > \lambda_p$) n emniyet katsayısını (TS 648 için n=2,5 iken AISC-ASD için n=23 / 12) göstermek üzere

$$\sigma_{bem} = \frac{\pi^2 E}{n\lambda^2} \quad (2)$$

bağıntısıyla plastik burkulma için ($\lambda < \lambda_p$)

$$\sigma_{bem} = \frac{\left[1 - \frac{1}{2} \left(\frac{\lambda}{\lambda_p}\right)^2\right] \sigma_a}{n} \quad (3)$$

bağıntısıyla verilmektedir. Plastik burkulmada n emniyet katsayısı

$$n = 1,5 + 1,2 \left(\frac{\lambda}{\lambda_p}\right) - 0,2 \left(\frac{\lambda}{\lambda_p}\right)^3 \geq 1,67 \quad (\text{TS 648}) \quad (4)$$

$$n = \frac{5}{3} + \frac{3}{8} \left(\frac{\lambda}{\lambda_p}\right) - \frac{1}{8} \left(\frac{\lambda}{\lambda_p}\right)^3 \quad (\text{AISC-ASD}) \quad (5)$$

bağıntıları ile ifade edilmektedir. AISC-ASD deki n emniyet katsayısı kendiliğinden $n \geq 1,67$ şartını sağlamaktadır.

S_{k_x} ve S_{k_y} çubuğun x-x ve y-y asal eksenlerine dik düzlemdeki burkulma boylarını, i_x ve i_y bu asal eksenlere göre atalet yarıçaplarını göstermek üzere bu iki eksene göre λ narinlik oranı

$$\lambda_x = \frac{S_{k_x}}{i_x} \quad (6)$$

$$\lambda_y = \frac{S_{k_y}}{i_y} \quad (7)$$

bağıntılarıyla belirlenmektedir.

Yapı elemanının narinliği, bağıntı (6) ve (7)'den belirlenen değerin büyüğüdür. Çerçevelerin basınç çubuklarının burkulma boyu katsayısı olan k , çerçevenin yanal ötelenme yapıp yapmamasına bağlı olarak ister abaklar isterse de, G_A ve G_B rijitlik oranlarını göstermek üzere

$$\frac{G_A G_B}{4} \left(\frac{\pi^2}{k^2} \right) + \left(\frac{G_A + G_B}{2} \right) \left(1 - \frac{\pi/k}{\tan \pi/k} \right) + \frac{2}{\pi/k} \tan \pi/2k = 1 \quad (\text{Y. \u00f6t. \u00f6nlenmi\u015f}) \quad (8)$$

$$\frac{G_A G_B (\pi/k)^2 - 36}{6(G_A + G_B)} = \frac{\pi/k}{\tan \pi/k} \quad (\text{Y. \u00f6t. \u00f6nlenmemi\u015f}) \quad (9)$$

bağıntıları yardımıyla belirlenebilmektedir.

Bu bağıntılarda rijitlik oranları; I_C ve I_B burkulma d\u00f6zlemindeki kolon ve kiri\u015fin atalet momentleri, L_C ve L_B kolon ve kiri\u015flerin akstan aksa uzunluklarını g\u00f6stermek \u00fczere

$$G_A = \frac{\sum I_C / L_C}{\sum I_B / L_B} \quad (10)$$

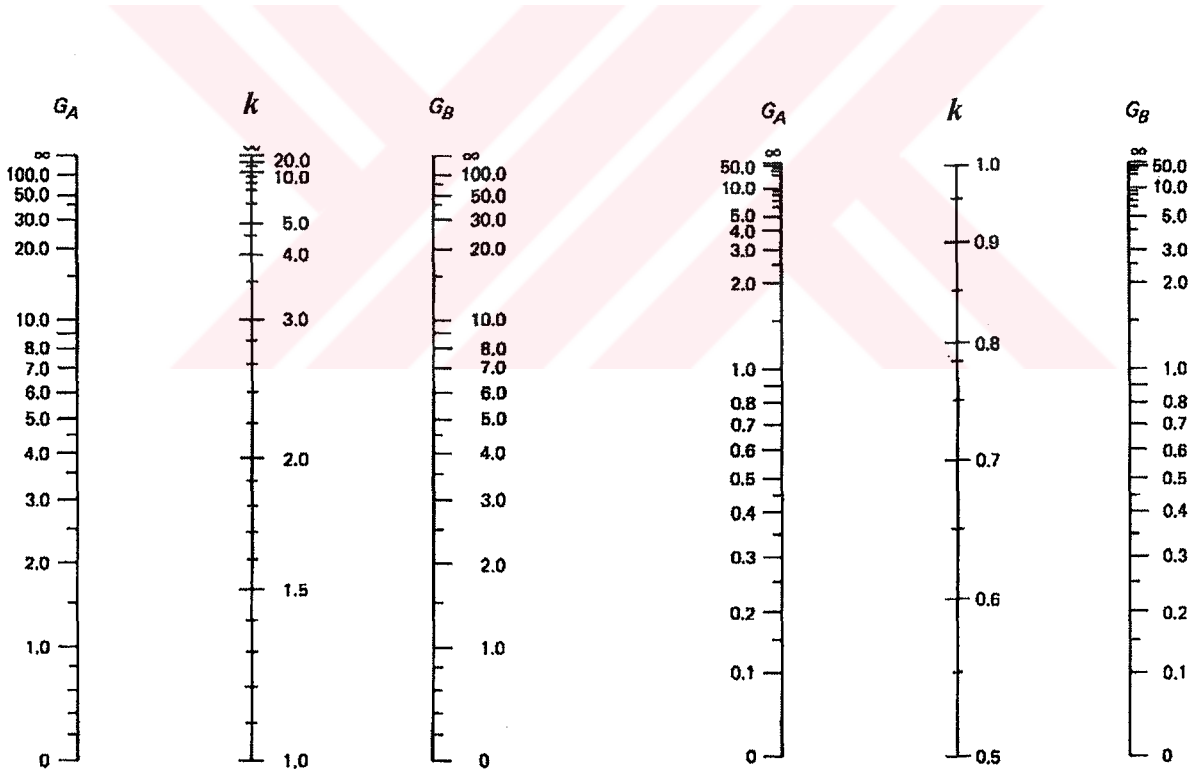
$$G_B = \frac{\sum I_C / L_C}{\sum I_B / L_B} \quad (11)$$

bağıntılarıyla belirlenmektedir.

E\u011fer A veya B ucundan biri ankastre ise $G=1$, basit mesnetli ise $G=10$ dur.

Bu bağıntılardan da g\u00f6r\u00fclebilece\u011fi gibi k katsayısını hesaplamak oldukça g\u00fc\u00e7t\u00fcr. Bu probleme Mahfouz (1999) en k\u00fc\u00e7\u00fck kareler y\u00f6ntemi ve genetik algoritma ile polinomlar tanımlayarak \u00e7\u00f6z\u00fcm aramıştır. Bu \u00e7alı\u015mada da k katsayısı i\u00e7in en k\u00fc\u00e7\u00fck kareler y\u00f6ntemi

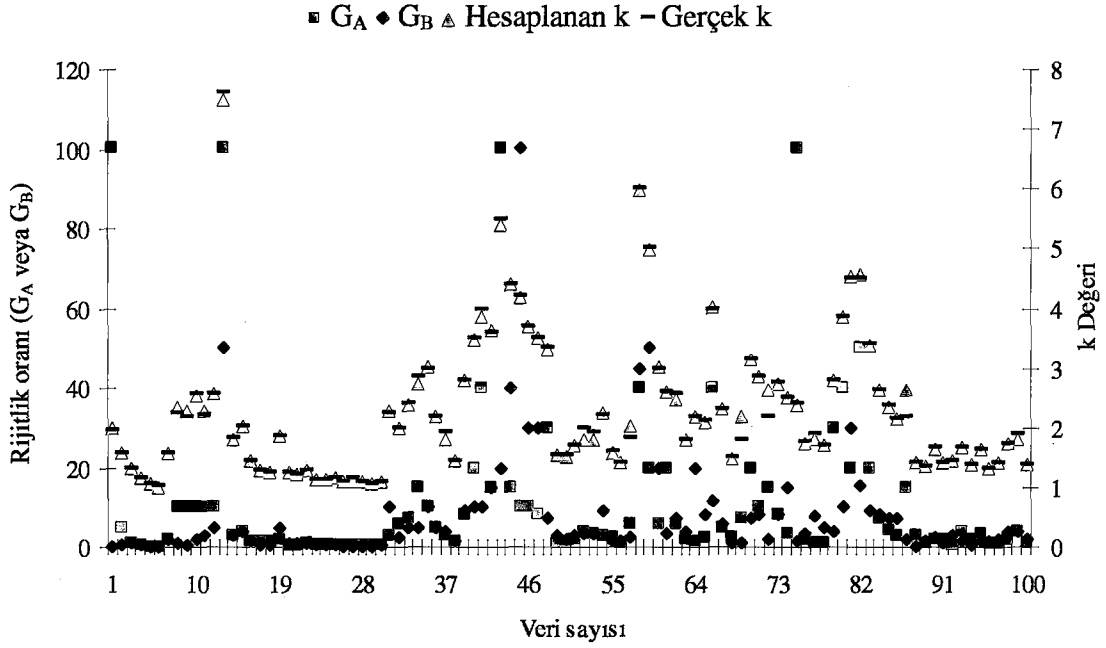
kullanılarak G_A ve G_B 'yi deęişken kabul eden 3.,4.,5.,6.ve 7. dereceden polinomlar bulunmuştur. Ayrıca (8) ve (9) denklemlerinde sayısal analiz yöntemleriyle k deęeri için çözüm aranmıştır. Bu yöntem ve polinomlardan bulunan k deęerleri, her G_A ve G_B için gerçek deęerleriyle örtüşmemektedir. Bu nedenle G_A ve G_B 'nin deęerlerine göre deneme yanılmayla deęer aralıkları belirlenmiştir. Her bir deęer aralığında gerçek k deęerine en yakın sonucu veren yöntem veya polinomlar kullanılmıştır. Polinomları oluşturmak için Şekil 2'de gösterilen abaklardan okumalar yapılmış olup bu deęerler için hesaplanan ve abaklardan okunan burkulma boyu katsayıları, ötelenmesi önlenmemiş ve önlenmiş sistemler için sırasıyla Şekil 3 ve 4'de verilmektedir. Bazı G_A ve G_B deęerleri için sapmalar görülebilmekle beraber genel olarak, hesaplanan ve gerçek k deęerleri örtüşmektedir. Bu sapmaların deęeri, okumalarda yapılan olası hatalardan ve okuma sayısından etkilenmektedir. Eđer okuma sayısı artırılır ve daha hassas okuma yapılabilirse hata payı azaltılabilmektedir.



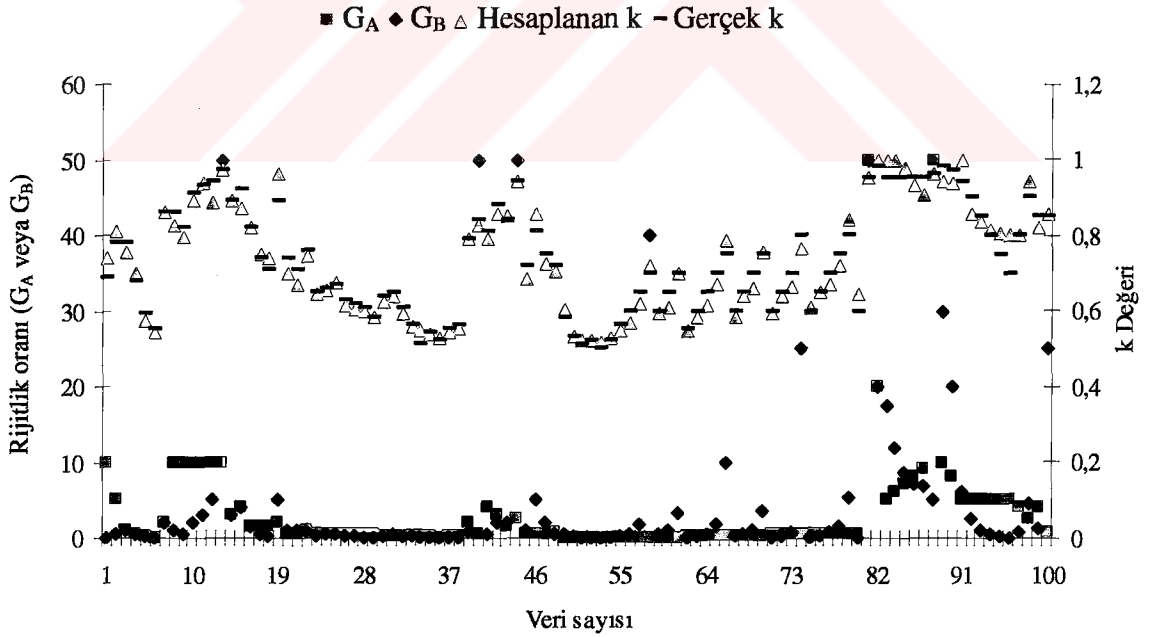
(a) Ötelenme önlenmemiş

(a) Ötelenme önlenmiş

Şekil 2. Yanal ötelenmesi (a) önlenmemiş (b) önlenmiş sistemler için burkulma boyu katsayıları



Şekil 3. Yanal ötelenmesi önlenmemiş sistemler için gerçek ve hesaplanan burkulma boyu katsayısı



Şekil 4. Yanal ötelenmesi önlenmiş sistemler için gerçek ve hesaplanan burkulma boyu katsayısı

2.1.1.2. Yanal Burkulma

Her iki standartta da iki adet eğilme emniyet gerilmesi bulunmakta olup istenilenin kullanılması serbest bırakılmaktadır. Bu çalışmada her iki emniyet gerilmesinden büyük olanına göre işlem yapılmaktadır (Odabaşı, 1997). Aşağıda verilen formüller I ve U profilleri için kullanılmaktadır.

Birinci olarak, eğer basınç başlığı dolu ve yaklaşık olarak dikdörtgen enkesite sahip ve çekme başlığı enkesitinden daha büyük ise eğilme emniyet gerilmesi, σ_B ; s çubuğun basınç başlığında dönmeye ve yanal deplasmana karşı mesnetleri arasındaki mesafeyi, d iki başlığın dış yüzeyleri arasındaki mesafeyi, A_b basınç başlığının enkesit alanını, C_b uç moment oranını göstermek üzere

$$\sigma_B = \frac{840000C_b}{sd/A_b} \sigma_a \leq 0,6\sigma_a \quad (\text{TS 648'de}) \quad (12)$$

$$\sigma_B = \frac{12 \times 10^3 C_b}{sd/A_b} \sigma_a \leq 0,6\sigma_a \quad (\text{AISC-ASD'de}) \quad (13)$$

bağıntılarıyla belirlenmektedir.

İkinci olarak, kgf ve cm birimlerinin kullanıldığı TS 648 göre eğilme emniyet gerilmesi; i_{by} , basınç başlığı ve gövdenin basınç bölgesinin 1/3'ünün gövde eksenine göre atalet yarıçapını göstermek üzere

$$\sqrt{\frac{6000000C_b}{\sigma_a}} \leq \frac{s}{i_{by}} \leq \sqrt{\frac{30000000C_b}{\sigma_a}} \text{ ise } \sigma_B = \left[\frac{2}{3} - \frac{\sigma_a (s/i_{by})^2}{90000000C_b} \right] \sigma_a \leq 0,6\sigma_a \quad (14)$$

$$\frac{s}{i_{by}} \geq \sqrt{\frac{30000000C_b}{\sigma_a}} \text{ ise } \sigma_B = \frac{10000000C_b}{(s/i_{by})^2} \sigma_a \leq 0,6\sigma_a \quad (15)$$

bağıntılarıyla belirlenmektedir. Bağntı (14)'deki $\sqrt{\frac{6000000C_b}{\sigma_a}} \geq \frac{s}{i_{by}}$ (yanal burkulmanın olmadığı durumlarda) şartı TS 648'de olmamasına rağmen, bu çalışmada yapı

elemanlarının yanal burkulma yapıp yapmadığını belirlemek için kullanılmıştır. AISC-ASD'ye göre de eğilme emniyet gerilmesi, birim olarak kip ve in kullanılmak üzere

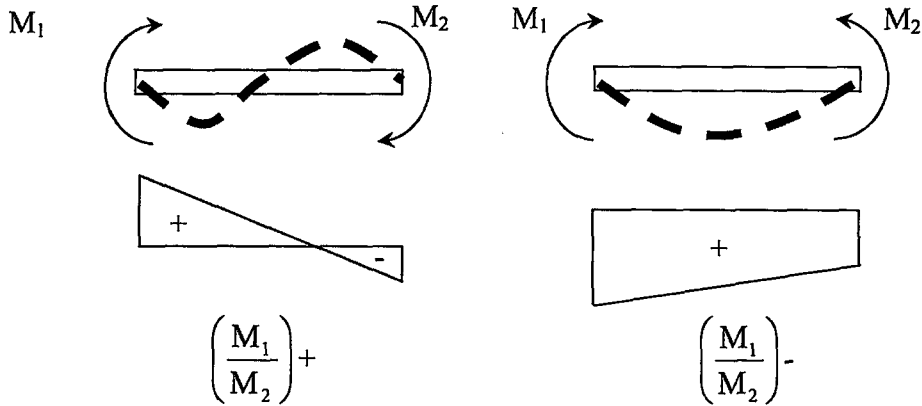
$$\sqrt{\frac{102 \times 10^3 C_b}{\sigma_a}} \leq \frac{s}{i_{by}} \leq \sqrt{\frac{510 \times 10^3 C_b}{\sigma_a}} \text{ ise } \sigma_B = \left[\frac{2}{3} - \frac{\sigma_a (s/i_{by})^2}{1530 \times 10^3 C_b} \right] \sigma_a \leq 0,6\sigma_a \quad (16)$$

$$\frac{s}{i_{by}} \geq \sqrt{\frac{510 \times 10^3 C_b}{\sigma_a}} \text{ ise } \sigma_B = \frac{170 \times 10^3 C_b}{(s/i_{by})^2} \sigma_a \leq 0,6\sigma_a \quad (17)$$

bağıntılarıyla belirlenmektedir. Yukardaki bağıntılarda C_b oranı; M_1 , yapı elemanlarının uç momentlerinin küçüğünü, M_2 yapı elemanlarının uç momentlerinin büyüğünü göstermek üzere

$$C_b = 1,75 + 1,05 \left(\frac{M_1}{M_2} \right) + 0,3 \left(\frac{M_1}{M_2} \right)^2 \leq 2,3 \quad (18)$$

bağıntısı ile verilmektedir. Şekil 5'de C_b için uç moment oranlarının işaretleri gösterilmektedir. Moment oranı M_1/M_2 ; M_1 ve M_2 aynı işarete sahip ise (iki yönlü eğilme) pozitif, ayrı işarete sahip ise (tek yönlü eğilme) negatiftir. Eğer açıklıkta iki uç momentinden daha büyük değere sahip bir moment var ise C_b 'nin değeri 1,0 olmaktadır.



