

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

**DÜZLEMSEL ÇELİK ÇERÇEVELERİN GENETİK
ALGORİTMA İLE OPTİMİZASYONU**

**YÜKSEK LİSANS TEZİ
İnş. Müh. Tacettin ŞEKER**

Anabilim Dalı : İNŞAAT MÜHENDİSLİĞİ

Programı : DEPREM MÜHENDİSLİĞİ

HAZİRAN 2008

**DÜZLEMSEL ÇELİK ÇERÇEVELERİN GENETİK
ALGORİTMA İLE OPTİMİZASYONU**

**YÜKSEK LİSANS TEZİ
İnş. Müh. Tacettin ŞEKER
(501051224)**

**Tezin Enstitüye Verildiği Tarih : 5 Mayıs 2008
Tezin Savunulduğu Tarih : 10 Haziran 2008**

**Tez Danışmanı : Doç. Dr. Ünal ALDEMİR (İ.T.Ü.)
II. Tez Danışmanı : Y. Doç. Dr. Hilmi ÇOŞKUN (M.K.Ü.)
Diğer Jüri Üyeleri : Prof. Dr. Reha ERTAN (İ.T.Ü.)
Prof. Dr. R. Faruk YÜKSELER (Y.T.Ü.)**

HAZİRAN 2008

ÖNSÖZ

Düzlemsel çelik çerçevelerin Genetik Algoritma ile ağırlık optimizasyonunun anlatıldığı bu çalışmada bana yol gösteren ve tezimin her aşamasında yardım eden danışman hocalarım Doç. Dr. Ünal ALDEMİR ve Yrd. Doç. Dr. Hilmi ÇOŞKUN'a teşekkür ederim.

Program hazırlama konusunda önemli katkısı olan Araş. Gör. Murat YILMAZ, araştırma sırasında kişisel görüşme yaptığım fikir ve bilgileri ile farklı bakış açısı kazandıran Öğr. Gör. Dr. Şenay ATABAY'a, Araş. Gör. Ayça ALTAY'a ve Yrd. Doç. Dr. Barlas Özden ÇAĞLAYAN'a, matematiksel modelleme ile problemlere farklı yaklaşımlar kazandıran ve bilgileri ile yardımcı olan Prof. Dr. Zekai ŞEN'e teşekkür ederim. Yine eğitim sürecinde, konaklama ile ilgili katkılarından dolayı, Kemalettin Eröge Polis Eğitim Merkezi emekli müdürlerinden emekli müdür Hüseyin GÜMÜŞ'e ve şu anda müdür olarak görev yapan Dr. Halis BÖGÜRCÜ'ye ve diğer tüm çalışanlara, İ.T.Ü. yurt müdürlüğüne ve burs ofisi personellerine, eğitim süresince bana burs veren Deprem Vakfına ve İ.T.Ü. Fen Bilimleri Burs Ofisine teşekkür ederim. Bana her yönden destek çıkan Elek. Müh. Hakan ERSOY'a, İnş. Müh. Hasan Hüseyin DENİZ'e, İnş. Müh. Erdal SUAT'a, Mak. Müh. Gökhan DEVELİ'ye, Araş. Gör. Hasan Faik KARA'ya, teşekkür ederim. Yüksek lisans sürecinde tanışıp benim ilerlememe katkısı olan ve yalnız bırakmayan arkadaşlarım olan Ayçin ÇAKMAK, Ayşegül YILMAZ, Selen GÖGERÇİN, Serdar ARSLAN , Yunus ÇELİK ve Zeynep AYDIN'a minnettarım. Yine bana her yönden emeği geçen tüm hocam ve arkadaşlarıma sonsuz teşekkür ederim.

Ayrıca, her zaman beni destekleyen, yanımda olan anlayış ve sabır gösteren benim her şeyim olan aileme bu çalışmayı ithaf ediyorum.

Haziran, 2008

Tacettin ŞEKER

İÇİNDEKİLER

KISALTMALAR	v
TABLO LİSTESİ	vi
ŞEKİL LİSTESİ	vii
SEMBOL LİSTESİ	viii
ÖZET	ix
SUMMARY	xi

1. GİRİŞ	1
1.1 Konu	1
1.2 Yapı Sistemlerinde Genetik Algoritma ile Yapılan Optimizasyon Çalışmaları	2
1.3 Çalışmanın Amacı ve Kapsamı	8
2. OPTİMİZASYON	9
2.1 Optimizasyon Algoritmaları	9
2.1.1 Giriş	9
2.1.2 Tarihsel gelişim	10
2.1.3 Temel kavramlar ve tanımlar	11
2.1.4 Optimizasyon probleminin sınıflandırılması	14
2.1.5 Optimizasyon yöntemlerinin sınıflandırılması	16
2.2 Sezgisel Algoritmalar	18
2.2.1 Giriş	18
2.2.2 Sezgisel algoritmaların sınıflandırılması	19
2.2.2.1 Klasik sezgisel algoritmalar	19
2.2.2.1.1 Bölgesel Araştırma İniş Metotları	20
2.2.2.2 Modern sezgisel algoritmalar(Yapay zeka yöntemleri)	21
2.3 Yapı Tasarımında Optimizasyon	23
2.3.1 Optimizasyon probleminin tanımı	23
2.3.1.1 Amaç fonksiyonu	24
2.3.1.2 Tasarım değişkenleri	25
2.3.1.3 Kısıtlamalar	27
2.3.2 Yapı optimizasyonunda kullanılan yöntemler	29
2.3.2.1 Genel bilgi	29
2.3.2.2 Genetik algoritma	31
3. GENETİK ALGORİTMALAR	33
3.1 Evrimsel Hesaplama	33
3.2 Genetik Bilgi	34
3.3 Genetik Algoritmalar	37
3.3.1 Genetik algoritmanın tarihçesi	37

3.3.2 Genetik algoritmanın tanımı	37
3.3.3 Genetik algoritmanın faydaları ve mahsurları	41
3.3.4 Genetik algoritmanın operatörleri	42
3.3.4.1 Kromozomun şifrelenmesi	43
3.3.4.2 Eşleşme havuzu oluşturma	45
3.3.4.3 Kopyalama (Üretim)	45
3.3.4.4 Çaprazlama (Gen Takası)	46
3.3.4.5 Mutasyon	47
3.3.5 Genetik algoritmanın parametreleri	49
3.3.5.1 Çaprazlama olasılığı	49
3.3.5.2 Mutasyon olasılığı	50
3.3.5.3 Popülasyonun (Topluluğun) büyüklüğü	50
3.3.5.4 Seçim	51
3.3.5.5 Seçkinlik	53
3.3.6 Hesapta izlenen yol	53
3.3.6.1 Gen haritasının hazırlanması	54
3.3.6.2 Optimum tasarım probleminin tanımlanması	57
3.3.6.3 Genetik algoritmada tablo işlemleri	59
4. DÜZLEMSEL ÇELİK ÇERÇEVELERİN OPTİMUM TASARIMI	62
4.1 Optimum Tasarım Problemi	62
4.1.1 Tasarım değişkenleri	62
4.1.2 Amaç fonksiyonu	62
4.1.3 Kısıtlamalar	63
4.2 Hazırlanan Bilgisayar Programlarının Tanıtılması	64
5. SAYISAL ÖRNEK	68
5.1 Örnek 1	
6. SONUÇLAR VE ÖNERİLER	74
KAYNAKLAR	75
ÖZGEÇMİŞ	79

KISALTMALAR

BS	: Beton Sınıfı
GA	: Genetik Algoritma
TS	: Türk Standartları
TS 648	: Çelik Yapıların Hesap ve Yapım Kuralları
YZD	: Yapay Zeka Dalları
∇f	: Gradyent vektörü

TABLO LİSTESİ

	<u>Sayfa No</u>
Tablo 3.1 : Genetik algoritma için önerilen kontrol parametre değerleri.....	51
Tablo 3.2 : Araştırma uzayının belirlenmesi	54
Tablo 5.1 : Örnek 1 için araştırma uzayı.....	70
Tablo 5.2 : Araştırma uzayını oluşturan profillerin geometrik özellikleri	71
Tablo 5.3 : Araştırma uzayını oluşturan profillerin diğer parametreleri.....	71
Tablo 5.4 : Örnek 1 için son uygun nesil	72

ŞEKİL LİSTESİ

	<u>Sayfa No</u>
Şekil 2.1 : Optimizasyon akış şeması	23
Şekil 3.1 : Genetik algoritmalarda evrimleşme döngüleri	40
Şekil 3.2 : Genetik algoritmalarda akış şeması	42
Şekil 3.2 : Ağaç kodlama için kromozom örneği	45
Şekil 4.1 : DN.txt veri dosyası	65
Şekil 4.2 : EL.txt veri dosyası	66
Şekil 4.3 : EV.txt veri dosyası	66
Şekil 4.4 : pveri.txt veri dosyası	67
Şekil 5.1 : Örnek 1 için tek açıklıklı çelik çerçeveler	68
Şekil 5.2 : Nesil-başarısı	73

SEMBOLLER

A_k	: k grubuna ait elemanların kesit alanları.
C	: Toplam ihlal edilme katsayısı
$g_{i(x)}$: Sınırlayıcılar
$F(x)$: Sınırlayıcısız fonksiyon
K	: Sabit
m_k	: k grubundaki çubuk sayısı
ng	: Grup sayısı
p	: Sınırlanmış deplasman sayısı
$W(x)$: Amaç fonksiyonu
ρ_i	: Malzemenin yoğunluğu
δ_j, δ_i	: j noktasındaki deplasmanı ve onun üst sınırı.
σ	: Gerilme
σ_a	: Çeliğin akma gerilmesi

DÜZLEMSEL ÇELİK ÇERÇEVELERİN GENETİK ALGORİTMA İLE OPTİMİZASYONU

ÖZET

Bu çalışmada, düzlemsel çelik çerçevelerin Genetik Algoritma Yöntemi ile ağırlık optimizasyonu yapılmaktadır. Çerçeve elemanlarının kesit alanları tasarım değişkenleri olarak alınmakta ve yapının toplam ağırlığını minimize eden kesit alanları aranmaktadır. TS 648 yapı gerilme, stabilite ve izin verilen deplasman maksimum deplasman ile ilgili koşullar kısıtlama olarak alınmaktadır.

Düzlemsel çelik çerçevelerin ağırlığının minimizasyonu için optimizasyon yöntemi olarak Genetik Algoritma'nın kullanıldığı bu çalışmada, sayısal uygulamaların çözümü için, yönetmelikte verilen gerilme, deplasman ve diğer koşullar optimizasyon algoritmasına yerleştirilerek iki boyutlu çelik çerçeve sistemlerin tasarımı için gerekli olan kesit alanlarını belirleyen bir yazılım geliştirilmiştir.

Tez altı bölümden oluşmaktadır. Birinci bölümde tezin konusu, yapı sistemlerinin optimizasyonu için genetik algoritma ile yapılan çalışmalar açıklanmıştır. Bunlarla birlikte, hazırlanan tez çalışmasının amacı ve kapsamı anlatılmıştır.

Çalışmanın ikinci bölümünde genel optimizasyon probleminin temel kavramları tanıtarak, yaygın olarak kullanılan klasik optimizasyon algoritmalarından bazılarının temel adımları verilmektedir.

Üçüncü bölümde, evrime dayalı algoritmalara giriş yapıldıktan sonra bu sınıftan optimizasyon yöntemi olarak kullanılan Genetik Algoritma ve hesapta izlenen yol anlatılmıştır.

Dördüncü bölümde çelik yapılarla ilgili tezde kullanılan yönetmelik koşulları anlatılarak düzlemsel çelik çerçevelerin ağırlık optimizasyonunda izlenecek yol açıklanmıştır. Ayrıca bu bölümde, yöntemin sayısal uygulamalarında kullanılmak amacıyla MATLAB programlama dilinde kodlanan YZDOPT1 programı ve kullanılan analiz programı tanıtılmıştır.

Beşinci bölümde, hazırlanan program ile bir örnek çözülmüş ve sonuçları verilmiştir. Örnekte 3 elemanlı basit bir çerçeve dikkate alınmıştır.

Son bölümde de önerilen yöntemin verimliliği tartışılmıştır. Ayrıca bu konuda gereken daha ileri çalışmalardan söz edilmiştir.

OPTIMIZATION OF PLANE STEEL FRAMES BY GENETIC ALGORITHM

SUMMARY

In the present work, a weight optimization study has been realized for the plane steel frames by Genetic Algorithm Method. The cross-sections of frame elements have been considered as design variables and it is aimed at searching the optimum cross-sectional sizes that minimize total weight of plane steel frames. The formulation of the design problem considers the displacement, stress and stability constraints as they are defined in the design code TS 648.