Şekil 5. Uç moment oranlarının, C_b , işaretleri

2.1.1.3. Kritik Burkulma Gerilmesi

Euler gerilmesi olarak da adlandırılan kritik burkulma gerilmesi, σ_e ; güvenlik faktörü, n , TS 648 ve AISC-ASD'de sırasıyla 2,5 ve 23/12 olmak üzere

$$\sigma_e = \frac{\pi^2 E}{n \lambda^2} \quad (19)$$

bağıntısı ile verilmektedir.

2.1.2. Eksenel Basınç ve Eğilmeye Çalışan Yapı Elemanlarında Gerilme Denetimi

Merkezi basınç kuvveti ile beraber M_1 ve M_2 uç eğilme momentlerine ($M_1 < M_2$) maruz elemanlarda, burkulmalı ve burkulmasız gerilme denetimi sırasıyla; σ_{eb} hesap edilen eksenel basınç gerilmesini, σ_{bx} ve σ_{by} hesap edilen eğilmeden dolayı oluşan gerilmeyi, σ_{bem} eksenel basınç emniyet gerilmesini, σ_{Bx} ve σ_{By} eğilme etkisindeki basınç emniyet gerilmelerini σ_{ex} ve σ_{ey} kritik burkulma gerilmelerini, C_m mesnetlenmeyle birlikte açıklık ve uç momentlerini dikkate alan bir katsayıyı göstermek üzere

$$\frac{\sigma_{eb}}{\sigma_{bem}} + \frac{C_{mx} \sigma_{bx}}{\left(1,0 - \frac{\sigma_{eb}}{\sigma_{ex}}\right) \sigma_{Bx}} + \frac{C_{my} \sigma_{by}}{\left(1,0 - \frac{\sigma_{eb}}{\sigma_{ey}}\right) \sigma_{By}} \leq 1,0 \quad (20)$$

$$\frac{\sigma_{eb}}{0,6\sigma_a} + \frac{\sigma_{bx}}{\sigma_{Bx}} + \frac{\sigma_{by}}{\sigma_{By}} \leq 1,0 \quad (21)$$

bağıntılarıyla yapılmaktadır. Eğer $\frac{\sigma_{eb}}{\sigma_{bem}} \leq 0,15$ ise bu bağıntılar yerine

$$\frac{\sigma_{eb}}{\sigma_{bem}} + \frac{\sigma_{bx}}{\sigma_{Bx}} + \frac{\sigma_{by}}{\sigma_{By}} \leq 1,0 \quad (22)$$

bağıntısı kullanılabilir. C_m katsayısı yanal ötelenmenin mümkün olduğu çerçevelerde 0,85 olmakla beraber düğüm noktalarının ötelenmesine müsaade edilmeyen çerçevelerde ve üzerinde eğilmenin düşünüldüğü düzlemde yük olmayan elemanlarda

$$C_m = 0,6 - 0,4 \frac{M_1}{M_2} \geq 0,4 \quad (23)$$

bağıntısıyla, düğüm noktalarının ötelenmesine müsaade edilmeyen çerçevelerde ve üzerine eğilmenin düşünüldüğü düzlemde yük olan çubuklarda ise

$$C_m = 1 + \psi \frac{\sigma_{eb}}{\sigma_e} \quad (24)$$

bağıntısıyla ifade edilmektedir. Bu bağıntılarda ψ ; δ_0 eğilme yükünden meydana gelecek maksimum deplasmanı, EI elemanın rijitliğini, M_0 açıklıktaki maksimum momenti, s_e çubuk boyunu göstermek üzere

$$\psi = \frac{\pi^2 \delta_0 EI}{M_0 s_e^2} - 1 \quad (25)$$

bağıntısıyla verilmektedir. Bu bağıntılar TS 648 ve AISC-ASD'de benzerdir.

2.1.3. Eksenel Çekme ve Eğilmeye Çalışan Yapı Elemanlarında Gerilme Denetimi

Çekme ve eğilmeye çalışan yapı elemanlarında gerilme denetimi

$$\frac{\sigma_{eb}}{0,6\sigma_a} + \frac{\sigma_{bx}}{\sigma_{Bx}} + \frac{\sigma_{by}}{\sigma_{By}} \leq 1,0 \quad (26)$$

bağıntısıyla yapılabilir.

2.1.4. Kayma Gerilmesi Denetimi

Kayma gerilmesi denetimi, σ_k kayma gerilmesini göstermek üzere

$$\frac{\sigma_k}{0,4\sigma_a} \leq 1,0 \quad (27)$$

ifadesiyle yapılabilmektedir.

2.1.5. Makaslama Denetimi

Makaslama denetimi, σ_v makaslama gerilmesini göstermek üzere

$$\sigma_v = \sqrt{(\sigma_{bx}^2 + \sigma_{by}^2 - \sigma_{bx}\sigma_{by} + 3\sigma_k^2)} \quad (28)$$

$$\frac{\sigma_v}{0,75\sigma_a} \leq 1 \quad (\text{EY yüklemesi için}) \quad (29a)$$

$$\frac{\sigma_v}{0,80\sigma_a} \leq 1 \quad (\text{EİY yüklemesi için}) \quad (29b)$$

bağıntılarıyla yapılabilmekte olup TS 648 için dikkate alınmaktadır.

2.2. Genetik Algoritma ve Optimizasyon

GA en iyinin hayatta kaldığı ve doğal seçilimin uygulandığı süreci taklit eden araştırma algoritmasıdır. Ayrıca GA, operatörleri kullanarak belli sayıdaki potansiyel çözümü keyfi üreten problemlere iyi çözümler bulmaya çalışan ve parametre kodlama esasına dayalı bir arama tekniğidir (Goldberg, 1989). GA'daki temel düşünce, yüksek uygunluktaki çözümlerin çoğaltılmasından ve yeni çözümleri üretmek için çoğaltılan

çözümlere genetik operatörlerin uygulanıp daha kapsamlı ve etkin çözümlerin bulunmasından ibarettir (Rasheed, 1998).

GA terminolojisinde her bir çözüme (tasarım) “birey” veya “kromozom”, çözümlerin oluşturduğu topluluğa da “popülasyon” denilmektedir. Meydana gelen yeni bireyler ise “çocuk” olarak adlandırılmaktadır. Bireylerdeki değişkenler, temsili karakterlerden (bit) yani “gen”lerden oluşabileceği gibi genlerin oluşturacağı alt dizilerden de meydana gelebilmektedir. Değişkenler tasarım sürecinde üstlendikleri rolden ötürü “tasarım değişkenleri” veya “karar değişkenleri” olarak da ifade edilmektedir. Bu çalışma içinde çözümler için birey, değişkenler için de tasarım değişkeni ifadesi kullanılmaktadır. GA, problemin çözümlerini oluşturan değişkenleri içeren dizilerden meydana gelen toplulukta (popülasyon) araştırma yaparken; tasarım değişkenleri de problem için dikkate alınacak ayrı değerlerin mevcut olduğu bilgi kümesinde (veri tabanı) karşılıklarını aramaktadır. Araştırma uzayını oluşturan dizilerden araştırmanın başlangıcında keyfi şekilde seçim yapıp ilk topluluk oluşturulmaktadır (Cao, 1996). Aday çözümlerin yani bireylerin uygunlukları, sınırlayıcılara karşı gösterdikleri performans ve amaç fonksiyonu arasındaki ilişkiye bağlı olarak belirlenmektedir. Her bireyin uygunluğu, sınırlayıcılarının sağlanıp sağlanmamasına bağlı olarak amaç fonksiyonunda ihlallerin performanslarını karakterize eden bir ceza fonksiyonu aracılığıyla indirgenmektedir. Böylelikle bireyler için, problemlere çözüm olma kapasitelerine sayısal olarak karşılık gelen skaler bir değer, yani uygunluk bulunmaktadır (Rasheed, 1998). GA yapısı gereği, popülasyon ortalamasının üstünde uyum gücü gösteremeyen kötü bireyleri yani uygun olmayan çözümleri, operatörleri sayesinde elemekte ve diğer uyumu yüksek dizileri ise zamanın ilerlemesiyle çoğaltmaktadır. Bu çoğalma, genetik işlemler aracılığı ile gerçekleşmekte ve sonucunda ebeveynlerden daha üstün özellikler taşıyan bireyler ortaya çıkmaktadır. Çözüm kalitesinin nesilden nesile artması, başarısız olan bireylerin kopyalanma şanslarının azaltılmasıyla birlikte kötüye gidişi zorlaştırmaktadır. GA'nın yapısı kötüye gidişi engellemekle kalmamakta, zaman içinde hızlı bir iyiye gidişe de olanak sağlayabilmektedir. Bu işlemler, bir döngü içerisinde, durdurma kriteri sağlanana kadar devam ettirilmektedir. Bir çok alanda uygulamaları olan GA'nın işleme adımları şöyle açıklanabilir (Emel ve Taşkın, 2002):

- ◆ Arama uzayındaki tüm mümkün çözümler (tasarım) dizi olarak kodlanır ve genellikle bu dizilerden oluşturulan keyfi bir çözüm kümesi seçilip başlangıç popülasyonu olarak kabul edilir.

- ◆ Her bir birey için dizilerin çözüm kalitesini gösteren bir uygunluk değeri hesaplanır.
- ◆ Bir grup birey belirli bir olasılık değerine göre seçilip kopyalama işlemi gerçekleştirilir.
- ◆ Yeni bireyler (çocuk) uygunluk değerleri hesaplanarak, çaprazlama ve mutasyon işlemlerine tabi tutulur.
- ◆ Önceden belirlenen şartlar sağlanana kadar yukarıdaki işlemler devam ettirilir ve nihayetinde sona erdirilir.
- ◆ Amaç fonksiyonuna göre en uygun olan birey seçilir.

2.2.1. Doğa ile Genetik Algoritmanın Benzerlikleri

GA, bir çok bakımdan doğa ile benzerlik göstermekte olup GA ve doğada kullanılan terimler arasındaki ilişki Tablo 1’de verilmektedir.

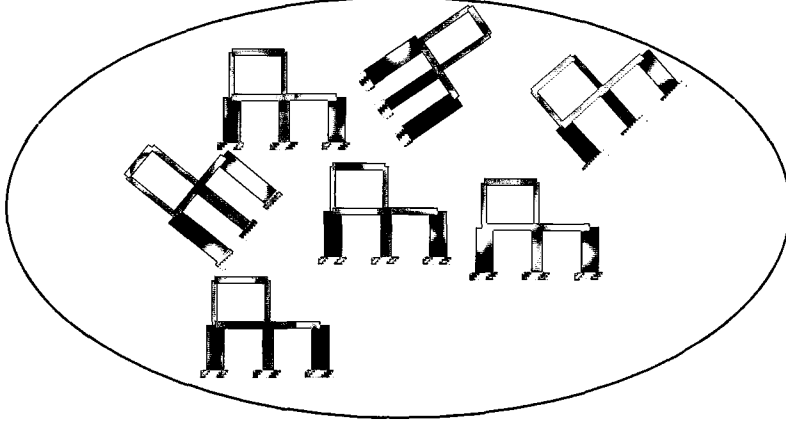
Tablo 1. Genetik algoritma ile doğanın karşılaştırılması

Genetik algoritma	Doğa
Optimizasyon problemi	Çevre
Uygulanabilir çözümler	Çevrede yaşayan bireyler
Çözümün kalitesi	Bireylerin çevrelerine adaptasyon dereceleri
Uygulanabilir çözümler kümesi	Bir organizma popülasyonu
Stokastik operatörler	Kopyalama, çaprazlama, mutasyon
Uygulanabilir çözümler kümesindeki stokastik operatörlerin ardışık uygulanması	Çevre koşullarına uyumun sağlanabilmesi için popülasyonun evrimi

2.2.2. Doğal Seçim ve Genetik Algoritma

GA, en genel halde, keyfi şekilde seçilmiş bireylerden oluşan başlangıç popülasyonunun oluşturulması ile başlamaktadır. Bir toplulukta “n” tane birey varsa o popülasyonun büyüklüğü “n”dir (Kılıç, 1997). Daha sonra evrim süreci GA’daki yerini almaktadır. Bu süreç boyunca bireyler uygunluklarına göre yarışmaktadırlar. Hayatta kalan ve eşlerini tavlayan bireyler çocuk yapma şansına sahip olmakta ve diğerlerinin çocuk yapma şansı azalmaktadır. Bu mücadele en iyinin hayatta kalmasını amaçlamaktadır. Daha iyi bireylerin seçiminden dolayı yeni nesiller, önceki nesillerin iyi bireylerinin

karakteristiklerinin büyük bir kısmını içermektedirler. Bundan dolayı yeni nesil önceki nesilden daha iyi ve ortalama uygunluğu daha yüksektir. Bu stratejinin tekrarıyla önceki nesillerdeki iyi karakterler popülasyonun içine nüfus etmektedirler (Kılıç, 1997).



Şekil 6. Birey topluluğu (Popülasyon)

2.2.3. Genetik Algoritmada Yapılan Temel Kabüller

Geliştirilen GA'da yapılan bazı kabüller aşağıda sıralanmaktadır.

- ◆ Birey uzunluğu yani bireyi oluşturan karakter sayısı sabit olabildiği gibi değişken de olabilmektedir. Yapısal optimizasyonda tasarım değişkenleri sayısı, her tasarım için sabit olduğundan düşünülen GA'da birey uzunluğu sabit olmaktadır.
- ◆ Popülasyon büyüklüğü yani popülasyondaki birey sayısı sabit olabildiği gibi değişken de olabilmektedir. Bu çalışma için popülasyon büyüklüğü sabit olarak alınmaktadır.
- ◆ Popülasyon büyüklüğünün niteliğine karar verilmelidir. Popülasyondaki tüm bireylerin yeni birey oluşumuna katılımının sağlanması, bireylerin eşlenmesi ve çaprazlama operatörünün çiftler arasında uygulanması amacıyla popülasyon büyüklüğü bu çalışma için çift değer olarak alınmaktadır.

2.3. Genetik Algoritma ile Yapıların Optimum Tasarımı

2.3.1. Genetik Algoritmada Çözümlerin Kodlanması

Bir problemin çözümü için bir GA geliştirmenin ilk adımı, tüm çözümlerin aynı boyutlara sahip karakterler (bit) dizisi biçiminde gösterilmesidir. Dizilerden her biri, problemin olası araştırma uzayındaki keyfi bir çözümünü simgelemekte olup kodlanmaları, probleme özgü bilgilerin GA'nın kullanacağı şekle çevrilmesine olanak tanımaktadır. GA, kodlanmış çözümleri ifade eden genotip ve gerçek çözümleri gösteren fenotip olmak üzere, iki ayrı uzay kullanmaktadır. Tablo 2'de bir popülasyondaki bireylerin ikilik, dörtlük, değer ve reel kodlamada genotip ve fenotipi verilmektedir.

Tablo 2. Bireyin fenotip ve genotip gösterimi

Sistem adı	Fenotip	Birey no	Genotip
İkilik kodlama	22 51 241 4	1	00010101 00111010 11110000 00000011
	18 50 166 23	2	00010001 00111011 10100101 00010110
	37 48 121 34	3	00100100 10111001 10100101 00010110

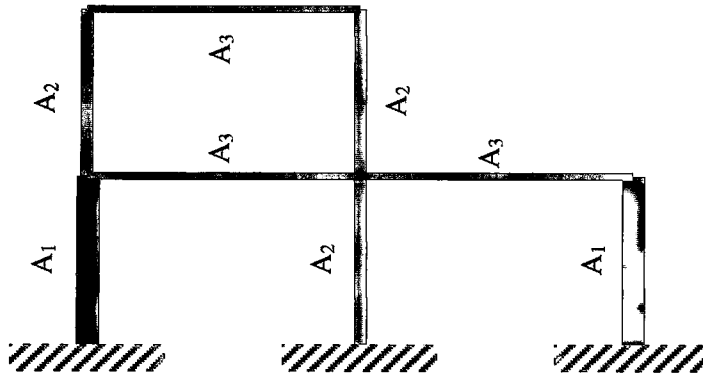
	198 89 107 8	20	11000101 01011000 01101010 00000111
Dörtlük kodlama	22 51 241 4	1	0111 0303 3300 0003
	18 50 166 23	2	0100 0301 2211 0112
	37 48 121 34	3	0210 0300 1320 0201

	198 89 107 8	20	3011 1120 1222 0013
Değer kodlaması	22 51 241 4	1	22 51 241 4
	18 50 166 23	2	18 50 166 23
	37 48 121 34	3	37 48 121 34

	198 89 107 8	20	198 89-107 8
Reel kodlama		1	41,87 85,81 631,61 22,9
		2	36,26 85,16 345,81 41,87
		3	65,52 81,29 216,13 58,84
	
		20	445,16 145,81 187,74 28,45

Temsillerde her bir çözüm için tasarım değişkeni kodlanmakta ve daha sonra bu kodlar sıralanarak çözümlerin genotipleri oluşturulmaktadır. Çözümlerdeki tasarım değişkenlerinin bilgi sıralamasında hangi sayıya karşılık geldiğini belirlemek için temsillerdeki kodlar tamsayıya çevrilmiştir. Böylece bireylerin tasarım değişkenlerinin tamsayılarla ifade edildiği fenotipler oluşturulmuş olmaktadır. En yaygın kodlama türleri ikilik ve gerçek değer kodlaması olup, değer kodlaması gibi yukardaki tabloda görülen başka tip kodlamalar da kullanılmaktadır. Reel kodlamada bireylerin genotip gösterimleri tasarım değişkenlerinin bizzat kendileriyle, değer kodlamasında ise tamsayılarla ifade edilmektedir. Tasarım değişkeni olarak profil alanları düşünüldüğünden reel kodlamadaki değerler alan cinsinden olmaktadır. Bu kodlama türlerinin dışında tüm sayılara göre kodlama yapmak mümkündür.