Genetic Algorithm Approach has been used as an optimization method for the weight minimization of plane steel frames in this study. After setting the optimization algorithm with stress, displacement and the other constraints given in the code, a computer program is also developed for determining the optimum plane steel frames for the minimum weight design of two dimensional structural systems.

The thesis consists of six chapters. In the first chapter, a brief description of the study is given as well as some other earlier studies that are on the optimization of structural systems are summarized. The objective and the scope of the study are also presented in this chapter.

In Chapter Two; basic concepts of a general optimisation problem are described. Basis steps of some widely used classic optimization algorithms are described.

In chapter Three; introduction based on evolution algorithms is presented and then, Genetic Algorithm approaches that are in class of evolution algorithms and the main steps of the optimisation process for a general optimisation problem are presented.

In forth chapter; building code for structures are given and the computation principles of plane steel frames for weight optimization are described. The flowcharts and the details of the computer program developed for the numerical examples that are coded at MATLAB language are presented.

In the fifth chapter; example problems are solved by the developed program and the obtained results are given.

The discussion of the results dealing with the used optimization approach and the solved problems are presented in the last chapter. Finally, the efficiency of the proposed approach and the suggestions for further research work on the subject are mentioned.

1. GİRİŞ

1.1 Konu

Temel bilim dallarından elde edilen bilgileri işleyerek istenilen sistemin tasarım ve analizinin yapılması ve gerçekleştirilmesi mühendislik biliminin ilgi alanına girer. Bu alanlar arasında ilişkileri çok yönlü olup birbirini tamamlayıcı niteliktedir. İhtiyaçlardan doğan düşünce aşamasından başlayıp tasarım, analiz ve gerçekleştirme süreciyle de istenilen amaca ulaşılması hedeflenir. Ancak tasarlanan bir sistemin istekleri sağlayan çok sayıda muhtelif alternatif çözümler içerebilir. Bunlar içerisinde en iyisinin en kısa zamanda bulunması çok önemli bir mühendislik çalışmasıdır. Optimizasyon problemi olarak adlandırılan bu çalışma, tasarlanacak sistemden beklentiler artıkça daha karmaşık hale gelmektedir. Aynı şekilde problemin çözümü de o derece zorlaşmaktadır.

Optimizasyon problemlerinin çözümüyle ilgili çalışmalar tüm mühendislik alanlarını ilgilendirdiğinden optimizasyon, son derece önemli bir bilim alanı haline gelmiştir.

Zamanla optimizasyon problemlerinin çözümü için geliştirilen optimizasyon yöntemleri, matematiksel ifadelerle dayalı belirgin yöntemlerdir. Klasik yöntemler olarak isimlendirilen bu yöntemler problemlere özel olması, problemin kesinlikle matematiksel fonksiyonlarla tanımlanma gerekliliği gibi bir çok mahsuru vardır. Bu matematiksel programlama tekniklerinin çoğunda da tasarım değişkenlerinin sürekli olduğu kabul edilmekte olup, çeşitli mühendislik problemine uygulanmaz.

Optimizasyon alanında çalışan bazı bilim adamları genel amaçlı, esnek ve performansı yüksek yöntemler geliştirmek için dikkatlerini tabiatta var olan sistemlere ve bu sistemlerde ve aralarında cereyan eden olaylara yöneltmiştir. Tabiatta var olan sistemleri ve olayları temel alarak oluşturulan bu optimizasyon yöntemlerine sezgisel yöntem, zeki yöntem veya yapay zeka yöntemi gibi çeşitli adlar verilmiştir. Günümüzde bu algoritmalar mühendislik alanlarında çok sayıda karmaşık ve zor problemlerin çözümünde kullanılır hale gelmişlerdir.

Genetik Algoritma yöntemi, ayrık tasarım değişkenlerini kullanan yöntemlerden biridir.

Bu çalışmada, optimum yapı tasarımı probleminin çözümü için; ilk olarak Goldberg tarafından önerilen ve birçok optimizasyon probleminin çözümü için önerilen Genetik Algoritma kullanılmıştır (Goldberg, 1989). Genetik Algoritma, “doğada güçlü olanların yaşaması, zayıf olanların elenmesi” olarak bilinen Darwin’in teorisine dayanmaktadır. Bu yöntem, bütün canlılarda yaşanan evrim sürecinde etkilenecek geliştirilen, kopyalama, çoğalma ve çaprazlama operatörleri kullanılarak, araştırma uzayındaki yeni noktaların belirlendiği doğal seçme ve doğal gen mekaniğine dayanan bir araştırma yöntemidir.

Genetik Algoritma’ların temel prensibi, her adımda bir önceki nesilden yeni bireyler oluşturarak, amaç fonksiyonunun uygunluğunu artırmak ve sonuç olarak belli kısıtlamalar altında amaç fonksiyonunu sağlayan en uygun değeri elde edilmektedir.

1.2 Yapı Sistemlerinde Genetik Algoritma ile Yapılan Optimizasyon Çalışmaları

Yapı sistemlerin hesabında genetik algoritma ile yapılan optimizasyon çalışmalarından bazıları incelenerek, kronolojik sıra ile aşağıda özet halinde sunulmuştur:

Değertekin ve diğ. (2006) tarafından yapılan çalışmada ise, uzay çelik çerçevelerin tabu ve genetik algoritma yöntemleri ile optimum tasarımı için karşılaştırılması yapılmıştır.

Kömür ve Altan (2006)’ ın çalışmasında da Genetik Algoritma ile optimizasyon kavramı, betonarme bir giriş ve kolon kullanılarak açıklanmaya çalışılmıştır. Yapılan bu çalışma da literatürde yapılan birkaç çalışma ile de karşılaştırılması yapılmıştır.

Dominguez diğ. (2006) tarafından yapılan çalışmada büyük kafes yapıların eşzamanlı olarak boyut, geometri ve topoloji olarak Genetik Algoritma eniyilemesi sonlu elemanlarla da analizi yapılmıştır.

Jean-Denis ve diğ. (2005) tarafından yapılan çalışmada da bir metal yapıda kompozit bağ yamasının eniyilemesi için Genetik Algoritma uygulanması tanımlanmıştır. Bu çalışmada amaç bazı gizli bölgelerdeki kaplanılmayan yama ve

açıktaki yamaların maksimum yüzeyinde sınır şartları altında gerilme seviyesini azaltmaktır.

Kaveh ve Rahami (2005) tarafından yapılan bu çalışmada, kuvvet yönteminin enerji formülasyonu sunulmuş ve Genetik Algoritma kullanılarak analiz programı hazırlanmıştır. Burada belirli koşullar altında tasarım sonucunda yapının ağırlığının minimum durumu hesaplanmıştır.

Zolfaghari ve diğ. (2005), Zemin kayma stabilitesi analizinde deprem ve bina yükü etkisinin araştırılması için genetik algoritmalar başarıyla kullanmıştır.

Wang ve diğ.(2005), çevresel bina tasarımında optimal çözüme ulaşmak için genetik algoritmaları başarıyla uygulamışlardır.

Perng ve diğ.(2005), tarihi binaların onarımında optimal bütçe tasarımı için genetik algoritmaları uygulamışlardır. Uygulama sonunda genetik algoritma metodunun geleneksel metotlardan daha etkili ve ekonomik karar önerdiğini tespit etmişlerdir.

Sahab ve diğ.(2005a), yapılan çalışmada binalarda düz plaka betonarme betonu için genetik algoritma değişimine dayanarak iki aşamalı melez optimizasyon algoritması kullanmışlardır. İlk aşamada yüzey araştırma tasarımı, ikinci aşamada Jeeves ve Hooke metotlarının genetik algoritma ile çözümü kritiğe tabi tutulmuştur. Sonuç olarak optimum çözüm için genetik algoritma ile çözümden önce yeterli miktarda değerlendirme fonksiyonu sağlanması gerektiğine karar verilmiştir.

Sahab ve diğ.(2005b), binadaki her bir kolon ve döşeme boyutlarının bulunması ve betonarme plak betonu için optimum maliyetin hesaplanmasında başarılı şekilde melez optimizasyon algoritmalarını kullanmışlardır.

Govindaraj ve diğ.(2005), Hindistan standartları şartnamelerini esas alarak sürekli giriş betonarme betonunun optimum detaylı tasarımı için genetik algoritmaları uygulamış rasyonel, güvenilir, ekonomik ve kullanışlı tasarım elde etmişlerdir.

Fairbairn ve diğ.(2004), kütle beton yapılarının inşasında (malzeme cinsi, ortam sıcaklığı, pompa yüksekliği vb. kriterleri kullanarak) optimum inşaat maliyetini elde etmek için genetik algoritmaları başarıyla uygulamışlardır.

Atabay ve Gülay (2004)'in çalışmasında ise, perdeli betonarme yapı sistemlerinin Genetik Algoritma Yöntemi ile maliyet optimizasyonu yapılmıştır. Tasarım değişkenleri perde boyutları olup yapının toplam malzeme maliyetini minimize eden

perde boyutları aranmaktadır. “Afet Bölgelerinde Yapılacak Yapılar Hakkında Yönetmelik” ve “TS 500” de perde boyutları ve donatı ile ilgili koşullar problemin kısıtlamaları olarak alınmıştır. Sayısal uygulamalarda kullanılmak üzere, yapısal sistemlerin minimum maliyet tasarımı için gerekli olan optimum perde boyutlarını belirleyen yazılım geliştirilmiştir.

Park ve diğ. (2003)'in çalışmasında, reçine aktararak kalıplama yöntemiyle üretilen kompozit tabakalandırılmış plakların ağırlık minimizasyonu yapılmıştır. Yöntemin uygulanmasında rijitlik kısıtlamaları veya mukavemet kısıtlamaları kullanılabilir.

Toğan ve Daloğlu (2003a), genetik algoritma kullanarak, çeşitli paralel başlıklı düzlem kafes sistemlerin boyut ve şekil optimizasyonlarını gerçekleştirmiş ve sonuçta sistem ağırlığının minimize edilebileceğini göstermişlerdir.

Bekiroğlu ve diğ.(2003), çalışmalarında genetik algoritmada değer kodlaması kullanarak kafes sistemlerinin minimum ağırlıklı tasarımını amaçlamışlar ve değer kodlaması kullanılarak hazırlanan programın kafes sistemlerin genetik algoritma ile minimum ağırlıklı olarak boyutlandırılmasında etkin bir şekilde kullanıldığını göstermişlerdir.

Toğan ve Daloğlu (2003b) yaptığı çalışmada genetik algoritma kullanarak çatı makaslarının farklı yüklemeler altında optimum tasarımlarını yapmıştır. Gerilme, burkulma ve çökme sınırlayıcılar göz önüne alınmış ve iki tip tasarım problemi incelemiştir. Önce çeşitli çatı makaslarının minimum ağırlıklı boyut optimizasyonları gerçekleştirilmiş daha sonra çatı makaslarının eleman kesit alanlarıyla birlikte optimum geometrilerini de bulacak şekilde boyut ve şekil optimizasyonları gerçekleştirilmiştir.

Torregosa and Kanok-Nukulchai (2002)'in çalışmasında ise düzlem çelik çerçeveler farklı yükleme durumunda ağırlık optimizasyonu için Genetik Algoritma geliştirilmiştir. Buradan elde edilen sonuçlar diğer optimizasyon yöntemleri ile de karşılaştırılmıştır.

Lagaros ve diğ.(2002), evrimsel hesaplama tekniğinin bir parçasını oluşturan genetik algoritmalar ve evrim stratejilerini, büyük ölçekli yapısal boyutlandırma problemlerinde kullanarak şekil ve boyut optimizasyonlarını gerçekleştirmişlerdir.

Paki ve Abdussamet (2001) tarafından yapılan bir çalışmada ise, altı açıklıklı sürekli bir kirişte maksimum açıklık ve mesnet momentlerini veren hareketli yük kombinezonlarının Genetik Algoritmalarla otomatik bir şekilde düzenlenmesi incelenmiştir. Bu çalışmada geliştirilen algoritmanın hızının diğer tesir çizgileri yöntemi gibi klasik yöntemlerle karşılaştırılması da yapılmıştır.

Kalyanmoy ve Surendra (2001)'in yaptığı çalışmada kafes yapılarının optimizasyonunda minimum ağırlığı elde etmede 2 ve 3-boyutlu kafeslerin kesit alanı, topoloji ve konfigürasyonu bulmak için gerçek kodlu genetik algoritmanın kullanımı uygulanmıştır.