Simetrik bir geometriye sahip olmayan Şekil 7'deki 8 elemanlı bir düzlem çerçeve üzerinde tasarım değişkeni ve bireyin adreslenmesinin ne anlama geldiği incelenmektedir. Sistemde alt kat kenar kolonları birinci çeşit, üst kat kolonları ile alt kat orta kolonu ikinci çeşit ve kirişler de üçüncü çeşit olmak üzere, çerçevenin elemanlarının 3 gruptan oluştuğu düşünülmekte olup elemanların sadece eksenleri doğrultusunda hareket etme yeteneklerine sahip oldukları varsayılmaktadır. Dolayısı ile tasarım değişkeni enkesit alanı olmaktadır. Tablo 3'de enkesit alan kodlamalarının veri tabanındaki sayı ve değer karşılığı verilmektedir. Kodlama için 4'lük sistem kullanılmaktadır. Yani tasarım değişkenleri 2 bitlik değerlerle ifade edilmektedir.



Şekil 7. Sekiz elemanlı çelik düzlem çerçeve

Tablo 3. Enkesit alan kodlamalarının veri tabanındaki sayı ve değer karşılığı

Dizi	Bilgi sırası	Enkesit alanı (cm ²)	Dizi	Bilgi sırası	Enkesit alanı (cm ²)
00	1	540,6	20	9	418,1
01	2	469,7	21	10	364,5
02	3	424,5	22	11	1238,7
03	4	374,2	23	12	1122,6
10	5	621,9	30	13	1006,4
11	6	565,2	31	14	903,2
12	7	508,4	32	15	825,8
13	8	462,6	33	16	748,4

Problem, yukarıda verilen tablodaki enkesit alanlarından oluşan veri tabanında, farklı yük ve yükleme durumları altında sistemin ağırlığının minimum olmasının amaç fonksiyonu olduğu ve oluşan gerilmelerin emniyet gerilmesini aşmamasının sınırlayıcı olduğu düzlem çerçevenin optimum boyutlandırılmasıdır. Bu problem için birey ve tasarım değişkeni gösterimi Tablo 4'de verilmektedir.

Tablo 4. Birey ve tasarım değişkenleri arasındaki ilişki

Tasarım dizisi (Birey)	332101		
Alt dizi	33	21	01
Tasarım değişkeni	A ₁	A ₂	A ₃
Bilgi numarası	16	10	2
Enkesit alanı (cm ²)	784,4	364,5	469,7

2.3.2. Başlangıç Stratejisi

GA'da mutlaka olası çözümlerin kodlandığı çözümler topluluğu (popülasyon) üretilmelidir. Başlangıç çözümleri genelde keyfi şekilde oluşturulmaktadır (Rasheed, 1998). GA'nın verimli çalışması için topluluktaki birey sayısının birey uzunluğuna bağlı olarak yeterli sayıda olması gerekmektedir (Aydın, 2000). Bireylerin uzunluğu da tasarım değişkeni sayısına ve bu tasarım değişkenleri için seçilen değişken kümesindeki değer sayısına bağlı olarak değişmektedir. Genelde popülasyon büyüklüğü 20-100 arasında değişmektedir. Bu değer çok küçük olduğunda, GA, yerel bir optimuma odaklanabilmektedir. Popülasyonun çok büyük olması ise çözüme ulaşma zamanını artırmaktadır (Emel ve Taşkın, 2002).

Dizilerin uzunluğu üç şekilde ayarlanabilmektedir: 1. Tasarım değişkeni sayısını azaltmak, 2. Kodlamalarda (genotip) farklı temsiller kullanmak, 3. Tasarım değişkeni değer kümesindeki bilgi sayısını azaltmak. Yapı sistemlerinde yapı elemanlarının her biri bir tasarım değişkeni olarak girilebileceği gibi elemanların bazıları yük durumları, stabilite ve kesit etkileri bakımından gruplanabilmektedir. Böylece bireyler grup sayısı kadar tasarım değişkeninden oluşmaktadır. Grup sayısı tasarımdaki hassasiyete bağlı olarak azaltılabilir ki bu uzun dizilerin oluşmasını engelleyebilir. Bireylerin, stokastik operatörlerin uygulandığı safhada, genotip formları kullanılmaktadır. Bu formda dizi uzunluğu seçilen kodlama türüne göre değişmektedir. İkilik kodlamada 256 adet bilginin bulunduğu veri tabanından 155. bilgiye ulaşmak için tasarım değişkenini

1 0 0 1 1 0 1 0

$$2^7 \times 1 + 2^4 \times 1 + 2^3 \times 1 + 2^1 \times 1 = 154 \quad 154 + 1 = 155$$

şeklinde 8 bitlik bir dizi ile gösterirken, dörtlük kodlamada

2 1 2 2

$$4^3 \times 2 + 4^2 \times 1 + 4^1 \times 2 + 4^0 \times 2 = 154 \quad 154 + 1 = 155$$

4 bitlik dizi ile ifade edilmesi mümkündür. Öte yandan değer kodlamasında ise değeri 155 olan 1 bitlik bir tamsayı ile direk temsil edilebilmektedir.

Tasarım deęişkenlerinin, karşılıklarını aldıkları bilgi kümesindeki (veri tabanı) deęerlerinin sayısında yapılacak azaltmada, dizi uzunluęu azalacaktır. Örneęin ikilik sistem kullanılarak tasarım deęişkeninin deęerinin 119 olduęu varsayılınsın. 256 bilginin varolduęu bir veri tabanından bu sayıya ulaşılmak istenirse 8 bitlik bir alt dizi ile, 128 bilginin varolduęu bir veri tabanından bu sayıya ulaşılmak istenirse de 7 bitlik bir alt dizi ile tasarım deęişkeni (genotip formunda) ifade edilmektedir. Böylelikle bir tasarım deęişkeninden 1 bitlik uzunluk kazanımı sağlanmaktadır. Öte yandan bireyleri oluşturan her bir tasarım deęişkeninin aynı veri tabanında farklı aralıktaki verileri kullandığı düşünöldüęünde, tasarım deęişkeni uzunlukları ikilik sistemde 3-4 bit'e kadar inebilmektedir. Bu tarz kullanım bazı problemler için programın ilk koşturulmasından itibaren uygulanabileceęi gibi bazılarında da birkaç koşturmadan sonra edinilen gözlem sonrasında yapılabilmektedir. Daha büyük kodlu tabanlar kullandıkça bu tarz için de dizi uzunluęu azalmaktadır.

Birey uzunluęunun yukarıda bahsedildięi gibi birey sayısını etkilemesi, yakınsama oranını da etkilemektedir. İstenen oranda yakınsama sağlanması için daha çok iterasyon gerekmektedir. Örneęin 20 bireyli bir popölasyonla çözüm arandığıında yakınsama olasılıęı %5 ile başlarken 60 bireyli bir popölasyonda % 1,667 ile başlamaktadır. %5'lik yakınsamayı 60 bireylik popölasyonda sağlamak için üç bireyin aynı kodda olması gerekmektedir. Bu bağlamda kullanılacak algoritmada tasarlanacak sisteme baęlı olarak popölasyon büyüklüęünün ayarlanması zaman ve yakınsama açısından önemli olmaktadır. Ayrıca dizi uzunluęunun, belirlenmesinde temel olabilecek kesin bir kriter olmamakla beraber GA ile sonuca ulaşabilmek açısından çok küçük de olması istenmemektedir. Örneęin 2 tasarım deęişkenli bir sistem için veri tabanındaki bilgi sayısının 64 olduęu ve 8'lik kodlamanın kullanıldığı varsayıldığıında bir birey için dizi uzunluęu 4 bit olacaktır. Bu gösterimdeki bir bireyle oluşturulacak topluluktaki çeşitlilik azalacaęından elde edilecek çözümün en ekonomik ve en optimum olma şansı azalmaktadır. Eęer ikilik sistem kullanılırsa dizi uzunluęu 12 bit'e yükselmektedir. Bu uzunluktaki bireylerden oluşturulan topluluktaki çeşit, yani çözüm potansiyeli ilk gösterime göre çok daha fazla olmaktadır. Bundan dolayı kullanılan ilk tarzda araştırma uzayının yeteri kadar iyi taranması için programın çok fazla sayıda koşturulması gerekmektedir. Ama ikinci tarz bir uygulama ile bu yükü tamamen program üstleneceęinden bir iki koşturmada sonuca ulaşılmaktadır.

Dizi uzunluęu ve topluluk içindeki dizi sayısı belirlendikten sonra ilk diziler yani başlangıç popölasyonu belirlenmelidir. Yukarıda da ifade edildięi gibi belirleme işlemi

genelde keyfi yapıldığı gibi kullanıcının önsezisi doğrultusunda da yapılabilmektedir. Böylelikle sonuca ulaşma süresi kısalabilmektedir. Az tasarım değişkenine sahip sistemlerde kullanıcı sezisi belki de programı hiç koşturmadan yeteri kalitede çözümlerin oluşturduğu topluluğu veri olarak oluşturabilir. Bunun yanı sıra tasarım değişkeni sayısı fazla olan sistemlerde program keyfi oluşturulmuş bir toplulukla bir iki defa koşturulduktan sonra da kullanıcı tarafından bu başlangıç topluluğu daha verimli belirlenebilmektedir.

2.4. Yapısal Optimizasyon Problemi

Yapısal optimizasyon problemi, bir amaç fonksiyonu ve sınırlayıcılardan oluşmaktadır. Bundan ötürü amaç fonksiyonun tanımı, sınırlayıcıların tanımı ve nasıl oluşturuldukları hakkında detaylı bilgi alt başlıklar halinde verilmektedir.

2.4.1. Sınırlayıcılar

GA'da sistemin eleman veya grup sayısına göre düzenlenen her tasarım değişkeni kombinezonu, bir bireyi ifade etmektedir. Fakat bunların hepsi sınırlayıcıları sağlayan aday çözümler değildir. Sınırlayıcılar, araştırma uzayında sonuç aranan probleme cevap olabilecek nitelikte aday çözümler bulunmasına ve GA'nın aday çözümler arasından en iyi sonuca ulaşmasına yardımcı olmaktadır. Tütüncü (1995)'nün ifade ettiği gibi sınırlayıcılar "tasarım sınırlayıcıları" ve "fiziksel sınırlayıcılar" şeklinde ayrılabilir. Tasarım sınırlayıcıları sistemin veya elemanların uzunluk, genişlik ve yükseklikleri biçiminde dış özelliklerle ilgili olabilmektedir. Fiziksel sınırlayıcılar ise yapı elemanlarının farklı güçlülere dayanma sınırları yani maksimum gerilmeleri veya sistemin düğüm noktalarının hareketleri şeklinde sıralanabilmektedir. Bu bağlamda basit olarak fiziksel sınırlayıcılar; δ_j j inci noktada hesaplanan yerdeğiştirme miktarını, δ_{uj} j inci nokta için belirlenen yerdeğiştirme miktarı üst sınırını, p yerdeğiştirmesi sınırlandırılmış nokta sayısını, $\sigma_{e,i}$ i inci çubuk için hesaplanan gerilme değerini, $\sigma_{em,i}$ i inci çubuğun taşıyabileceği maksimum gerilme miktarını ve nm çerçeve sistemdeki toplam eleman sayısını göstermek üzere

$$\delta_j \leq \delta_{u,j} \quad j = 1, 2, 3, \dots, p \quad (30)$$

$$\sigma_{e,i} \leq \sigma_{em,i} \quad i = 1, 2, 3, \dots, nm \quad (31)$$

bağıntıları ile verilmektedir.

2.4.2. İhlaller ve Ceza Fonksiyonu

GA, daha önce de belirtildiği gibi, sınırlayıcısız bir yöntemdir. Dolayısıyla da optimum tasarım probleminin sınırlayıcısız bir probleme dönüştürülmesi gerekmektedir. Sınırlayıcıların boyutsuz değerlere indirgenmesi, ihlallerin bir fonksiyon yardımıyla amaç fonksiyonuna eklenmesi ile yerine getirilmiş olmaktadır. Böylece, sınırlayıcıların ihlal edilmesi ölçüsünde amaç fonksiyonunun değeri artırılmaktadır. Amaç fonksiyonunun değerinin artması, tasarımı, uygun tasarım olma niteliğinden uzaklaştıracağı için bir çeşit cezalandırma biçimi olmaktadır. Bundan dolayı ihlalleri içeren fonksiyona “ceza fonksiyonu” denilmektedir. Ceza fonksiyonunun amaç fonksiyonuna eklenmesiyle de problem sınırlayıcısız bir probleme dönüştürülmüş olmaktadır. Sınırlayıcılara bağlı olarak ceza fonksiyonunun belirlenebilmesi için, sınırlayıcıların normalize edilmiş formları aşağıdaki gibi gösterilebilmektedir. Sınırlayıcı olarak da; gerilme, yerdeğiştirme, stabilite, sehim ve geometrik şartlar dikkate alınmaktadır.

Yerdeğiştirme ihlalleri; δ_j j noktasında hesaplanan yerdeğiştirmeyi, $\delta_{u,j}$ j noktası için izin verilen yerdeğiştirme miktarını, p ise yerdeğiştirmesi sınırlandırılmış nokta sayısını göstermek üzere

$$g_j = \frac{\delta_j}{\delta_{u,j}} - 1 \leq 0 \quad j = 1, 2, 3, \dots, p \quad (32)$$

bağıntısıyla verilmektedir.

Kirişlerde sehim ihlalleri; δ_k k noktasında hesaplanan yerdeğiştirmeyi, $\delta_{u,k}$ k noktası için izin verilen sehim miktarını, r ise sehimi sınırlandırılmış eleman sayısını göstermek üzere

$$g_k = \frac{\delta_k}{\delta_{u,k}} - 1 \leq 0 \quad k = 1, 2, 3, \dots, r \quad (33)$$

bağıntısıyla verilmektedir.

Gerilme ihlalleri ise aksenal basınç ve eğilmeye çalışan çubuklar için σ_{eb} hesap edilen aksenal basınç gerilmesini, σ_{bx} ve σ_{by} hesap edilen eğilmeden gelen gerilmeyi, σ_{bem} aksenal basınç emniyet gerilmesini, σ_{Bx} ve σ_{By} eğilme etkisindeki basınç emniyet gerilmesini σ_{ex} ve σ_{ey} kritik burkulma gerilmesini, C_m mesnetlenmeyi, açıklık ve uç momentlerini dikkate alan bir katsayıyı, n hem aksenal basınca hem de eğilmeye çalışan çubuk sayısını göstermek üzere

$$g_i = \frac{\sigma_{eb,i}}{\sigma_{bem,i}} + \frac{C_{mx,i} \sigma_{bx,i}}{\left(1,0 - \frac{\sigma_{eb,i}}{\sigma_{ex,i}}\right) \sigma_{Bx,i}} + \frac{C_{my,i} \sigma_{by,i}}{\left(1,0 - \frac{\sigma_{eb,i}}{\sigma_{ey,i}}\right) \sigma_{By,i}} - 1 \leq 0 \quad i= 1, 2, 3, \dots, n \quad (34)$$

$$g_m = \frac{\sigma_{eb,m}}{0,6\sigma_a} + \frac{\sigma_{bx,m}}{\sigma_{Bx,m}} + \frac{\sigma_{by,m}}{\sigma_{By,m}} - 1 \leq 0 \quad m= 1, 2, 3, \dots, n \quad (35)$$

bağıntılarıyla ifade edilmektedir. Eğer $\frac{\sigma_{eb,i}}{\sigma_{bem,i}} \leq 0,15$ ise yukarıdaki bağıntılar yerine

$$g_i = \frac{\sigma_{eb,i}}{\sigma_{bem,i}} + \frac{\sigma_{bx,i}}{\sigma_{Bx,i}} + \frac{\sigma_{by,i}}{\sigma_{By,i}} - 1 \leq 0 \quad i= 1, 2, 3, \dots, n \quad (36)$$

ifadesi yazılabilmektedir.

Çekmeye ve eğilmeye çalışan yapı elemanlarında gerilme ihlali

$$g_p = \frac{\sigma_{eb,p}}{0,6\sigma_a} + \frac{\sigma_{bx,p}}{\sigma_{Bx,p}} + \frac{\sigma_{by,p}}{\sigma_{By,p}} - 1 \leq 0 \quad p= 1, 2, 3, \dots, n \quad (37)$$

ifadesi ile yazılabilmektedir. Tüm elemanlarda kesme ihlali

$$g_u = \frac{\sigma_{k,u}}{0,4\sigma_a} - 1 \leq 0 \quad u= 1, 2, 3, \dots, n \quad (38)$$

şeklinde ifade edilebilmektedir. Burada $\sigma_{k,u}$ u uncu çubuğun kayma gerilmesidir.

Tüm eğilmeye çalışan yapı elemanlarında makaslamadan dolayı ihlal

$$\frac{\sigma_{v,f}}{0,75\sigma_a} - 1 \leq 0 \text{ (EY yüklemesi için)} \quad (39a)$$

$$\frac{\sigma_{v,f}}{0,80\sigma_a} - 1 \leq 0 \text{ (EİY yüklemesi için)} \quad (39b)$$

şeklinde ifade edilebilmektedir. Burada $\sigma_{v,f}$ f inci çubuğun makaslama gerilmesidir.

Ayrıca kolonlarda enkesit alanlarının kot arttıkça azalmasını sağlamak için geometrik sınırlayıcılar kullanılmaktadır. Bu sınırlayıcıların gerilme, yerdeğiştirme ve stabilite sınırlayıcıları gibi doğrudan formülasyonu yoktur. Bu çalışma için başlangıçta bireyler bu kriteri sağlayacak şekilde oluşturulmakta, ilerleyen nesillerde ise bu şartı sağlamayan elemanların ceza katsayıları artırılmaktadır. Böylelikle bir üst kolonun enkesit alanının bir alt kolonunkinden küçük ya da ona eşit olması hedeflenmektedir.

Normalleştirilmiş sınırlayıcılar yani ihlaller yukarıda verilen bağıntılarla ifade edildikten sonra problemi sınırlayıcısız bir probleme dönüştürmek için gerekli olan ceza fonksiyonu, C, hesaplanmaktadır. Bu fonksiyon; m sınırlayıcıların toplam sayısını, c_i her bir sınırlayıcının ihlal edilme değerini göstermek üzere

$$C = \sum_{i=1}^m c_i \quad (40)$$

bağıntısı ile verilmektedir. Burada sınırlayıcıların ihlal edilme katsayısı, c_i

$$c_i = g_i(x), \quad g_i(x) > 0 \text{ için} \quad (41)$$

$$c_i = 0, \quad g_i(x) < 0 \text{ için} \quad (42)$$

bağıntıları ile belirlenmektedir.

Ceza katsayısı P; ceza fonksiyonunun, sınırlayıcıların ve sistemin ağırlığının ne derece etkili olacağına belirlenmesinde kullanılmaktadır. Sınırlayıcıların ihlal edilmesinin daha önemli olduğu durumlarda P sayısı artırılarak sınırlayıcıların kesinlikle ihlal edilmediği bir çözüm kümesinde minimum ağırlıklı çözümlerin aranması sürdürülmekte veya çalışmada minimum ağırlık ön planda ise P sayısı azaltılarak ağırlığı çok az olan çözüm kümesinde sınırlayıcıları ihlal etmeyen çözümler aranabilmektedir (Aydın, 2000). Bu açıdan bakıldığında, kısa sürede iyi çözümlere ulaşılabilmesi için P sayısının hassasiyetle tespiti daha da önem kazanmaktadır (Ayvaz ve Aydın, 2000).

2.4.3. Amaç Fonksiyonu

Yapısal bir problem için araştırma uzayında tüm sınırlayıcıları sağlayan birçok aday çözüm bulunmaktadır. Aday çözümler arasından seçim yapmak için sınırlayıcılardan farklı bir kriter olan “amaç fonksiyonu”na ihtiyaç duyulmaktadır. Amaç fonksiyonun en iyi çözümü belirlemesi, aday çözümleri bir formülasyona bağlı olarak kıyaslaması ile mümkündür. Sonuca ulaşmak için yetersiz bir formülasyon yerel veya istenmeyen çözümlere yönlendirebilir. Yapı sistemleri için amaç fonksiyonu genelde yapı maliyeti olarak düşünülmektedir. Yapı maliyetini etkileyen, yapının bakımı ve birleşim noktalarının teşkili gibi bir çok parametre vardır. Yapının tümünün ve yapıyı oluşturan elemanların her birinin özelliklerine bağlı bir bağıntı olarak W amaç fonksiyonu; p_m malzeme özelliklerini, p_c birleşim noktalarının özellikleri, p_s yapısal karakteristikleri göstermek üzere

$$W=f(p_m, p_c, p_s) \quad (43)$$

bağıntısı ile verilmektedir. Bu amaç fonksiyonu ile genel anlamda bir yapısal optimum tasarım problemi; g_1, g_2, \dots, g_n sınırlayıcı fonksiyonlarını göstermek üzere

$$\text{minimum } W = f(p_m, p_c, p_s) \quad g_1, g_2, \dots, g_n \geq 0 \quad (44)$$

bağıntısı ile ifade edilebilmektedir (Aydın, 2000). Bu parametreleri içerecek yukarıdaki gibi bir formülü kurmak kolay olmamaktadır. Dolayısıyla amaç fonksiyonu olarak yapı maliyeti içinde en etkin olduğu düşünülen yapı ağırlığı dikkate alınmaktadır. Bu çalışmada amaç fonksiyonunda tek terim olarak sadece minimum ağırlık alınmaktadır. Bu durumda

amaç fonksiyonu, W ; ρ_i i inci çubuğun yoğunluğunu, L_i i inci çubuğun boyunu, A_i i inci çubuğun enkesit alanını, nm sistemdeki toplam eleman sayısını göstermek üzere

$$\min W = \sum_{i=1}^{nm} \rho_i L_i A_i \quad (45)$$

şeklinde ifade edilebilmektedir. Gruplandırma yapıldığında, amaç fonksiyonun formülasyonu; A_k k inci gruba ait elemanların enkesit alanlarını, ng problemdeki toplam grup sayısını göstermek üzere

$$\min W = \sum_{k=1}^{ng} A_k \sum_{i=1}^{nm} \rho_i L_i \quad (46)$$

şeklini almaktadır. Bağntı (45) ile verilen amaç fonksiyonuna aşağıdaki gibi bir ilaveyle, sınırlayıcısız probleme ait cezalandırılmış amaç fonksiyonu, $\Phi(x)$, PC ceza fonksiyonunu göstermek üzere

$$\Phi(x) = W(x)[1 + P.C] \quad (47)$$

şeklinde elde edilmektedir.

Yapısal optimizasyon problemlerinde, amaç fonksiyonu olarak yapı ağırlığı düşünüldüğünde popülasyondaki n adet birey yani tasarım, ağırlıklarına bağlı olarak uygunluk değeri almaktadır. Eğer sınırlayıcılar tam olarak sağlanmış ise en düşük ağırlığa sahip olan tasarım en yüksek uygunluğa sahip olmaktadır. Böylelikle, tüm istenen kriterleri sağlayan iyi bireyler yani düşük ağırlıklı tasarımlar, genetik döngü sonucunda bulunmuş olmaktadır.

2.4.4. Uygunluk Değerinin Hesaplanması

Çözümü aranan her problem için bir uygunluk fonksiyonu mevcuttur. Verilen belirli bir birey için uygunluk fonksiyonu, o bireyin temsil ettiği çözümün kullanımıyla veya yeteneğiyle orantılı olan sayısal bir uygunluk değeri vermektedir. Bu bilgi, her nesilde

daha uygun çözümlerin seçiminde yol göstermektedir. Bir çözümün uygunluk değeri ne kadar yüksekse, yaşama ve çoğalma şansı o kadar fazladır ve bir sonraki nesilde temsil edilme oranı da o denli yüksektir (Emel ve Taşkın, 2002). Bireylerin uyum derecelerinin, F_i , belirlenmesinde bir kritere gereksinim duyulmaktadır. Bu çalışmada bu kriter; $\Phi_i(x)$ i inci bireyin cezalandırılmış amaç fonksiyonu değerini, $\Phi(x)_{\max}$ ve $\Phi(x)_{\min}$ popülasyon içinde cezalandırılmış amaç fonksiyonunun aldığı en büyük ve en küçük değerleri göstermek üzere

$$F_i = (\Phi(x)_{\max} + \Phi(x)_{\min}) - \Phi_i(x) \quad (48)$$

şeklinde belirlenmektedir. Her bireyin uyum faktörü, $F_{c,i}$, \bar{F} çözüm dizilerinin uyum derecelerinin ortalamasını göstermek üzere

$$F_{c,i} = F_i / \bar{F} \quad (49)$$

bağıntısı ile hesaplanmaktadır. Bağıntı (48) ile tasarımlar için hesaplanan uygunluklar sadece içinde buldukları popülasyon için geçerli olmaktadır. Farklı tasarımların bulunduğu başka bir popülasyonda, aynı tasarım, aynı uygunluğu büyük bir ihtimalle almamaktadır.

2.4.5. Genetik Operatörlerin Uygulanması

Genetik operatörler diğer bir deyişle stokastik operatörler, tasarım sürecine biyolojik bir boyut kazandıran işlemler dizisidir. Ayrıca GA'nın araştırma uzayını taramak için kullandığı araçlarıdır. Bu operatörler daha önce de belirtildiği gibi 1. Kopyalama, 2. Çaprazlama, 3. Mutasyon şeklinde sıralanmaktadır. Bu operatörler tek tek kullanılabilirdiği gibi kombinasyonları da kullanılabilir. Ayrıca GA'nın performansını ifade etmek için kontrol parametreleri adı verilen bir takım parametreler kullanılmaktadır. Optimum kontrol parametrelerini bulmak için bir çok çalışma yapılmıştır. Fakat tüm problemler için genel olarak kullanılacak parametreler bulunamamıştır. Bununla birlikte kontrol parametreleri; popülasyon büyüklüğü, çaprazlama olasılığı, mutasyon olasılığı, seçim stratejisi ve fonksiyon ölçeklemesi olarak sayılabilmektedir (Emel ve Taşkın, 2002).

Popülasyon büyüklüğü kısım 2.3.2’de dikkate alınmış olup fonksiyon ölçeklemesi de kısım 2.4.4’de açıklanmıştır. Geriye kalan parametrelere de ilgili operatörler ifade edilirken değinilmektedir.

2.4.5.1. Seçim Stratejisi ve Kopyalama Operatörü

Uygunluğun hesaplanması adımından sonra mevcut nesilden yeni bir popülasyon yaratılmaktadır. Eski nesli yenilemenin çeşitli stratejileri mevcuttur. Nesilsel stratejide, mevcut popülasyondaki bireyler tamamen yavrular ile yer değiştirmektedir. Popülasyonun en iyi bireyi de yenilediğinden dolayı bir sonraki nesile aktarılamamakta ve bu yüzden bu strateji en uygun (elitist) stratejisiyle beraber kullanılmaktadır. En uygun stratejisinde, popülasyondaki en iyi bireyler hiçbir zaman yenilenmemektedir. Bu iki stratejide, uygunluk değerine bağlı rulet tekerleği seçimi, turnuva seçimi ve sıralama seçimi gibi yöntemler uygulanabilmektedir. Denge durumu stratejisinde ise, her nesilde yalnızca birkaç birey yenilenmektedir. Genellikle, yeni bireyler popülasyona katıldığından en kötü bireyler yenilenmektedir (Emel ve Taşkın, 2002).

Seçim işlemi, bir sonraki nesile yeni bireyler (çocuk) üretmek amacıyla hangi ailelerin kullanılması gerektiğine karar vermektir. Burada aileler; önce aileyi oluşturacak bireylerin kimler olacağı, sonra da hangi bireylerin eş olacağı belirlenerek oluşturulur. Seçim işlemi, doğal seçimdeki en uygunun yaşaması durumuna benzemekte ve amacı gereği uygunluk değeri yüksek bireylere çoğalma fırsatı tanımaktadır. Bu süreç ilk önce bir sonraki nesil için popülasyona katkıda bulunacak bireylerin seçimi ile başlamaktadır. Diğer bir deyişle popülasyondaki kötü bireyler popülasyondan atılmakta, yani öldürülmektedir. Popülasyon büyüklüğünün GA’da sabit tutulması gerektiği varsayılarak (Değişken büyüklükte popülasyona sahip GA da kullanılmaktadır), popülasyonda reel değer olan uygunluk faktörleri, “2” tamsayı değerini geçmeyecek şekilde tamsayılara dönüştürüldükten sonra uygunluk değeri “0” olan bireyler (öldürülen bireyler) yerine uygunluğu “2” olan bireylerden, bir bireyden en fazla iki adet olmak üzere, kopyalama yapılarak, yeni bireylerin ebeveyn adayları oluşturulmaktadır. Tamsayıya dönüştürme $0 \leq f < 0,5$ için “0”, $0,5 \leq f < 1,5$ için “1” ve $1,5 \leq f \leq 2,0$ için de “2” alınarak yapılmaktadır. Böylelikle, iyi olan kalıtsal özellikler gelecek nesile aktarılmış olmaktadır. Bireyleri uygunluk değerlerine göre kopyalama; daha yüksek uygunluk değerine sahip bireylerin, bir sonraki nesildeki bir veya daha fazla yavruya daha yüksek bir olasılıkla

katkıda bulunması anlamına gelmektedir. Oluşan bu ara popülasyona “eşleme havuzu” veya “çaprazlama havuzu” denilmektedir. Daha sonra, bir önceki nesilden elde edilen bu havuzda, yeni bireyleri (çocuk) oluşturacak çiftleri belirleme işlemine geçilmektedir. Popülasyon büyüklüğü çift sayı olarak düşünüldüğünde çiftler keyfi olarak belirlenmekte ve çaprazlama sürecine girilmektedir.

Bu çalışmada, uygunlukların bireylerin hayatta kalıp kalmamalarını nasıl etkilediği Tablo 5’de görülmektedir. Popülasyon büyüklüğü 12 olan toplulukta uygunluk faktörleri tamsayıya dönüştürülmektedir. Sonuçta öldürülecek birey sayısı ve bunların hangi bireyler olacağı aşağıdaki tabloda (5., 7., 10., bireyler) verilmektedir. Öldürülen bireyler yerine kopyalanacak bireyler (1., 6., 8., bireyler), uygunluğu “2” olan birey olmadığı için en iyi üç bireyin uygunluğu “2” yapılarak belirlenmektedir. Tablo’nun sondan ikinci sütununda eşleme havuzuna girecek birey numaraları gösterilmektedir. Son sütunda da birinci sütunda verilen birey numaralarına karşılık gelen eş numaraları verilmektedir.

Tablo 5. Birey uygunluklarının bulunuşu

Birey no	F_i	$F_{c,i}$	Tamsayı değerler	Dönüştürülmüş tamsayı değerler	Hayatta kalan bireyler	Eşleme havuzundaki bireyler	Eşler
1	43,922	1,935	1	2	Evet	1	8
2	25,989	1,145	1	1	Evet	2	10
3	35,069	1,545	1	1	Evet	3	11
4	23,969	1,056	1	1	Evet	4	12
5	11,531	0,508	0	0	Hayır	8	7
6	40,608	1,789	1	2	Evet	6	9
7	8,330	0,367	0	0	Hayır	1	5
8	44,194	1,947	1	2	Evet	8	1
9	27,442	1,209	1	1	Evet	9	6
10	19,929	0,878	0	0	Hayır	6	2
11	36,635	1,614	1	1	Evet	11	3
12	27,760	1,223	1	1	Evet	12	4

2.4.5.2. Çaprazlama Operatörü

Mevcut tasarım değişkeni havuzundaki, diğer bir deyişle, popülasyondaki bireyleri oluşturan tasarım değişkenlerinin potansiyelini araştırmak üzere, bir önceki nesilden daha iyi nitelikler içeren yeni bireyler yaratmak amacıyla kopyalama ve seçim işleminden sonra çaprazlama operatörü kullanılmaktadır. Çaprazlama, popülasyondaki mevcut çözüm potansiyelini kullanan bir operatördür. Bu operatör temel olarak, popülasyondaki tasarım değişkeni sayısı aynı kalırken keyfi olarak belirlenen eşler (ebeveynler) arasında seçili tasarım değişkenlerinin karşılıklı olarak yer değiştirmeleri ile ebeveynler yerine yeni bireyler üretmektedir. Yeni bireyler çocuk olarak nitelendirilmektedir. Çaprazlama operatöründe çaprazlama olasılığı olarak iki tanım yapılmaktadır. Birinci tanıma göre birey çiftlerindeki tasarım değişkenleri çaprazlama olasılığı ile çaprazlamaya uğramak üzere seçilmekte ve sadece o çiftlere çaprazlama uygulanmaktadır. İkinci tanım ise her çift için çiftler arasında çaprazlama olasılığına bağlı olarak çaprazlama yapılmaktadır. Her iki anlamda da çaprazlama olasılığının artması, bazen iyi bireylerin bozulma olasılığını artırmaktadır. Çaprazlama türleri aşağıda kısaca açıklanmaktadır.

Aritmetik Çaprazlama: Bu çaprazlama sadece reel kodlamada tercih edilebilmektedir. Ebeveynlerin her bir karakterinin değerlerinin karşılık olarak ortalaması alınarak çocuk oluşturulmaktadır. Operatörün yapısı gereği tek çocuk oluşmaktadır. $K_1, \dots, K_5; M_1, \dots, M_5;$ dizileri ile ifade edilen ebeveynlerin bu operatöre göre çocuğu $\frac{K_1 + M_1}{2}, \frac{K_2 + M_2}{2}, \frac{K_3 + M_3}{2}, \frac{K_4 + M_4}{2}, \frac{K_5 + M_5}{2}$ olmaktadır (Rasheed, 1998).

Lineer Çaprazlama: Sadece reel kodlamada kullanılan diğer bir çaprazlama tekniğidir. Ebeveynlerin karakterleri \bar{K} ve \bar{M} ve "a" sabiti $[0,1]$ arasında değişen keyfi bir reel sayı olmak üzere, çocuk $a.\bar{K} + (1-a).\bar{M}$ formülünde ebeveynlerin karşılıklı her karakteri için oluşturulmaktadır. Eğer $a=0,5$ olursa bu operatör aritmetik çaprazlama operatörü ile aynı olmaktadır (Rasheed, 1998).

Höristik Çaprazlama: Reel kodlamada kullanılan başka bir çaprazlama operatörüdür. Diğer reel kodlama operatörlerine göre araştırma uzayını daha etkin taramaktadır. Örneğin, \bar{K} ve \bar{M} ebeveynlerinden \bar{M} 'nin uygunluğu daha iyi olduğu kabul

edilsin. Öyleyse çocuk karakterleri "r" değeri [0,1] arasında bir sayı olmak üzere $\bar{M} + r.(\bar{M} - \bar{K})$ formülünden bulunmaktadır.