Çağdaş (2001) çalışmasında, betonarme döşemelerin akma çizgisi yöntemiyle çözümünde akma çizgisi deseninin belirlenmesi için genetik algoritma tekniğini uygulamıştır. Alışılmış döşemelerin yanı sıra değişik geometrilere sahip döşemeler, değişik mesnetlenme ve yükleme koşullarında analiz edilmiştir.

Kaya (2001) çalışmasında, yüksek kirişlerde donatı çaplarının optimum tasarımını gerçekleştirmek üzere kesit özellikleri pratikte hazır olan standart kesitlerden seçebilen bir genetik yaklaşım sunulmaktadır. Değerlendirme, seçim, kopyalama, çaprazlama, ve mutasyon operatörlerinden oluşan genetik algoritma kullanılarak yüksek kiriş elemanları için üzerlerine etki eden yükleri belli bir emniyetle taşıyacak şekilde yatay ve dikey ortogonal donatı çapları belirlenmiştir. Yüksek kirişlerin ACI' da belirtilen koşullara göre analizi yapılarak ACI' nin yüksek kirişlerin analizinde uygulamada karşılaşılan güçlükler ve sınırlamalar ortaya konmuştur.

Aynı problemlerin geliştirilen bilgisayar programı ile de analizi yapılarak bu programın ne derece kullanışlı olduğu ortaya konmuştur.

Raich ve Ghaboussi (2000) tarafından yapılan çalışmada etkili tasarım topolojisi ve geometrisine ilave olarak seçilen yapısal elemanların özelliklerinin dışında önemli olan kavramsal tasarım süresince maliyet durumunun birleşimi geliştirilmiştir. Bunun için evrimsel hesaba dayanan bir algoritma geliştirilmiş olup çerçeve topolojilerinin değişik evrimselleşmesi Genetik Algoritma ile sağlanmıştır.

Pezesk ve diğ. (2000)' nin çalışmasında, geometri bakımından doğrusamaz (doğrusal olmayan) çelik çerçeve yapılan iki boyutlu tasarımı için genetik algoritma ile optimizasyon yöntemi geliştirilmiştir. Bu çalışmada, grup seçme mekanizması

kullanılarak, geliştirilmiş, uygulanabilir çaprazlama operatörü ele alınmış ve ceza fonksiyonu seçimi için çeşitli önerilerde bulunulmuştur.

Hadi ve Arfiadi (1998)' nin araştırmasında, deprem etkisindeki, çok serbestlik dereceli yapılar için kütle salınımını sönmüleyen bir aracın parametreleri genetik algoritma ile eniyilemesi yapılmıştır.

Saka (1998) tarafından yapılan çalışmada, eleman kesit özelliklerini uluslar arası standart kesitlerden alan ızgara sistemlerin optimum tasarımı için bir Genetik Algoritma programı hazırlanmıştır. Tasarım probleminde sehim sınırları ve gerilme kısıtlamaları hesaba katılmıştır.

Rafiq ve Southcombe (1998) tarafından yapılan çalışmada da optimal tasarım için yeni bir yaklaşım ile Genetik Algoritma kullanılarak iki eksenli eğilme altında betonarme kolonlar için ayrıntılı bir çözüm yapılmıştır. Tasarım kısıtlamaları olarak kesit boyutları, aksenal yük ve kolonun her iki eksenini doğrultusundaki eğilme hesaba katılmıştır. Bu çalışmada, tasarım kısıtlamaları ile iki eksenli bir kolon için optimal donatı çubukları ve bu donatıların yerleşimlerinin bulunmasında Genetik Algoritma'nın uygulanması anlatılmıştır. Maksimum eğilme kapasitesini aşmayacak şekilde ve donatı alanının minimize edildiği kısıtlamalar altında en ekonomik çözüm aranmıştır. Kolon kesitindeki donatı dağılımı bulunurken British Standart (BS 8110) kullanılmıştır. Kırılma durumu (ULS) ve inşa edilebilirlik konularının her ikisi de hesaplarda dikkate alınmıştır.

Daloğlu ve Armutçu (1998)' in çalışmasında, düzlem çelik çerçevelerin optimum tasarımını gerçekleştirmek üzere kesit özelliklerini uygulamada hazır olan standart kesitlerden seçebilen bir genetik yaklaşım sunulmuştur. Kopyalama, çaprazlama ve değişim operatörlerinden ibaret olan basit genetik algoritma hazırlanmış ve yapı, gerilme, stabilite ve deplasman sınırlayıcıları altında boyutlandırılmıştır. Bileşik gerilme sınırlayıcısı, TS 648' e göre dikkate alınmıştır. Çeşitli sayısal örnekler yapılmıştır.

McMahon ve diğ. (1998)' in yapmış olduğu çalışmada ise kompozit lamine yapı tasarımı için Genetik Algoritma hazırlanmıştır.

Camp ve diğ. (1998) tarafından yapılan araştırmada da iki boyutlu çelik yapıların ayrık optimizasyonu için birleştirilmiş bir basit Genetik Algoritma programı hazırlanmıştır. Amaç fonksiyonu yapının toplam ağırlığının güvenilirlik ve

kullanışlılık esaslarına dayalı olarak minimizasyonu amaçlanmıştır. Genetik Algoritma esaslı FEAPGEN yöntemi, Sonlu Elemanlar Analiz programına (FEAP) ek olarak geliştirilmiştir. FEAPGEN; ayrık tasarım değişkenlerini içermektedir ve (AISC-ASD) kullanılarak tasarım kontrolü yapılmıştır.

Friswell ve diğ. (1998)'in yaptıkları çalışmada ise, titreşim verileri kullanılarak ortaya çıkan hasar problemine Genetik Algoritma'nın uygulanması gösterilmiştir. Amaç bir yapıda bir veya daha fazla bölgede meydana gelen hasarları teşhis etmek ve bu hasarların büyüklüklerini tahmin etmektir. Genetik Algoritma, ayrık hasar yerleri değişkenlerinin optimizasyonu için kullanılmaktadır. Bu yöntemler, bir giriş ve bir plak üzerinde uygulanarak açıklanmıştır.

Daloğlu ve Armutçu (1998) yapılan çalışmada, düzlem çelik çerçevelerin optimum tasarımını gerçekleştirmek üzere kesit özelliklerini pratikte hazır olan standart kesitlerden seçebilen bir genetik yaklaşım sunulmaktadır.