Tek Noktalı Çaprazlama: En basit çaprazlama yöntemi olan tek noktalı çaprazlamanın yapılabilmesi için her iki bireyin de aynı gen uzunluğunda olması gerekmektedir. Ebeveynler keyfi olarak seçilen bir noktadan ikiye ayrılmaktadırlar. Ayrılan kısım, yer değiştirmeden, bireyler arasında değiş tokuş edilir. Böylece iki yeni birey meydana gelmektedir. Örneğin, tasarım değişkenlerinin tek karakterle temsil edildiği 5 tasarım değişkenli iki birey alınsın. Birinci bireyin genotipi K1, K2, K3, K4, K5 ikinci bireyin genotipi ise M1, M2, M3, M4, M5 ve çaprazlama noktası 3 olsun. Öyleyse yeni bireyler M1, M2, M3, K4, K5 ; K1, K2, K3, M4, M5 şeklinde oluşmaktadır.

İki Noktalı Çaprazlama: Bu teknikte bireyler keyfi şekilde iki noktadan kesilmekte ve bu iki nokta arasındaki kısımlar bireyler arasında değiştirilmektedir. Keyfi olarak seçilen noktalar tasarım değişkenlerinin arasına denk gelmelidir. Aksi halde bireyler arasında çaprazlama yaparken aynı zaman da iki bireyde olmayan tasarım değişkenleri çocuklara aktarılmış olmaktadır.

Düzenli Çaprazlama: Sadece ikilik kodlamada uygulanabilmektedir. Keyfi biçimde ebeveynlerin dizi uzunluğunda çaprazlama maskesi adı verilen bir dizi oluşturulmaktadır. Bu dizi üstündeki karakter değeri "1" ise, birinci çocuğun geni birinci ebeveyninden, "0" ise ikinci ebeveyninden alınmaktadır.

Birinci ebeveyn	1001 1010 0011 0110
İkinci ebeveyn	0110 0011 1110 0100
Çaprazlama maskesi	1111 0000 1100 0011
Birinci çocuk	1001 0011 0010 0110
İkinci çocuk	0110 1010 1111 0100

Yukarıdaki operatörde yapılan açıklama gereği ebeveynlerde olmayan tasarım değişkenlerinin çocuklara aktarılmaması için çaprazlama maskesi 1111 0000 1111 0000, 0000 1111 0000 1111 gibi düzenlenebilir. Böylelikle keyfi olarak tek noktalı, iki noktalı veya çok noktalı çaprazlamada yapılabilmekte ve diğer kodlama türleri içinde bu operatör

kullanılabilmektedir. Genelde düzgün çaprazlama ayrık değişkenli tasarımın yapıldığı GA'da, bir noktalı çaprazlama da sürekli tasarımın yapıldığı GA'da tercih edilmektedir (Chen, 1997).

Çok Noktalı Çaprazlama: Bu çaprazlama türü her türlü kodlamaya uygulanabilmektedir. Ebeveynler üzerinde keyfi sayıda tasarım değişkenine karşılık gelen karakter veya karakterler dizisi seçilir. Bireyler arasında seçilen kısımlar değiş tokuş edilerek çocuklar oluşturulmaktadır. Diğer çaprazlama yöntemlerine göre popülasyonun çözüm bulma potansiyelini en etkin biçimde kullanmaktadır. Çok noktalı çaprazlama bu çalışmada tercih edilen operatör türüdür. Çok noktalı çaprazlama ile düzgün çaprazlamanın tek farkı, düzgün çaprazlamada maske adında ayrıca bir dizinin kullanılmasıdır.

Yoshimoto vd. (2003) reel kodlamada kullanmak üzere, yeni bir yöntem geliştirmişlerdir. İşleyiş tarzı gereği çok noktalı çaprazlama operatörüne benzeyen bu yöntem en iyi bireyin en yüksek uygunluğa sahip olduğu düşüncesi ile birlikte diğer kodlama türleri içinde kullanılması için yeniden düzenlenmiştir. Yöntemin çok noktalı çaprazlamadan farkı, değiştirilecek kısımların sayısına keyfi değil de bireylerin uygunluklarının söz sahibi olduğu bir çaprazlama oranı ile karar verilmesidir. P_c çaprazlama olasılığı; f_{\max} maksimum uygunluğu, \bar{f} ortalama uygunluk değerini, f' çaprazlanacak iki bireyin en büyük uygunluklu olanın uygunluğu göstermek üzere

$$P_c = \begin{cases} \frac{(f_{\max} - f')}{(f_{\max} - \bar{f})} & (f' \geq \bar{f}) \\ 1 & (f' < \bar{f}) \end{cases} \quad (50)$$

bağıntısı ile verilmektedir. Uygunluğu \bar{f} 'e eşit olan veya daha büyük olan bireyler çaprazlanır. Çaprazlama olasılığı f' , f_{\max} 'a yaklaştıkça küçülmekte ve hatta sıfır olabilmektedir. Bu operatörle amaç birey uygunluğu iyiye gittikçe çaprazlama operatörünün etkinliğini azaltarak o bireyin popülasyonda hayatta kalma şansını artırmaktadır. Ayrıca popülasyonda yakınsama arttıkça bireyler iyiye doğru gittiklerinden çaprazlama oranı sıfıra yaklaşmaktadır. Tablo 6'da çaprazlama olasılığının nasıl işleme katıldığı ve yürütüldüğünün uygulaması görülmektedir.

Tablo 6.Çaprazlama yapılacak tasarım değişkeni sayısının belirlenmesi

1. Eş	2. Eş	f'	$f_{\max} - f'$	f_{ort}	$f_{\max} - f_{\text{ort}}$	$\frac{(f_{\max} - f')}{(f_{\max} - f_{\text{ort}})}$	P_c	Çaprazlama sayısı	Dönüştürülmüş değer
1,935	1,947	1,947	0	1,268	0,679	0	0	0	0
1,145	0,878	1,145	0,802	1,268	0,679	1,181	1	4	4
1,545	1,614	1,614	0,333	1,268	0,679	0,490	0,490	1,960	1
1,056	1,223	1,223	0,724	1,268	0,679	1,066	1	4	4
0,508	0,367	0,508	1,439	1,268	0,679	2,119	1	4	4
1,789	1,209	1,789	0,158	1,268	0,679	0,233	0,223	0,892	0

Bu tabloda görüldüğü gibi uygunluk değeri yüksek olan bireyler küçük uygunluk değerli eşlerle eşleştiklerinde çaprazlama yapılmamaktadır.

Genetik operatörlerden etki alanlarına göre kombinasyonlar oluşturulabilmektedir. Farklı operatörler bir arada kullanılabileceği gibi, bir operatörün farklı türleri de bir arada kullanılabilmektedir. Örneğin, çaprazlama operatörü olarak, optimizasyon süresinin belirli bir devresinde, keyfi biçimde belirlenen başlangıç popülasyonunun çözüm potansiyelini zenginleştirerek iterasyon sayısını kısaltmak için düzgün çaprazlama kullanılmakta ve daha sonra bir, iki noktalı veya diğer çaprazlama türlerine geçilmektedir.

Tek noktalı çaprazlama ile araştırma uzayını etkin bir biçimde taramak zor olabilmektedir. Bir bireyde ihlale neden olan ardışık olmayan iki tasarım değişkeni için aynı anda sadece bu tasarım değişkenlerini değiştirmek imkansızdır (Chen, 1997). Aynı durum, tasarım değişkeni sayısı arttıkça iki noktalı çaprazlamalarda da mümkün olabilmektedir. Dolayısıyla tasarım değişkenleri sayısına göre çaprazlama operatörünün GA'daki etkinliğini korumak için farklı çaprazlama türlerini dikkate almak gerekmektedir. Bu açıdan bakıldığında bu çalışmada kullanılan çaprazlama operatöründe, çaprazlama oranı kullanıldığı için bu olumsuzluktan etkilenilmemektedir.

2.4.5.3. Mutasyon Operatörü

Çaprazlama, mevcut tasarım değişkenlerinin potansiyellerini araştırmak üzere kullanılmaktadır. Fakat popülasyonda gerekli sayıda kodlanmış bilgi mevcut değilse, çaprazlama tatmin edici bir çözüm üretmemektedir. Bundan dolayı, mevcut bireylerden yeni bireyler üretme yeteneğine sahip bir operatör gerekmektedir. Bu görevi mutasyon gerçekleştirmektedir (Emel ve Taşkın, 2002). Mutasyon, popülasyondaki genetik çeşitliliği

sağlayan ve koruyan bir operatördür. Mutasyon yapılarak mevcut genetik bilgi kapasitesinin; araştırma uzayındaki çeşitliliği oldukça temsil etmesi ve araştırma uzayının daha kapsamlı taranması sağlanmaktadır. Mutasyon tüm kodlama türlerinde en basit şekilde, bir bit değerini, kodlamaya adını veren tanım aralığındaki - n'lik kodlamanın tanım aralığı [0, 1, 2, 3,..., n-1] - başka bir bit değerine dönüştürmekle yapılabilmektedir. Mutasyon, bireyde hiçbir özelliği sorgulamadan işlem yaptığı için genellikle popülasyonda herhangi bir bireyin bir veya iki tasarım değişkenini değiştirecek şekilde en düşük seviyede ve sabit olasılıkla uygulanmaktadır. Öte yandan sabit bir mutasyon olasılığında başka, popülasyonun yapısına dayalı değişen mutasyon olasılığı gibi değişimlerde kullanılmaktadır. Ayrıca mutasyon kullanımı, zamansız yakınsamaları önlemektedir. Zamansız yakınsamadan kasıt, popülasyonun orta karar çözümler etrafında yakınsamasıdır. Bu yersiz yakınsamayı önlemek için çeşitliliğin azaldığı durumlarda mutasyon devreye girerek çeşitliliği artırmaktadır (Arun, 1996). Öte yandan mutasyon operatörünün de birey belirlemede daha etkin şekilde kullanılması da mümkündür. Örneğin bazı GA'larda operatör olarak sadece mutasyon kullanılmaktadır.

Mutasyona uğrayacak karakterlerin seçilme olasılığı mutasyon olasılığı parametresi tarafından kontrol edilmektedir. Mutasyon, P_m mutasyon olasılığı ile bir bireydeki her bitte meydana gelebilir. Eğer mutasyon olasılığı gereğinden fazla artarsa, genetik arama kayıp genetik malzemeyi bulmaktan ziyade, keyfi bir aramaya dönüşebilmektedir (Emel ve Taşkın, 2002).

Çaprazlamada olduğu gibi mutasyonda da kontrol parametresi olan mutasyon olasılığının iki tip kullanımı bulunmaktadır. Literatürde bu olasılık, popülasyon içindeki birey sayıları için uygulandığı gibi birey içindeki tasarım değişkeni sayısı için de uygulanmaktadır. Bu çalışmada, bu olasılığa bağlı olarak bir birey içinde seçilecek sayıdaki tasarım değişkenlerinin değerleri değiştirilmektedir. Bu tasarım değişkenlerinin ait olduğu birey artık yeni bir çözümü temsil etmektedir. Yoshimoto vd. (2003) ayrıca mutasyon olasılığı için de bir formülasyon gerçekleştirmişlerdir. Çaprazlama oranında da olduğu gibi en iyi bireyin en yüksek uygunluğa sahip olduğu düşüncesi ile birlikte diğer kodlama türlerinde kullanılmak üzere, mutasyon olasılığı

$$P_m = \begin{cases} 0,5(f_{\max} - f) / (f_{\max} - \bar{f}) & (f \geq \bar{f}) \\ 1 & (f < \bar{f}) \end{cases} \quad (51)$$

şeklinde yeniden formüle edilmiştir. f birey uygunluğu, \bar{f} 'den küçük olduğunda %100 mutasyon, \bar{f} 'ye eşit olduğunda %50 mutasyon yapılmaktadır. Ayrıca f , f_{\max} 'a yaklaştıkça mutasyon olasılığı sıfıra ulaşmaktadır. Bu formül, birey uygunluğu arttıkça mutasyon olasılığının azalmasını sağlayacak şekilde düzenlenmiştir.

Mutasyon olasılığının nasıl işleme katıldığı ve yürütüldüğünün uygulaması da Tablo 7'de görülmektedir. Bu tablonun son sütununda, örnekteki çaprazlanacak tasarım değişkenleri sayısı bireyler bazında verilmektedir. Bireyin tüm değişkenlerinin çaprazlanmasının, ebeveynlerin aynen bir sonraki nesile aktarılması anlamına geldiğinden sanki etkin bir işlem yapılmıyormuş izlenimi oluşmaktadır. Ancak yine tablodan görüldüğü gibi o bireylerdeki mutasyon olasılığı oldukça yüksektir. Dolayısıyla çaprazlama operatöründe uygunluğu, çaprazlama yapmaya yeterli bulunmayan bireyler mutlaka mutasyon operatöründe değerlendirilmektedir. Bu bağlamda operatörlere bir bütün olarak bakıldığında çözümlerin değerlendirildiği ve uygun olmayan çözümlerin belirlenip ya tamamen ya da kısmen iyileştirilerek araştırma uzayının taranmasına devam edildiği görülmektedir. Çaprazlama ve mutasyon operatörlerinin uygulanması sırasında kötü, iyi birey ayrımı ortalama uygunluk değerine göre yapılmaktadır. Uygunluğu, ortalama uygunluktan (\bar{f}) büyük olan bireyler iyi, küçük olan bireyler ise kötü birey olarak nitelendirilmektedir.

Tablo 7. Mutasyona uğrayacak tasarım değişkeni sayısının belirlenmesi

f	$f_{\max} - f$	$f_{\max} - f_{\text{ort}}$	f_{ort}	$0,5(f_{\max} - f) / (f_{\max} - f_{\text{ort}})$	P_m	Mutasyon sayısı	Dönüştürülmüş sayı	Çaprazlanacak değişken sayısı
1,935	0,012	0,679	1,268	0,009	0,009	0,035	0	0
1,145	0,802	0,679	1,268	0,591	0,591	2,362	2	4
1,545	0,402	0,679	1,268	0,296	0,296	1,184	1	1
1,056	0,891	0,679	1,268	0,656	0,656	2,624	2	4
0,508	1,439	0,679	1,268	1,060	1	4	4	4
1,789	0,158	0,679	1,268	0,116	0,116	0,465	0	0
0,367	1,580	0,679	1,268	1,163	1	4	4	4
1,947	0	0,679	1,268	0,000	0,000	0	0	0
1,209	0,738	0,679	1,268	0,543	0,543	2,174	2	0
0,878	1,069	0,679	1,268	0,787	0,787	3,149	3	4
1,614	0,333	0,679	1,268	0,245	0,245	0,981	0	1
1,223	0,724	0,679	1,268	0,533	0,533	2,133	2	4

Mutasyona uğratılacak birey ve o bireyde kaç tane tasarım değişkeninin olduğu belirlendikten sonra hangi tasarım değişkenlerinin seçilmesi gerektiği sorusuna cevap aranmalıdır. Genelde bu işlem keyfi olarak yapılmaktadır. Fakat mutasyon aracılığıyla bireyin uygunluğunu düşüren tasarım değişkenlerinin tümünün veya bir kısmının değişimi yapılırken, bireye daha az ceza getiren bir tasarım değişkeni de değiştirilebilmektedir. Bu yersiz işlem optimizasyon sürecini en azından zaman açısından olumsuz yönde etkilemektedir. Hangi tasarım değişkeninin mutasyona uğrayacağına karar verirken onların aldığı ihlaller göz önünde tutulup bir sıralama yapılmalı ve belirlenen sayı kadar en fazla ihlal yapan tasarım değişkeninden başlanmalıdır. Bu işlemi yapan mutasyon türü "Akıllı Mutasyon"dur. Bu çalışmada da bu mutasyon türü kullanılmıştır. Öte yandan operatörler içindeki karar mekanizmaları arttırılırken GA'nın büyümesini oluşturan keyfilik kavramı çok dikkatli bir biçimde sınırlandırılmalıdır. Yoksa gereksiz yere çeşitlilik azalmasına neden olup global olan optimum çözüme ulaşma şansı azaltılabilmektedir. Örneğin; yapısal optimizasyonda mutasyona uğratılacak tasarım değişkeni, belirtildiği gibi bulunduktan sonra, yerine gelebileceklerin –enkesit alanları tasarım değişkeni olarak düşünüldüğünde- şu andaki değerinden daha küçük olmaması istenebilmektedir. Fakat bireydeki başka bir tasarım değişkeninin ilerde değişmesi, değiştirilmesi düşünülen bu tasarım değişkeninin değerinin yeterli olmasına vesile olabilmektedir. Dolayısı ile böyle bir sınırlandırma, taramada olumsuz atlamalara neden olabilmektedir. Jenkins (2002) çalışmasında akıllı mutasyonu kullanmıştır. Ancak ona göre akıllı mutasyon bir sınırlayıcı gibi, gerilmesi yüksek olan elemanın enkesit alanını arttırmakta, gerilmesi yeteri kadar az olan elemanın ise enkesit alanını azaltmaktadır.

Yukarıda izah edilmeye çalışılan çaprazlama ve mutasyon olasılıkları kavramlarıyla bir bakıma bir önceki nesildeki iyi bireylerden birinin veya birkaçının bir sonraki nesile aktarımını sağlayan elitist strateji de uygulanmaktadır. Uygunluğu en yüksek olan birey ne çaprazlanmaya ne de mutasyona uğramaktadır. Hatta uygunluğu f_{max} 'a yakın olan bireylere bile operatörler uygulanmamaktadır (bkz. Tablo 6 ve 7). Öte yandan elitist stratejide bir sonraki nesile aktarılması düşünülen bireylerden birer tane veya birkaç tane de aktarılabilmektedir. Bu bağlamda elitist stratejiden farklı olarak aktarılabilecek bireylerin adet bazında birden fazla olma şansı yoktur.

Genelde mutasyon operatörü çaprazlama operatöründen sonra uygulanmaktadır. Fakat bu çalışmada mutasyon operatörü çaprazlama operatöründen önce uygulanmıştır. Mutasyon operatörü olarak akıllı mutasyon kullanıldığından, birey uygunluklarına göre

yapılan değerlendirme çaprazlamadan sonraki bireyleri değil önceki bireyleri ilgilendirmektedir.

2.4.6. Yeni Kuşağın Oluşması ve Döngünün Durdurulması

Yeni nesil, kopyalama işlemiyle uyum faktörüne bağlı olarak çaprazlama havuzu oluşturulması ve çaprazlama işlemi yapılarak mutasyona geçilmesinden sonra tanımlanmakta ve bir sonraki neslin ebeveynleri olmaktadır. Bu süreç; çözüm dizisi topluluğu içindeki bireylerin %80 veya %85'i aynı bireyden oluşana kadar ya da başka bir durdurma kriteri sağlanana kadar devam etmektedir. Ayrıca istenen hassasiyet derecesine göre de maksimum iterasyon sayısı belirlenebilmekte ve iterasyon bu sayıya ulaştığında döngü durdurulabilmektedir.

2.5. Genetik Algoritmanın Avantaj ve Dezavantajları

Optimizasyon problemlerinin hemen hemen hepsi çok geniş bir çözüm havzasının taranmasını gerektirmektedir. Bu çözüm havzasının geleneksel yöntemlerle taranması çok uzun sürmesine karşın, GA ile kısa bir sürede kabul edilebilir bir sonuç alınabilmektedir (URL-9, 2003).

GA, çözümlerden oluşan popülasyonu eş zamanlı incelediğinden yerel (lokal) çözümlere takılmamaktadır (Emel ve Taşkın, 2002). Bundan dolayı GA global çözümler sunabilmektedir (Erbatur vd., 2000).

Geleneksel optimizasyon yöntemlerine göre farklılıkları olan GA, parametre kümesini değil kodlanmış biçimlerini kullanmaktadır. Olasılık kurallarına göre çalışan GA, yalnızca amaç fonksiyonuna gereksinim duymaktadır (Emel ve Taşkın, 2002).

GA, ayrık tasarım değişkeni kullanması, sınır ifadelerini içermesi, çoklu yük durumlarının dikkate alması, probabilistik karakteri ve çoklu mümkün çözümleri araştırması gibi önemli özelliklere sahip olduğundan önemli avantajlara haizdir (URL-7, 2003).

GA, amaç fonksiyonunun ifade ettiği anlamdan bağımsızdır. Aynı zamanda sistemin sınırlayıcıları hakkında da hiçbir bilgiye ihtiyaç duymamaktadır. GA, bu özelliğiyle her tip optimizasyon problemi için uygulanabilir niteliktedir (Cao, 1996).

GA pratik olduđu için güvenilir ve etkili bir optimizasyon yöntemidir. Özellikle yapay zeka ve fonksiyon optimizasyonunda kompleks ve zor problemlerin çözümü için uygundur (Kılıç, 1997).

GA, pek çok problem türü için uygun parametreler ile çalışıldığı taktirde optimuma yakın çözümler vermektedir (URL-10, 2003).

GA sınırlandırılmamış optimizasyon yöntemidir. Bundan dolayı sınırlandırılmış optimizasyon problemlerinin sınırlandırılmamış hale getirilmesi gerekmektedir (Kılıç, 1997).

GA'nın geleneksel optimizasyon yöntemlerinde olduğu gibi sözü edilen diğer yapay zeka araçlarına göre de çeşitli alanlarda üstünlükleri bulunmaktadır. Bu üstünlükler GA'nın arama yapısı ile ilgilidir (Emel ve Taşkın, 2002).

Geleneksel yöntemlerde çözümler bir önceki adımdaki çözümlere göre bulunduğundan GA'ya kıyasla global optimum bulma olasılıkları daha azdır (Kılıç, 1997).

GA pratik çözümler sunduğundan insan performansına en az ihtiyaç duyan yöntemlerden biridir (Kılıç, 1997).

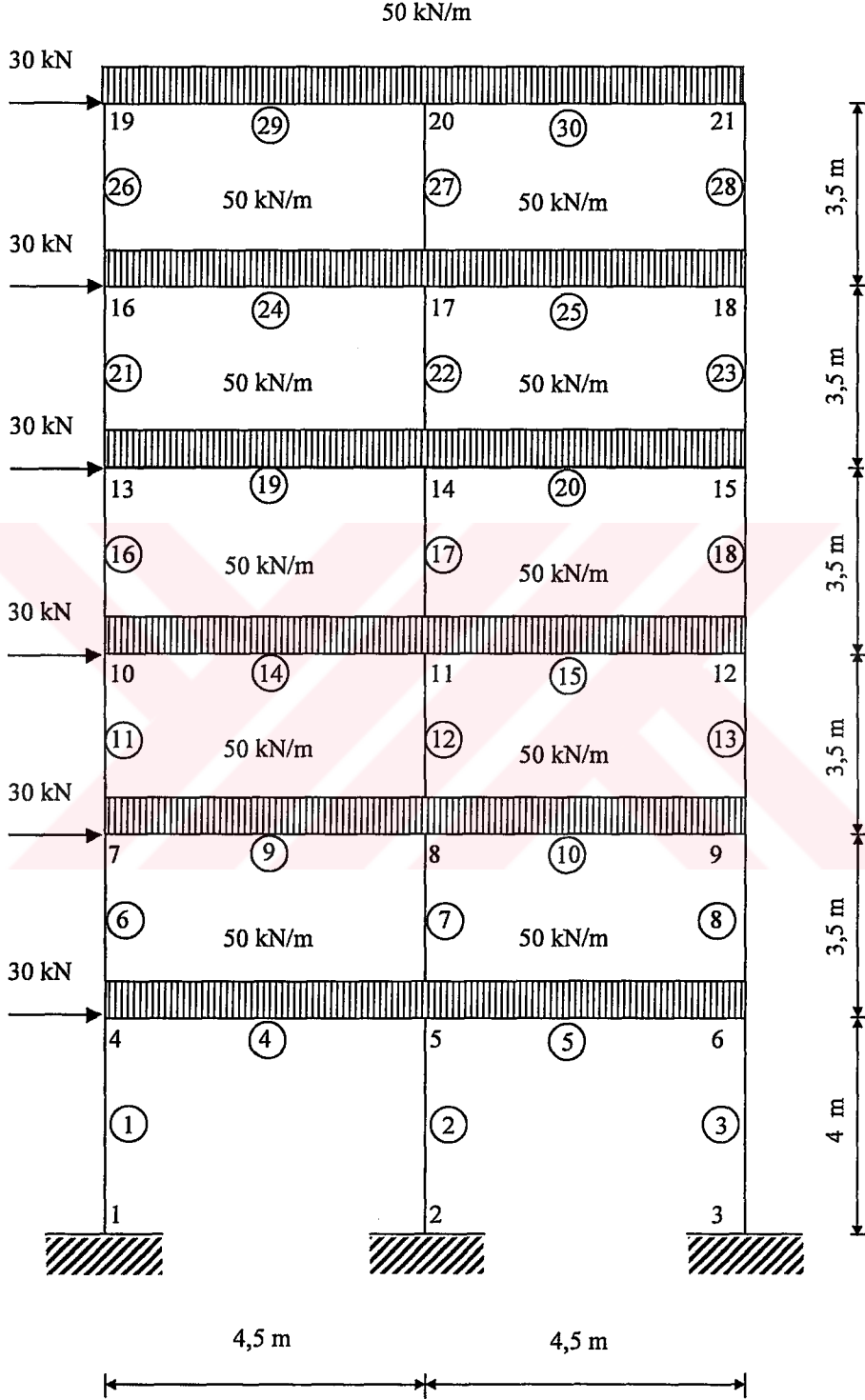
2.6. Çerçeve Sistemlerin Tasarım Örnekleri

Bu başlık altında çelik çerçeve sistemlerin, Fortran dilinde kodlanan genetik algoritma programı ile optimum tasarımı gerçekleştirilmektedir. Programın, her sistem elemanı için ayrı hesaplanan k değerleri, uygulanan çok noktalı çaprazlama ve akıllı mutasyonun uygulanabilirliğini göstermek amacıyla literatürden seçilen örnekler üzerinde sonuçlar karşılaştırılmaktadır. Bu örnekler altı katlı iki açıklıklı çerçeve, diyaonal kafes elemanlı altı katlı iki açıklıklı çerçeve (Armutçu, 1997; Daloğlu ve Armutçu, 1998), bir katlı bir açıklıklı çerçeve (Tütüncü, 1995) sistemlerden oluşmaktadır. Altı katlı çerçeve sistemler Saka (1991) tarafından AISC-ASD dikkate alınarak optimumluk kriteri yöntemiyle optimize edilmiştir.

2.6.1. Altı Katlı İki Açıklıklı Çerçeve

Şekil 8'de görülen, ankastre mesnetler üzerine oturan, altı katlı iki açıklıklı çerçevenin TS 648 ve TS 910 dikkate alınarak GA ile optimum boyutlandırılması

yapılmaktadır. Çerçevedeki elemanların grup numaralarına ait bilgiler Armutçu'nun (1997) çalışmasına paralel olarak Tablo 8'de verilmektedir.



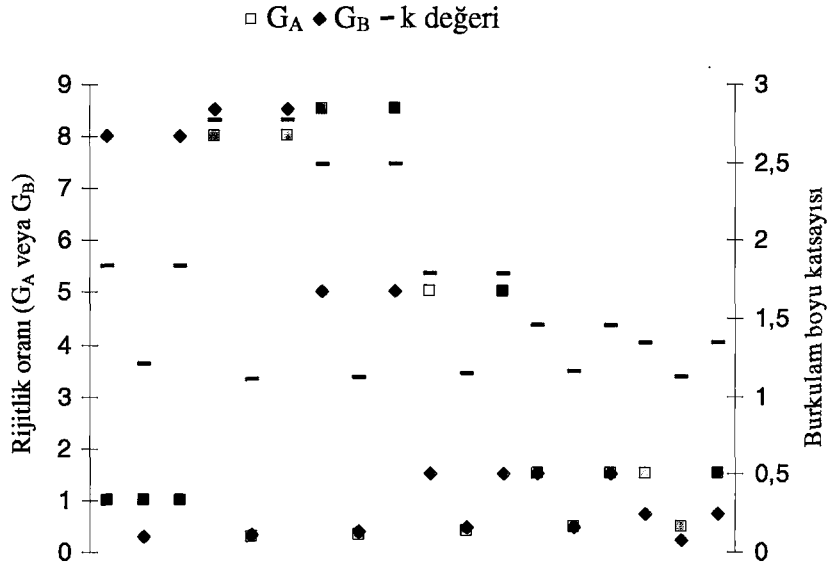
Şekil 8. Altı katlı iki açıklıklı çerçeve

Tablo 8. Altı katlı iki açıklıklı çerçevenin eleman grup numaraları

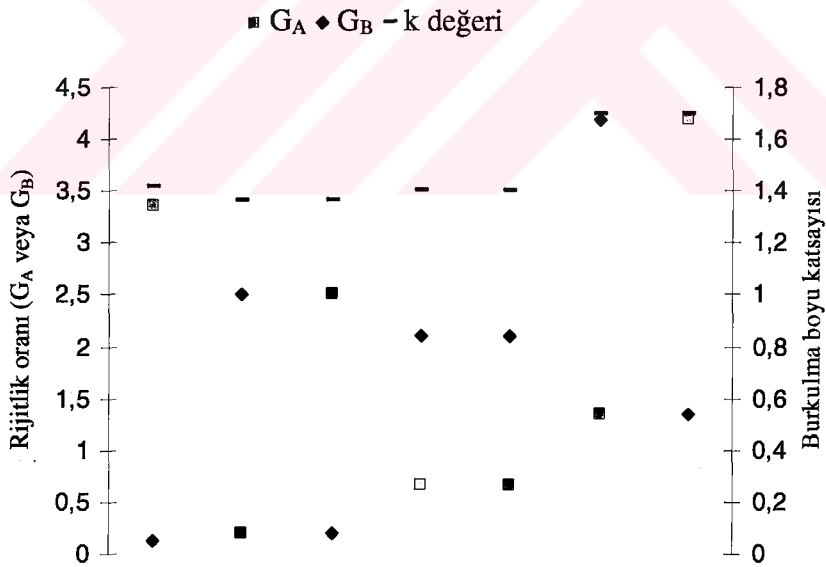
Grup numarası	Eleman numarası
1	1,3,6,8,11,13
2	16,18,21,23,26,28
3	2,7,12
4	17,22,27
5	4,5,9,10,14,15,19,20,24,25,29,30

Sistemde tüm kirişler üzerinde 50 kN/m'lik düzgün yayılı yük ile kat hizalarında 30 kN'luk yatay tekil yükler tanımlanmaktadır. Çerçevede en üst katta izin verilen maksimum yatay yerdeğiştirme 7,16 cm'dir. Şekil 9 ve 10'da görüldüğü gibi k değeri çoğunlukla Armutçu (1997) tarafından kirişlerde 1, kolonlarda 1,2 olarak alınan sabit değerlerden büyüktür.

Bu örnek, kolonlarda geniş başlıklı I (IPG) profili, kirişlerde ise orta geniş başlıklı I (IPOG) profili kullanılarak çözülmüştür. Ayrıca ikinci tasarımında kolonlarda orta geniş başlıklı I (IPOG) profili de kullanılmıştır. Bu çalışmada ise Armutçu (1997) tarafından kullanılan profillere ek olarak da kolonlarda hafif tip geniş I (IPG_h), orta geniş I (IPOG) profilleri kullanılmaktadır. Analizde elastisite modülü $E=21000 \text{ kN/cm}^2$, akma gerilmesi $\sigma_a=36 \text{ kN/cm}^2$ olarak dikkate alınmaktadır. Elemanların ağırlıkları her profil için profil tablolarındaki birim boy ağırlıklarından hesaplanmaktadır. Genetik çözüm için 20 adet birey ile maksimum 300 iterasyona izin verilmiş olup yakınsama oranı %85 olarak dikkate alınmaktadır. Cezalandırılmış amaç fonksiyonu için ceza katsayısı $P=2$ alınmaktadır. Kodlama türü olarak da değer kodlaması seçilmektedir. Programın altı defa koşturulması ile elde edilen optimizasyon sonuçları Tablo 9'da verilmektedir.



Şekil 9. Altı katlı iki açıklıklı çerçevenin 5. tasarıma göre kolonlardaki burkulma boyu katsayısı



Şekil 10. Altı katlı iki açıklıklı çerçevenin 5. tasarıma göre kirişlerdeki burkulma boyu katsayısı

Tablo 9. Altı katlı iki açıklıklı çerçevenin tasarım sonuçları

Grup no	Enkesit alanları (cm ²)								
	Armutçu GA (1) (1997)	Armutçu GA (2) (1997)	Armutçu Optimumluk Kriteri (1997)	Bu çalışma					
				1. tasarım	2. tasarım	3. tasarım	4. tasarım	5. tasarım	6. tasarım
1	131,00	156,00	119,67	212,00	198,00	133,00	178,00	143,00	143,00
2	91,00	84,50	74,15	106,00	97,30	64,30	98,60	72,70	84,50
3	254,00	254,00	203,30	149,00	198,00	239,00	159,00	159,00	159,00
4	131,00	134,00	92,51	143,00	86,80	286,00	254,00	98,60	98,60
5	116,00	98,80	104,42	84,50	84,50	72,70	72,70	98,60	98,60
Toplam ağırlık (kN)	119,358	115,687	102,330	114,424	110,157	108,629	112,575	100,433	102,365

Tablo 9'dan görüldüğü gibi bu çalışmada profil listesinin genişletilmesi ile Armutçu (1997) tarafından elde edilen sonuçlara göre daha hafif tasarımları elde edilmektedir.

Bu tablonun 5.,7.,8. ve 9. sütunlarında verilen tasarımlar sırasıyla 300. iterasyonda % 60 yakınsama, 300. iterasyonda % 25 yakınsama, 179. iterasyonda % 85 yakınsama ve 300. iterasyonda % 75 yakınsama ile elde edilmiştir. Bu tasarımlarda yakınsama oranlarının çoğu öngörülen % 85'lik orana sahip değildir. Yakınsama kriterinin, elde edilen yakınsama oranları ile kıyaslandığında optimum çözümün bulunmasında tek başına etkin bir kriter olmadığı görülmektedir. Bundan dolayı tasarımlarda yakınsama kriterinin yanı sıra maksimum iterasyon sayısının da sonlandırma kriteri olarak dikkate alınmasının uygun olduğu anlaşılmaktadır.