Kopyalama, çaprazlama ve değişim operatörlerinden ibaret olan basit genetik algoritma kullanılmış ve yapı gerilme, stabilite ve izin verilebilen maksimum deplasman sınırlayıcıları altında boyutlandırılmıştır. Bileşik gerilme sınırlayıcısı TS 648'e göre dikkate alınmıştır. Yöntemin uygulanabilirliğini göstermek üzere çeşitli sayısal örnekler verilmiştir. Karşılaştırma yapılabilmesi amacıyla örnekler literatürden seçilmiştir.

Wei ve diğ. (1997)'nin yaptıkları bir çalışmada ise zaman-maliyet iş optimizasyonu için Genetik Algoritma'nın ilkelerine dayalı bir algoritma oluşturulmuştur ve bu algoritma ile bir yazılım geliştirilmiştir.

Cheng ve Li (1997)'nin yaptığı çalışmada ise Pareto Genetik Algoritma ve Bulanık Ceza Fonksiyon yönteminin birleştirilmesiyle kısıtlanmış çok amaçlı optimizasyon yöntemi anlatılmıştır. Pareto Genetik Algoritma, beş temel operatörlerden meydana gelmektedir. Üreme, çaprazlama, dönüştürme, uygun yer ve Pareto yerleşim filtresi. Ceza fonksiyonları kullanılarak kısıtlanmış problem kısıtlanmamış probleme dönüştürülmüştür.

Daloğlu ve Armutçu (1997)'nin yaptığı çalışmada da kafes sistemlerin optimum tasarımı için bir algoritma sunulmuştur. Gerilme, deplasman ve stabilite sınırlayıcıları TS 648'e uygun olarak göz önüne alınmaktadır.

Turgut ve diğ. (1997) tarafından yapılan çalışmada ise, tasarım değişkenleri sınırlı yapı sistemlerinin optimizasyonu için basit bir Genetik Algoritma hazırlanmıştır. Basit mesnetli bir betonarme giriş üzerinde ayrıntılı olarak Genetik Algoritma kullanılarak en iyilemesi yapılmıştır.

Ramasamy ve Rajasekaran (1996) tarafından yapılan çalışmada, endüstriyel çatıların tasarımına bir uzman sistem uygulanmıştır. Tasarım yapılırken bu konuda bilinen bütün şartlar hesaba katılmıştır.

Rajeev ve Krishnamoorthy (1992)' in yaptığı çalışmada ise ayrı tasarım değişkenli yapısal sistemlerin optimizasyonu için basit bir genetik algoritma geliştirilmiştir. Kısıtlanmış problemlerin kısıtlanmamış probleme dönüştürülme işleminde ceza esaslı dönüştürme yöntemi kullanılmıştır.

Jenkins (1992) tarafından yapılan çalışmada, düzlem çerçeve olarak sınıflandırılan yapıların optimum tasarımı için bir algoritma geliştirilmiştir. Tasarım optimizasyonu Genetik Algoritma ile yapılmakta, tasarım kısıtlamaları ceza fonksiyonları ile değiştirilerek amaç fonksiyonu için uygun değer bulunmaya çalışılmıştır.

1.3 Çalışmanın Amacı ve Kapsamı

Kaynak özetlerinden de görüldüğü gibi Genetik Algoritma ile yapısal optimizasyon konusunda bir çok çalışma yapılmıştır. Özellikle, çelik yapılar konusunda yapılmış bir çok çalışma mevcuttur. Ancak bu çalışmalar belli tasarımlar çerçevesinde yoğunlaşmış genel bir çalışma hedeflenilmemiştir.

Bu çalışmanın amacı, genel bir yapısal optimizasyon için bir program geliştirilmesidir. Bundan sonraki çalışmalarda ise bu yazılımın genelleştirilmesine ve çeşitli yapı sistemlerinde uygulanmasına gidilecektir.

Düzlemsel çelik çerçevelerin ağırlık optimizasyonunda tasarım değişkenleri olarak elemanların kesit alanları ve amaç fonksiyonu olarak ağırlık fonksiyonu alınmıştır. Optimizasyon probleminde kısıtlama olarak da çelik yapıların yönetmelik ve davranış koşulları göz önüne alınmış ve bir düzlem çerçeve örneğinde genetik algoritma çalışması izlenmiştir.

Ağırlık minimizasyonu yapılırken, 1980 yılında yürürlüğe giren TS 648 de birleşik gerilme sınırlayıcıları dikkate alınmıştır.

2. OPTİMİZASYON

2.1 Optimizasyon Algoritmaları

Bu bölümde optimizasyon diğer bir deyişle en iyileme problemleri ve bunların çözümleri için geliştirilen teknikler ile ilgili genel bilgiler verilecektir.

2.1.1 Giriş

Varlıkların kendilerinde meydana gelen olayları ve varlıklar arasındaki ilişkileri inceleyerek anlamak ve bunları bilgi formuna dökmek kimya, biyoloji, fizik ve astronomi gibi Temel Bilim Dalları'nın konusudur. Bu bilim dalları tarafından elde edilen bilgilerden faydalanarak, ihtiyaçlarımıza uygun şekilde görev yapacak varlıkların (cisimlerin veya ürünlerin diğer bir deyişle sistemlerin) tasarlanması ve gerçekleştirilmesi ise Mühendislik Bilim Dalları'nın görevidir. Var olan veya tasarlanan yani sonradan üretilen bir sistemin belirli şartlarda nasıl davranış sergilediğinin, beklenen davranışı gösterip göstermediğinin incelenmesi Analiz olarak adlandırılır. Mühendislikte yapılan bilimsel çalışmaları genel olarak iki ana gruba ayırabiliriz:

- 1) Tasarım ve
- 2) Analiz

Tasarlanan bir sistemin istekleri sağlayan çok sayıda muhtemel alternatif çözümler içerisinde en iyisinin (optimum) en kısa zamanda bulunması bir eniyileme (optimizasyon) problemi olup çok önemli bir mühendislik çalışmasıdır. Eniyileme problemi olarak adlandırılan bu çalışma, tasarlanacak sistemden beklentiler arttıkça problemin çözümü o derece zorlaşmaktadır(**Karaboğa,2004**).

Eniyileme problemlerinin çözümü ile ilgili bilimsel çalışmalar mühendislik alanlarını ilgilendirdiğinden önemli bir bilim haline gelmiştir. Bu tür problemlerinin çözümü için geliştirilen çoğu yöntemler, çeşitli kabullere dayalı matematiksel ifadelerle bağlı belirgin yöntemlerdir.

Klasik yöntemler olarak adlandırılan bu yöntemlerin problemlere özel olması, matematiksel bağıntılarla tanımlanma zorunluluğu gibi mahsurları vardır.

Son zamanlarda en iyileme ile ilgilenen bazı bilim adamları genel amaçlı ve performansı yüksek yöntemler geliştirmek için ilgilerini doğal sistemlere ve bu sistemlerle etkileşim içerisinde olan olaylara yöneltmişlerdir. Doğadaki sistemleri ve olayları temel alarak oluşturulan diğer en iyileme yöntemleri de sezgisel yöntemler olarak adlandırılır. Bu yöntemler farklı mühendislik dallarında birçok karmaşık problemlerin çözümünde kullanılır hale gelmiştir.

2.1.2 Tarihsel Gelişim

Belli sınırlamalar altında en uygun çözümü bulma tekniği olarak tanımlanan eniyilemenin insanlık tarihi ile birlikte doğduğu ve karar vermede güçlü bir araç olarak kullanıldığı yapılan araştırmalar ile ortaya çıkmıştır. İnsanoğlunun, günümüzde olduğu kadar geçmişte de çalıştırdığı karar mekanizmasının altında, belirlenen amacı en uygun şartlarda elde etme ve bu amaca ulaştırılan yollar içinden en iyisini seçme eğiliminin varlığı görülmektedir.

17. yüzyılda Newton ve Leibnitz tarafından matematiksel analizin kurulması ile **(Bell, 1940)** eniyileme teorisinin de temeli bilimsel olarak atılmıştır. Eniyileme, matematiksel bir temele oturtulduğu 17. yüzyıla kadar sezgi ve tecrübeye dayandırılarak problemlerin çözümüne uygulanmaktaydı.

Eniyileme, Gauss tarafından geliştirilen ve “En Küçük Kareler Metodu” adı verilen, deneysel verilere eğri uydurma tekniğinde de kullanılmıştır. Regresyon analizi adı verilen söz konusu eğri uydurma tekniğinde, model eğrisinin katsayıları, deney sonucunda bulunan noktaların sapma miktarlarının karelerinin toplamı minimum olacak şekilde tayin edilmektedir.

17. yüzyılda matematiksel analizin kurulması, eniyileme teorisi ile ilgili bilimsel çalışmaların yapılmasına neden olmuştur. Söz konusu gelişimi takip eden yıllarda, eniyileme problemleri, genellikle farklı değişkenlerle tanımlanan bir amaç fonksiyonu ile sınırlayıcı şartları belirleyen eşitlik ve eşitsizliklerle ifade edilerek çözülmeye başlanmıştır. Ancak, analitik eniyileme yöntemleri, teorik yapısından gelen birtakım güçlükler nedeniyle, son derece sınırlı bir uygulama alanı bulabilmiştir.

2.1.3 Temel Kavramlar ve Tanımlar

Eniyileme problemi, belirli sınırlamaları sağlayacak şekilde, bilinmeyen parametre değerlerinin bulunmasını içeren herhangi bir probleme verilen bir addır.

Eniyileme işleminde;

1) İlk adım olarak karar parametreleri (karar değişkenleri = tasarım parametreleri) olarak adlandırılan parametre setinin tanımlanması gerekir.

2) Sonra, bu parametrelere bağlı olarak;

- en küçük yapılacak (minimize edilecek) bir maliyet fonksiyonu veya
- en büyük yapılacak (maksimize edilecek) bir kar fonksiyonu ve
- problemle ilgili sınırlama fonksiyonları

tanımlanmalıdır.

Maliyet fonksiyonu, daha iyi çözümü temsil eden parametre değerlerinin kullanılması durumunda daha düşük bir nümerik değer üretirken; kar fonksiyonu ise daha yüksek bir nümerik değer üretmektedir.

Sınırlamalar, parametrelerinin alamayacağı değerleri tanımlamakta ve karar parametrelerine bağlı olarak ifade edilmektedir. Bazı sınırlamalar eşitsizlikler, bazıları ise eşitlikler biçiminde olabilir.

Maliyet fonksiyonu ve sınırlamalar matematiksel şekli aşağıdaki gibi tanımlanabilir:

n-değişkenli $x = (x_1, x_2, x_3, x_4, \dots, x_n)$ vektörünü tanımlayalım. Burada x_i , i. parametrenin değerini göstermektedir.

Mâliyet fonksiyonu $f(x)$,

$$f(x) = f(x_1, x_2, x_3, x_4, \dots, x_n) \quad (2.1)$$

aşağıdaki gibi tanımlanan p-tane eşitlik sınırlamalarına

$$h_j(x) = h_j(x_1, x_2, x_3, x_4, \dots, x_n) = 0, \quad 1 \leq j \leq p \quad (2.2)$$

ve yine aşağıdaki gibi tanımlanan m tane eşitsizlik sınırlamalarına

$$g_i(x) = g_i(x_1, x_2, x_3, x_4, \dots, x_n) \leq 0, \quad 1 \leq i \leq m \quad (2.3)$$

sahip olabilir.

Problem için sınırlamaları sağlayan mümkün tüm çözümlerin oluşturduğu bölge, araştırma yapılabilecek uygun çözüm bölgesi olarak isimlendirilir. Bazı problemlerde birden fazla maliyet fonksiyonu bulunabilir. Yani, problem birden fazla enküçük yapılacak fonksiyon içerebilir. Bu tür problem, çok amaçlı optimizasyon problemi olarak isimlendirilir.

En iyi (optimum) çözüm,

- en küçük yapılacak problem durumunda uygun bölgede en düşük maliyet değerine sahip çözümken,
- en büyük yapılacak problem durumunda ise enbüyük amaç fonksiyon değerine sahip çözümdür.

$f(x)$ maliyet fonksiyonu, $x^* = (x_1^*, x_2^*, x_3^*, x_4^*, \dots, x_n^*)$ çözümü için

$$f(x^*) \leq f(x) \quad (2.4a)$$

denklem (2.4a) ile verilen şartı, uygun bölge (S) içindeki tüm muhtemel çözümler (x) için sağlıyor ise $f(x)$ fonksiyonu x^* 'de mutlak (küresel) enküçüğe (minimuma) sahiptir denir. Eğer denklem (2.4a) uygun bölge içerisinde, x^* 'in dar bir komşuluğu (N) içindeki tüm muhtemel çözümler için sağlanıyorsa $f(x)$ fonksiyonu x^* 'de yerel (bölgesel) enküçüğe (minimuma) sahiptir denir.

x^* 'in komşuluğu (N), küçük bir değere sahip olan δ parametresine bağlı olarak

$$N = \{ x \mid x \in S \text{ ve } \|x - x^*\| \leq \delta \} \quad (2.4b)$$

şeklinde tanımlanır. Bu, geometrik olarak x^* noktasını ihtiva eden küçük uygun bir bölgeyi tanımlamaktadır.