Bu tabloda verilen optimum tasarımlara ulaşılırken iterasyon sayısının fazla olması başlangıç neslinin keyfi olarak oluşturulmasından ileri gelmektedir. Örneğin 1. tasarım sonuçlandırıldıktan sonra elde edilen enkesitler performanslarına göre değerlendirilerek 2. tasarımda olduğu gibi programa veri şeklinde girildiğinde, tek bir iterasyonda, 1. tasarıma nazaran daha iyi bir çözüm elde edilmiştir. Dolayısıyla başlangıç neslinin, programın birkaç defa koşturulmasından sonra edinilecek kanı ile kullanıcı tarafından oluşturulması tasarımı daha iyi sonuçlara götürebilmektedir. Öte yandan başlangıç neslinin elle

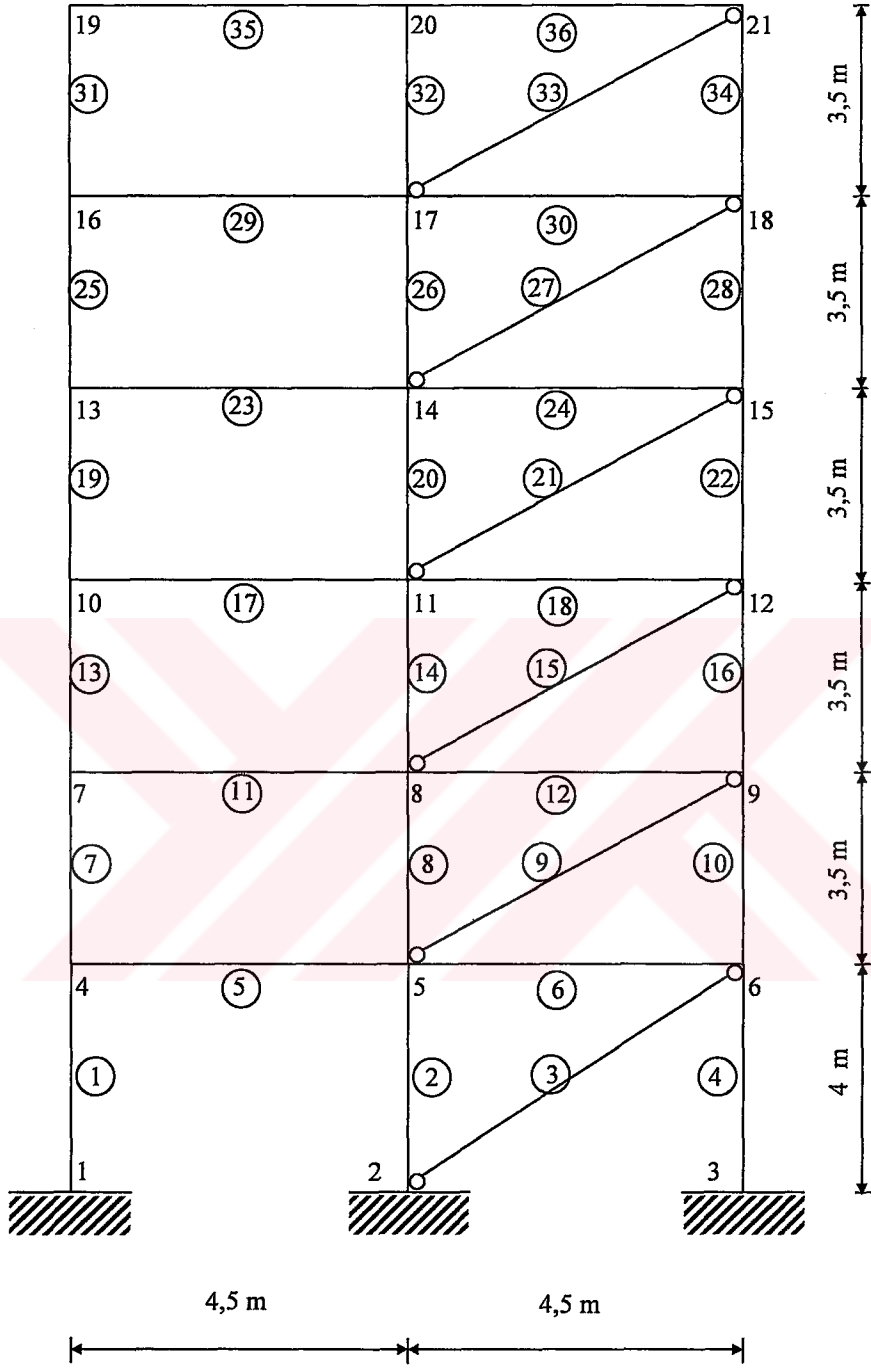
hazırlanmasından ziyade kullanılan veri tabanında daraltmaya gitmek de etkili olabilmektedir. Örneğin, Tablo 9'daki 4., 5. ve 6. tasarımlar, bu düşüncenin ürünü olup 89 adet profilden oluşan veri tabanı 44 adete düşürülerek optimize edilmişlerdir. Ayrıca 6. tasarım, 5. tasarım için bulunan optimum sonucun yer aldığı popülasyonda olup ilk dört tasarıma göre daha hafiftir. Görüldüğü gibi yakınsama sağlanan bir popülasyonda optimum çözümden başka uygulanabilir çözümler de bulunmaktadır.

Bu tablodaki 3. tasarım optimum tasarım olarak kabul edilmemelidir. Bir tasarımın optimum tasarım olması, aynı zamanda, o tasarımın uygulanmasında hiçbir sakıncanın olmamasını gerektirmektedir. Buna göre 3. tasarımda, orta üst kat kolonlarının enkesitleri alt kolon enkesitlerinden büyüktür. Bu nedenle bu tasarım, uygulama açısından, bu yönde bir sınırlayıcı tanımlanmamasına rağmen, uygun değildir. Bu tabloda verilen diğer tasarımlarda bu durum dikkate alınmıştır.

Bu çalışmada kullanılan çok noktalı çaprazlama ve akıllı mutasyon ile optimum sonuca ulaşılırken iterasyonlar boyunca çok iyi bir bireyin şansız bir mutasyonun veya çaprazlamanın uygulanması sonucu kaybolması ihtimali oldukça minimuma indirgenmiştir. Örneğin 5. tasarım için çok zor olsa da tüm iterasyondaki bireyler hem kullanıcı tarafından hem de programda oluşturulan ve tüm optimizasyon süresince cezası sıfır olan en hafif bireyin saklandığı bir mekanizmayla taranmış olup sunulan optimum sonuçtan cezası sıfır olan daha hafif bir sonuca rastlanmamıştır. Kullanılan değer kodlaması sayesinde de 20 bireylik bir popülasyonla etkin bir arama gerçekleştirilmiştir. Eğer başka bir kodlama türü kullanılsa idi bu sonuçlara ulaşmak büyük bir olasılıkla ya programın daha fazla koşturulmasına ya da popülasyon büyüklüğünün büyütülmesine neden olabilirdi. Sonuç olarak tüm tasarımlara göre 5. tasarımın global çözüm olduğu bütün ihtimaller taranmadan iddia edilemese de sahip olmuş olduğu $W=100,433$ kN ağırlık ile yukarıda verilen tablodaki diğer tasarımlara göre daha global olduğu görülmektedir.

2.6.2. Diagonal Kafes Elemanlı Altı Katlı İki Açıklıklı Çerçeve

Ankastre mesnetler üzerine oturan diagonal kafes elemanlı altı katlı iki açıklıklı çerçeve sistem Şekil 11'de görülmektedir. Bu sistem için yapılan optimizasyonda dikkate alınan değerler önceki tasarımdaki sistem ile aynı olup TS 648 ve TS 910 dikkate alınarak GA ile optimum boyutlandırılma yapılmaktadır. Eleman grup numaralarına ait bilgiler Tablo 10'da verilmektedir.



Şekil 11. Diyagonal kafes elemanlı altı katlı iki açıklıklı çerçeve

Tablo 10. Diyagonal kafes elemanlı altı katlı iki açıklıklı çerçevenin eleman grup numaraları

Grup numarası	Eleman numarası
1	1,4,7,10,13,16
2	19,22,25,28,31,34
3	2,8,14
4	20,26,32
5	5,6,11,12,17,18,23,24,29,30,35,36
6	3,9,15,21,27,33

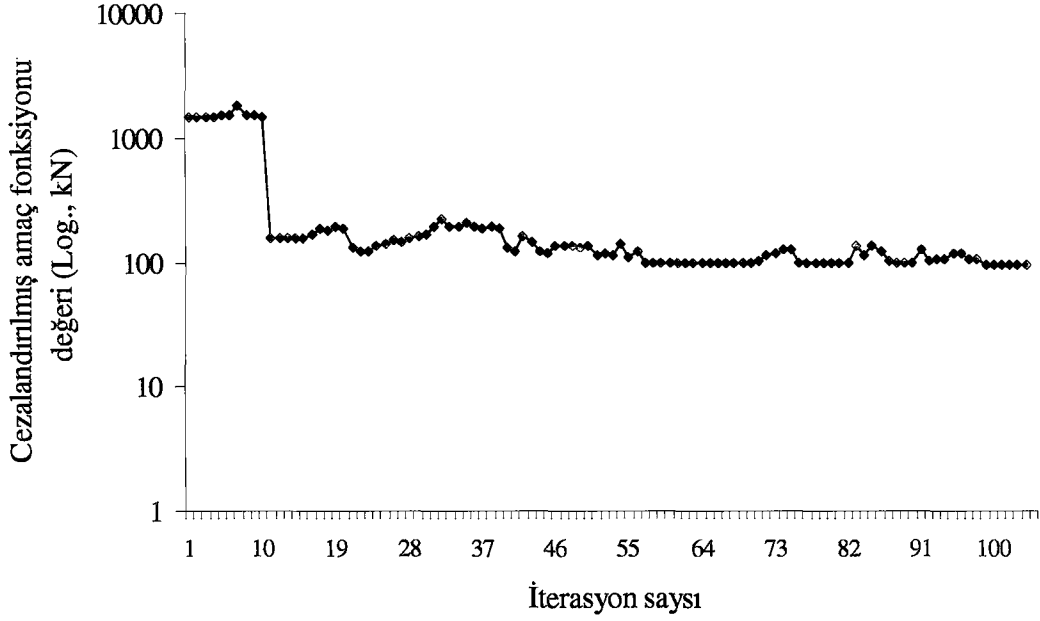
Boyutlandırma için kolonlarda geniş başlıklı I (IPG) profilinden 24 adet, kirişler ve diyagonal kafes elemanlarda ise orta geniş başlıklı I (IPOG) profilinden 18 adet kullanılmıştır. Bu tasarıma ait sonuçlar Tablo 11’de verilmektedir.

Tablo 11. Diyagonal kafes elemanlı altı katlı iki açıklıklı çerçevenin tasarım sonuçları

Grup no	Enkesit alanları (cm ²)	
	Armutçu GA (1997)	Bu çalışma
1	131,00	149,00
2	65,30	78,10
3	216,00	181,00
4	106,00	149,00
5	116,00	62,60
6	16,40	10,30
Toplam ağırlık (kN)	114,400	95,829

Bu tabloda verilen tasarım 104. iterasyonda % 85 yakınsama ile elde edilmektedir. Tasarımın ağırlığı $W=95,829$ kN olup Armutçu’nun (1997) bulduğu tasarımdan daha hafiftir. Ayrıca tasarım değişkeni bazında her iki optimum boyutlandırma incelendiğinde enkesit alanlarının örtüştüğü dolayısıyla kodlanan programın verimli çalıştığı bir kez daha doğrulanmaktadır.

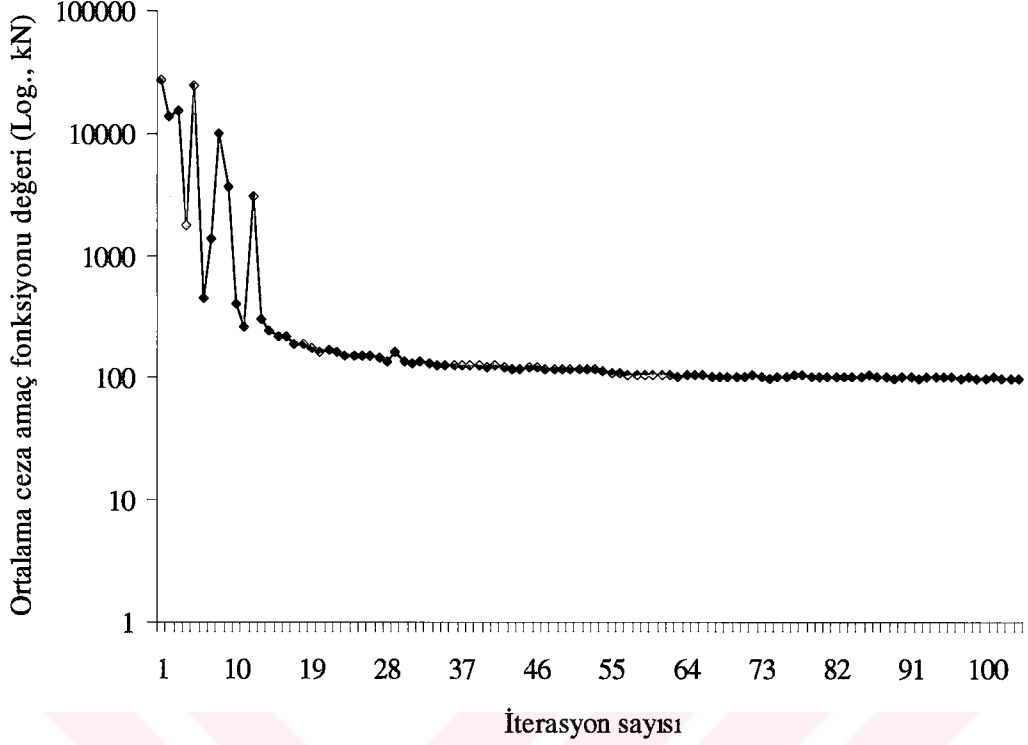
Tasarımın 20. bireyinin optimizasyon süresince cezalandırılmış amaç fonksiyonunun değerinin Şekil 12’de iterasyon sayısına bağlı olarak değişimi verilmektedir.



Şekil 12. Optimum tasarıma sahip 20. bireyin iterasyon sayısına bağlı olarak cezalandırılmış amaç fonksiyonu değişimi

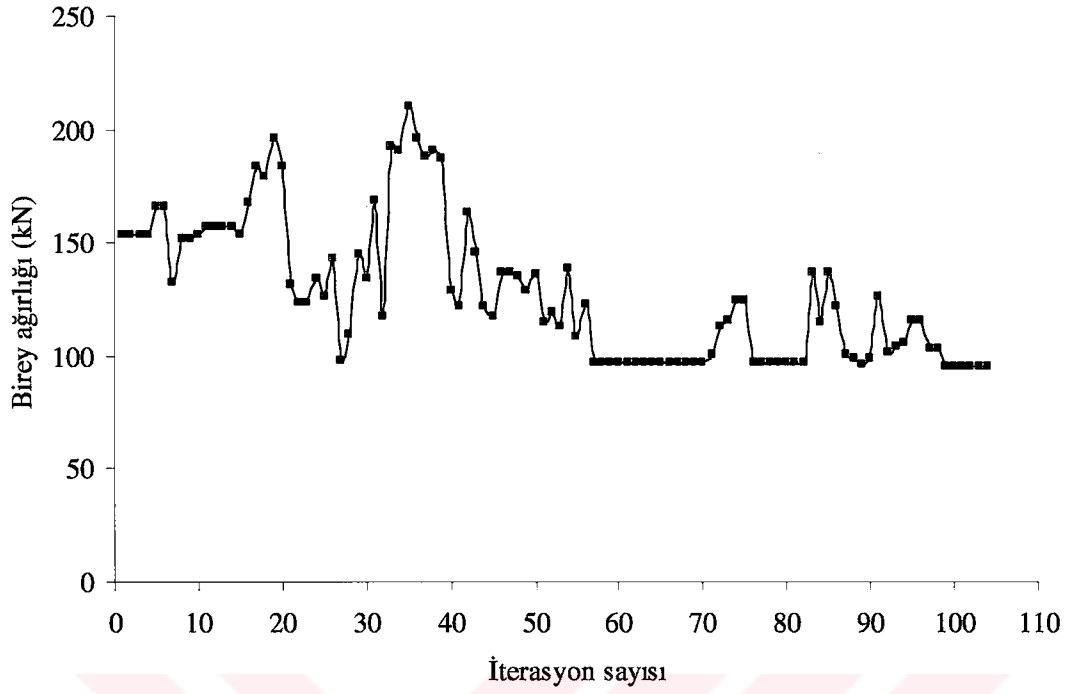
Bu şekilde görüldüğü gibi, cezalandırılmış amaç fonksiyonu yaklaşık 1500 kN değerinden optimum tasarımın da ağırlığı olan $W=95,829$ kN kadar azalmaktadır. Çünkü cezalandırılmış amaç fonksiyonundaki toplam ceza, optimum sonuca varıldığında sıfır olmaktadır. Bu şekildeki yüksek değerler keyfi seçilen başlangıç neslinde 20. bireyin performansının optimum tasarımdan uzak olduğunu ifade etmektedir. 16. iterasyondan 57. iterasyona kadar ki dalgalanma ise algoritmadaki optimum değere ulaşmadaki mücadeleyi göstermektedir. 57. iterasyondan sonra aslında 20. birey için optimum tasarıma ulaşılmıştır. Fakat yakınsama kriteri nedeniyle diğer bireylerde de bu sonucun en az % 85 oranında baskın olması gerektiğinden bu birey için iterasyon 104'e kadar devam etmiştir.

Şekil 13'de ortalama cezalandırılmış amaç fonksiyonunun iterasyon sayısına bağlı olarak değişimi verilmektedir.

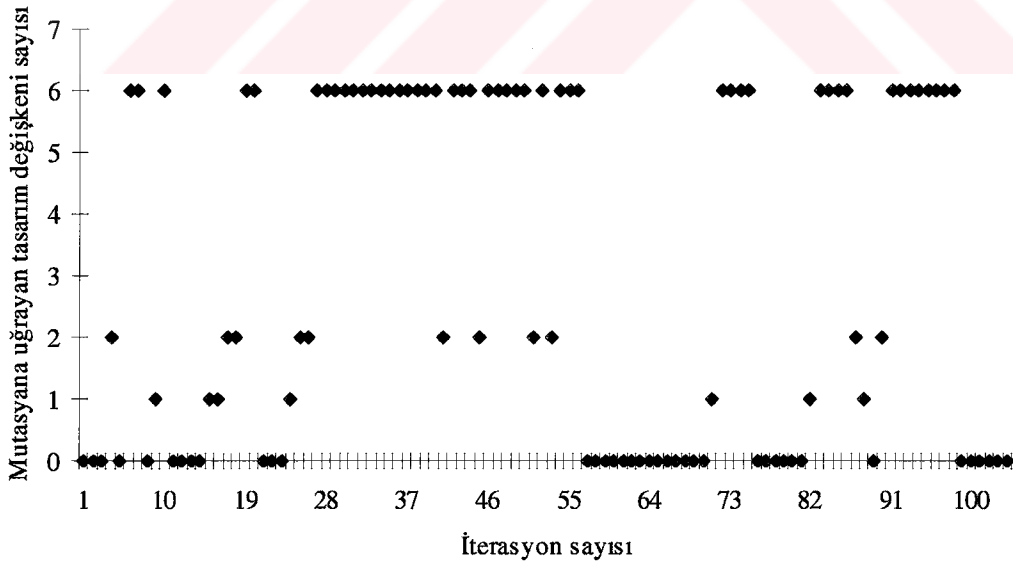


Şekil 13. Ortalama cezalandırılmış amaç fonksiyonunun iterasyon sayısına bağlı olarak değişimi

Bu şekilde de Şekil 12'dekine benzer olarak ilk 12 iterasyonda aşırı kötü bireylerin olduğu, 57. iterasyona kadar optimum tasarıma geçişin sağlandığı ve sonrasında da gerekli yakınsama oranının elde edilmesi beklenmektedir. Şekil 14'de, tasarımın 20. bireyinin ağırlığının iterasyon sayısına bağlı olarak değişimi verilmektedir. Birey ağırlığındaki değişimde de cezalandırılmış amaç fonksiyonlarında ifade edilen özellikler gözlenmektedir. Bu şekildeki değişimlerin nedeni Şekil 15'deki 20. bireyin mutasyona uğrayan tasarım değişkenleri sayısının iterasyon sayısına bağlı değişimi ile bağdaştırılabilir. Şekil 14'den görüldüğü gibi, 57. iterasyona kadarki her büyük ani değişim öncesinde bu problem için tanımlanan bireyi oluşturan 6 adet değişken yani tüm tasarım değişkenleri değiştirilmektedir. Çünkü uygulanan akıllı mutasyon gereği 57. iterasyona kadar 20. birey gerekli performansı yakalayamamıştır. Dolayısı ile popülasyona keyfi birey girişi yapılmaktadır. 20. birey için 57. iterasyondan sonraki ani değişimler bu bireyin popülasyonun ortalama uygunluk değerinden uzaklaştığını göstermekte olup paralel şekilde mutasyonla bu bireyin iyileştirilmeye çalışıldığı görülmektedir. Ayrıca 100. iterasyondan sonra mutasyon miktarının sıfır olması da 20. bireyin popülasyonda en iyi bireyler arasına girdiğini ifade etmektedir.



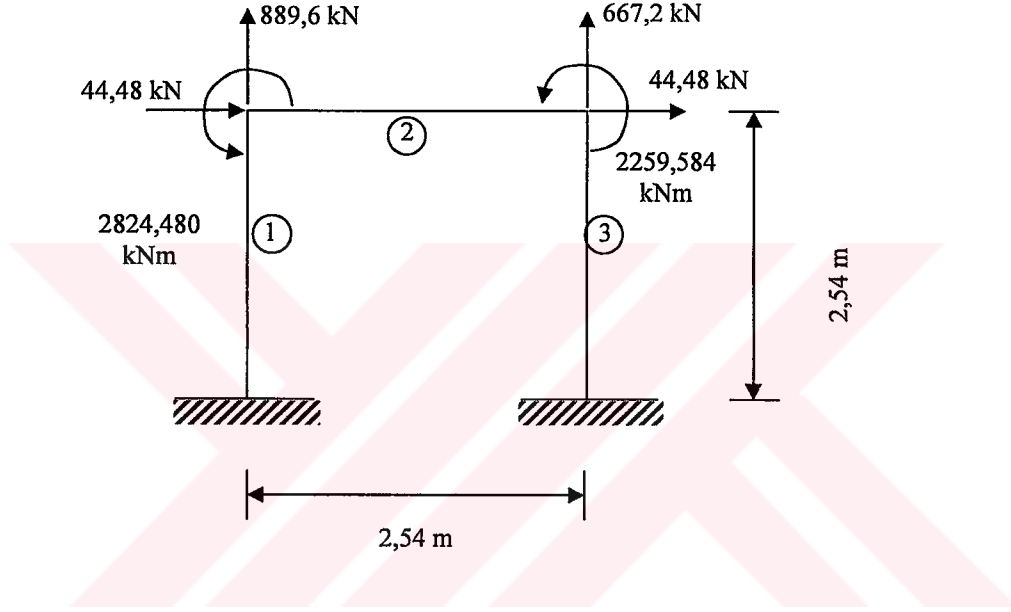
Şekil 14. Optimum tasarıma sahip 20. bireyin ağırlığının iterasyon sayısına bağlı olarak değişimi



Şekil 15. Optimum tasarıma sahip 20. bireyin mutasyona uğrayan tasarım değişkeni sayısının iterasyon sayısına bağlı olarak değişimi

2.6.3. Bir Katlı Bir Açıklıklı Çerçeve

Şekil 16'da görülen bir katlı bir açıklıklı ankastre mesnetler üzerine oturan çerçeve AISC-ASD'ye göre optimize edilmektedir. Tasarımda bu standartta bulunan 295 adet I profili dikkate alınmakta, sistem için her hangi bir yerdeğiştirme sınırlayıcısı uygulanmamakta ve sistem için eleman sayısı az olduğundan gruplandırma yapılmamaktadır.



Şekil 16. Bir katlı bir açıklıklı çerçeve

Sistem kat hizasındaki düğüm noktalarında 44,48 kN (10 kip)'luk yatay tekil kuvvetler, 889,6 kN (200 kip) ve 667,2 kN (150 kip)'luk düşey tekil kuvvetler ile 2824,480 kNm (25000 kip-in) ve 2259,584 kNm (20000 kip-in)'lik tekil momentler ile yüklenmiştir. Elastisite modülü $E=199,955 \times 10^6$ kN/m² (29000 ksi), akma gerilmesi $\sigma_a=248220$ kN/m² (36 ksi) maksimum iterasyon 300, ceza katsayısı $P=0,5$ olarak dikkate alınmıştır.

Bu çerçeve sistemin tasarım sonuçları Tablo 12'de verilmektedir. Bu tablodan görüldüğü gibi, belirlenen $V=225650$ cm³ değerindeki hacim izin verilen maksimum iterasyon sayısı olan 200. iterasyonda % 50'lik bir yakınsamayla optimum sonuç olarak bulunmuştur.

Tablo 12. Bir katlı bir açıklıklı çerçevenin tasarım sonuçları

Grup no	Enkesit alanları (cm ²)					
	Briggs ve Willmert (1978)	Erbatur ve Al-hussainy (1992)	Tütüncü GA (1) (1995)	Tütüncü GA (2) (1995)	Tütüncü GA (3) (1995)	Bu çalışma
1	127,355	247,096	246,451	160,000	159,355	144,516
2	679,870	493,547	462,580	472,902	469,676	508,386
3	194,387	204,516	210,967	220,645	206,451	235,483
Hacim (cm ³)	254409	240070	233680	216801	212212	225650

3. SONUÇLAR VE ÖNERİLER

Bu çalışmanın amacı ayrıık tasarım deęişkenlerini kullanarak geliştirilen bir GA ile çelik çerçeve sistemlerin minimum ağırlıklı tasarımını yapmaktır. Geliştirilen GA'da çok noktali çaprazlama, akıllı mutasyon operatörleri ve farklı kodlama çeşitleri kullanılmaktadır. Bu amaç doğrultusunda çelik çerçevelerin GA ile optimizasyonu üç ayrı sistem üzerinde örneklendirilmiştir. Bu çalışmadan çıkarılabilecek sonuçlar ve öneriler aşağıda maddeler halinde verilmektedir.

- Bu çalışmadan elde edilen sonuçlar, GA'nın çelik çerçevelerin optimizasyonunda kullanılabilir olduğunu göstermektedir.
- GA'da ayrıık tasarım deęişkenleri ile çalışılabildięi için bulunan tasarım sonuçları deęiştirilmeden kullanılabilirlerdir.
- Yakınsama kriterinin tek başına kullanımı bazen optimizasyon sürecinde gereksiz zaman kaybına neden olmaktadır. Yakınsamanın sağlanmaması optimum sonuca ulaşmayı engellemektedir. Düşük yakınsama oranlarında dahi, ideal çözümler bulunabilmektedir. Dolayısıyla yakınsama kriteri programı sonlandıran tek kriter olmamalıdır. Yakınsama kriterinin yanında ikinci bir sonlandırma kriterinin kullanılmasında fayda bulunmaktadır.
- Optimizasyon boyunca bir çok uygulanabilir çözüm bulunmaktadır. Bu çözümlerden biri, süreç içinde baskın gelmekte ve optimum sonuç olarak sunulmaktadır. Fakat sunulan sonucun her zaman programın bir kere koşturulmasından elde edilen tüm uygulanabilir çözümlerin optimumu olduęu garanti edilememektedir. Baskın olan çözümün optimizasyon sürecindeki en ideal sonuç ile aynı olmaması, uygulanan genetik operatör türlerinden kaynaklanabilmektedir. Bu çalışma için seçilen çok noktali çaprazlama ve akıllı mutasyon sayesinde baskın gelen bireyin, programın koşturulması boyunca elde edilen uygulanabilir çözümlerin en ideali ile aynı olması sağlanmaktadır. Böylelikle, bu operatörlerin GA'nın performansını arttırdığı görülmektedir.
- Problemdaki karakteristik sayısına ve veri tabanındaki bilgi sayısına göre farklı kodlama türlerinin kullanımı optimizasyon işlemini verimli hale getirmektedir.

- Başlangıç neslinin kalitesi optimizasyon zamanını kısaltmasıyla doğru orantılı olmaktadır. İyi bir başlangıç nesli programın birkaç kez koşturulmasından sonra elde edilebilmektedir.
- Keyfi olarak başlangıç neslinin oluşturulması, iterasyon sayısını artırabilmektedir.
- Optimizasyon için kullanılan profil veri tabanının optimizasyon içinde sınırlandırılması tasarımı daha etkin çözümlere ulaştırabilmektedir.
- Problem için popülasyon büyüklüğü ile birey uzunluğunun uyumlu olması optimum sonuca ulaşmak için önemli olmaktadır.
- Farklı çaprazlama ve mutasyon operatörleri tanımlanarak daha etkin arama yapılabilir. Fakat GA'nın global sonuç bulma yeteneğini sağlayan keyfilik durumu dikkatlice sınırlandırılmalıdır.
- Bu çalışmada ikilik kodlamanın yanı sıra dörtlük, sekizlik, değer ve reel kodlaması gibi farklı kodlama türlerinin kullanılabilmesi şeklinde hazırlanan bilgisayar programı çerçeve sistemlerin minimum ağırlıklı boyutlandırılmasında etkin bir şekilde kullanılabilir.
- Çerçeve sistemlerde rezonans etkisinden kaçınmak için sistem periyodunun belli değerler arasında kalması gibi GA'da sistemlere ve karşılaşılan problemlere göre farklı sınırlayıcılar oluşturulabilir.
- GA'da sadece uygulanabilir çözümlerin bulunduğu bir popülasyonla optimizasyona başlamak, keyfi olarak oluşturulan bir popülasyona göre iterasyon sayısında etkinlik sağlamaktadır. Sonrasında uygulanabilir çözümlere mutasyon operatörünün yer almadığı genetik operatörler uygulanmalıdır. Popülasyondaki her tasarım değişkeni rasgele bir seçim olmadığından operatörlerin uygulanması sonucunda bulunacak çözümlerin olası optimum çözüme doğru ilerleme şansı daha fazladır. Böylelikle uygun olmayan çözümler için yapılan gereksiz işlemlerden kaçınılmaktadır.
- GA'nın uygulandığı çerçeve sistemler başka eleman modelleri tanımlanarak geliştirilebilir. Örneğin döşeme veya perde duvar gibi düzlem elemanların tanımıyla uygulamada daha sık görülen problemlere GA'nın uygulanması sağlanabilir.
- GA, diğer optimizasyon sistemleri ile birlikte kullanılabilir.
- GA, kafes ve çerçeve gibi sistemlerin boyutlandırılmasında kullanılabilmesi gibi barajlarda uygun dolu savak prototiplerinin boyutlandırılması, yine barajlarda basamaklı su geçişlerindeki optimum basamak genişliği ve basamak yüksekliğinin

bulunması ve istinat duvarları için en uygun kayma dairesi merkezinin bulunması gibi deneme yanılmaya dayalı her türlü probleme uygulanabilir.

- GA ile çerçeve sistemlerde diyagonal ve perde duvar gibi rijitleştirici elemanlarla topoloji optimizasyonu yapılabilir.



4. KAYNAKLAR

- Armutçu, M., 1997. Çelik Çerçevelerin Genetik Algoritma ile Optimizasyonu, Yüksek Lisans Tezi, K.T.Ü., Fen Bilimleri Enstitüsü, Trabzon.
- Arun, C. S., 1996. A Modular Genetic Algorithm and Its Implementation in A Distributed Environment for Structural Optimization, M. Sc. Thesis, The University of Texas at Arlington.
- Aydın, Z., 2000. Uzay Kafes Sistemlerin Genetik Algoritma ile Optimum Tasarımı, Yüksek Lisans Tezi, K.T.Ü., Fen Bilimleri Enstitüsü, Trabzon.
- Ayvaz, Y. ve Aydın, Z., 2000. 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.
- Cao, G., 1996. Optimized Design of Framed Structures Using a Genetic Algorithm, Ph. D. Dissertation, The University of Memphis.
- Chen, S. Y., 1997. Using Genetic Algorithms for The Optimal Design of Structural Systems, Ph. D. Dissertation, Arizona State University.
- Choi, C. K., Lee, D. G. ve Lee, H. W., 1991. Optimization for Large- Scale Steel Structures with Discrete Sections, Computers & Structures Vol.39, No. 5, pp. 547-556.
- Daloğlu, A. ve Armutçu, M., 1998. Genetik Algoritma ile Düzlem Çelik Çerçevelerin Optimum Tasarımı, İMO Teknik Dergi, 1601-1615.
- Emel, G. ve Taşkın, Ç., 2002. Genetik Algoritmalar ve Uygulama Alanları, Uludağ Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi Cilt XXI, Sayı 1, S. 29-152.
- Erbatur, F. ve Al-hussainy, M. M., 1992. Optimum Design of Frames, Computers & Structures Vol. 45, No. 5/6, pp. 887-891.
- Erbatur, F., Hasaıcebi, O., Tütüncü, İ. ve Kılıç, H., 2000. Optimal Design of Planar and Space Structures with Genetic Algorithms, Computers and Structures 75 pp. 209-224.
- Jenkins, W. M., 2002. A Decimal-Coded Evolutionary Algorithm for Constrained Optimization, Computer & Structures, pp. 471-480.
- Kılıç, M. H., 1997. Use of Genetic Algorithms in Frame Optimization ,M. Sc. Thesis, The Middle East Technical University, Ankara.
- Mahfouz, S. Y., 1999. Design Optimization of Structural Steelwork, Ph. D. Dissertation, Department of Civil Engineering University of Bradford.

Manuel of Steel Construction – Allowable Steel Design, 1989, 9th Ed., Am. Inst.of Steel Constr., Chicago, III.

Odabaşı, Y., 1997. Ahşap ve Çelik Yapı Elemanları, İstanbul.

Rasheed, K. M., 1998. GADO: A Genetic Algorithm for Continuous Design Optimization, Ph. D. Dissertation, The State University of New Jersey.

Saka, M. P., 1991. Optimum Design of Steel Frames with Stability Constraints, Computers & Structures Vol.41, No. 6, pp. 1365-1377.

Tezcan, S., 1970. Çubuk Sistemlerin Elektronik Hesap Makineleri ile Çözümü, İstanbul.

TS 648, 1982. Çelik Yapıların Hesap ve Yapım Kuralları, T.S.E., Ankara.

TS 910, 1982. Çelik I Profilleri Sıcak Haddelenmiş, T.S.E., Ankara.

Tütüncü, İ., 1995. Optimization of Steel Frames by Genetic Algorithms, M. Sc., Thesis, The Middle East Technical University, Ankara.

URL-1, http://cmm.info.nih.gov/intro_simulation/node27.html, Simulated Annealing, 26 Mayıs 2003.

URL-2, http://www.dai.ed.ac.uk/CVonline/LOCAL_COPIES/BMVA96Tut/node27.html, Simulated Annealing, 26 Mayıs 2003.

URL-3, http://www.ulib.org/webRoot/Books/Numerical_Recipes/bookcpdf/c10-9.pdf, Simulated Annealing Methods, 26 Mayıs 2003.

URL-4, http://www.epcc.ed.ac.uk/computing/training/document_archive/GAs-course/node9.html, Simulated Annealing, 26 Mayıs 2003.

URL-5, <http://www1.gantep.edu.tr/~dereli/turkce/ky.ppt>, Toplam Kalite Yönteminin Işığında Yapay Zekanın Endüstriyel Problemlerin Çözümünde Kullanımı, 26 Mayıs 2003.

URL-6, <http://members.tripod.com/~Bagem/bagem/index.html>, Yapay Zeka, 27 Mayıs 2003.

URL-7, <http://www.yapay-zeka.org>, Yapay Sinir Ağları, 27 Mayıs 2003.

URL-8, http://yapayzeka.hypermart.net/yz/uzman_sistemler.htm, Uzman Sistemlerin Temel Bileşenleri, 29 Mayıs 2003.

URL-9, http://www.mmo.org.tr/muhendismakina/arsiv/2001/ekim/Genetik_Algoritma.htm, Genetik Algoritma ve Uygulamaları, 2 Haziran 2003.

URL-10, http://www1.dogus.edu.tr/dogustru/journal/icindekiler_6.htm, Akıř Tipi Çizelgeleme Problemlerinin Genetik Algoritma Yardımı ile Çözümünde Uygun Çaprazlama Operatörünün Belirlenmesi, 2 Haziran 2003.

Yoshimoto, F., Harada, T. ve Yoshimoto, Y., 2003. Data Fitting with A Spline Using A Real-Coded Genetic Algorithm, Computer-Aided Design, Elsevier Science.

Zaduryan, O., 1999. U zay Ç erçevelerin B ilgisayar i le Ç özümlenmesi, Y üksek Lisans Tezi, İstanbul Üniversitesi Fen Bilimleri Enstitüsü, İstanbul.



ÖZGEÇMİŞ

Serkan BEKİROĞLU 1979 yılında Trabzon'da doğdu. İlkokulu 100.Yıl İlköğretim Okulu'nda, orta okulu Cumhuriyet Orta Okulu'nda, liseyi ise Trabzon Yorma Fen Lisesi'nde tamamladı. 1996-1997 öğretim yılında Karadeniz Teknik Üniversitesi Mühendislik-Mimarlık Fakültesi İnşaat Mühendisliği Bölümü'nde lisans öğrenimine başladı. İnşaat Mühendisliği Bölümü'nden bölüm ikincisi olarak mezun olup aynı yıl bu bölümde yüksek lisans eğitimine başladı. 2001-2002 güz döneminde Karadeniz Teknik Üniversitesi Fen Bilimleri Enstitüsü'nde araştırma görevlisi olarak göreve başladı. İngilizce bilen BEKİROĞLU halen bu görevini sürdürmektedir.