$f(x)$ fonksiyonun x^* noktasındaki x_1 'e göre kısmi türevi

$$\left. \frac{\partial f(x)}{\partial f(x_1)} \right|_{x=x^*}$$

şeklinde tanımlanır. Eğer tüm kısmi türevler bir sütun vektörü formunda düzenlenirse, vektör gradyent vektörü olarak adlandırılır ve ∇f veya grad f ile gösterilir.

Geometrik olarak gradyent vektörü, x^* noktasına teğet olan $f(x)$ sabit düzlemine diktir ve $f(x)$ fonksiyonunun en hızlı artış gösterdiği yönü belirtmektedir. Gradyent vektörünün bu özelliği, klasik gradyente dayalı en iyileme algoritmalarının temelini teşkil etmektedir.

Gradyent vektörünün ikinci kez türevini almak suretiyle klasik optimizasyonda önemli bir yeri olan Hessian matris H elde edilir. Dolayısıyla bu matris, $f(x)$ fonksiyonun ikinci dereceden kısmi türevlerinden oluşmaktadır. Hessian matris, simetrik özelliğe sahiptir ve optimallik için yeterli şartların tayininde önemli rol oynar.

Optimum noktada sağlanması gereken şartlar, gerek şartlar olarak adlandırılır. Gerek şartları sağlayamayan noktalar optimum olamaz. Ancak, gerek şartların sağlanması tek başına o noktanın optimum olmasını garanti edemez. Bu yüzden gerek şartları sağlayan optimum ve optimum olmayan noktaları ayırt etmek için yeter şartlar kullanılır. Gerek şartlar sağlandıktan sonra yeter şartlar da sağlanıyorsa o noktanın optimum nokta olduğu söylenir.

x^* , $f(x)$ fonksiyonun bölgesel minimumu olan bir nokta ve x, x^* 'ın yakın civarında diğer bir nokta olsun. Bu durumda, çözümlerin x^* civarındaki değişimi d ve bundan dolayı amaç fonksiyonunda ortaya çıkan $f(x^*)$ civarındaki değişimi Δf sırasıyla,

$$d = x - x^* \quad \text{ve} \quad \Delta f = f(x) - f(x^*)$$

olarak tanımlanabilir. x^* bölgesel minimum olduğu için yakın civarında, yani küçük d değerleri için

$$\Delta f = f(x) - f(x^*) \geq 0$$

eşitsizliğini sağlaması gerekir. Bu eşitsizlik, bölgesel minimum için gerek ve yeter şartların türetilmesi amacıyla kullanılır. d oldukça küçük olduğundan Δf , x^* noktasında Taylor serisine açılarak optimallik şartları türetilir.

Tek değişkenli fonksiyon için birinci dereceden gerek şart

$$f'(x^*) = 0 \tag{2.5a}$$

dır. Ancak bu şartı hem maksimumlar hem de minimumlar, yani değişimin sıfır olduğu durağan noktalar sağlamaktadır.

Minimumluk için yeter şart

$$f''(x^*) > 0 \quad (2.5b)$$

dir. Eğer $f''(x^*) = 0$, ise, x^* 'in bir minimum nokta olmadığı sonucuna varılamaz. Bu durumda daha yüksek dereceden türevlerin hesaplanması gerekir.

Tek değişkenli fonksiyon için ikinci dereceden gerek şart ise

$$f''(x^*) \geq 0$$

ile tanımlanır ve $f''(x^*) < 0$ şartını sağlayan herhangi bir nokta bölgesel minimum olamaz. $f''(x^*) = 0$ ise, bu noktanın bölgesel minimum olabilmesi için

$$f'''(x^*) = 0 \quad (2.5c)$$

gerekli şart ile

$$f''''(x^*) > 0 \quad (2.5d)$$

yeterli şartını sağlaması gerekir. Bu türev sıfıra eşit ise yine şüpheli durum söz konusudur ve bir üst dereceden türeve bakılmalıdır.

Sonuç olarak çift-derece türev sıfırdan küçükse aday nokta kesinlikle bölgesel minimum şartını sağlamıyor demektir ama sıfıra eşitse bir üst tek-dereceli türevin sıfıra eşit ve bir üst çift-derece türevin ise sıfırdan büyük olup olmadığı kontrol edilmelidir.

2.1.4 Optimizasyon Probleminin Sınıflandırılması

- 1) $f(x)$ amaç fonksiyonunun, x ile ilgili herhangi bir sınırlama olmaksızın minimizasyonu veya maksimizasyonu sınırlamasız optimizasyon; x ile ilgili sınırlamanın veya sınırlamaların bulunduğu optimizasyon problemi ise sınırlamalı optimizasyon problemi olarak isimlendirilir.
- 2) Bir optimizasyon problemi, doğrusal (lineer) amaç ve sınırlama fonksiyonlarına sahip ise bu problem lineer programlama problemi; bu fonksiyonlardan herhangi biri veya bütünü doğrusamaz (eğrisel = nonlinear) ise nonlinear programlama problemi olarak adlandırılır.

Nonlinear programlama [NLP] kavramı, nonlinear problemleri çözmek amacıyla geliştirilen nümerik yöntemler için kullanılmaktadır. Bu tür problemler için nümerik metotlara uygulanmasının zor olması hatta bazen mümkün olmamasıdır. Analitik yöntemlerde, gerekli şartlar yazılır ve bunlar aday bölgesel minimum noktalar için çözülür.

Nonlinear programlamaya ihtiyaç duyulmasının sebepleri olarak şunlar söylenebilir(**Arora, J.S. (1989) Introction to Optimum Design, McGraw-Hill**).

- a) Tasarım değişkenlerinin ve sınırlamaların sayısının çok fazla olması durumunda, oldukça fazla sayıda denklem oluşabilir ve dolayısıyla gerekli şartların çözülmesi çok zor olabilir. Bu durum özellikle sınırlamalı optimizasyon problemlerinde ortaya çıkmaktadır.
- b) Tasarım probleminin fonksiyonları yüksek seviyede nonlinearlik içerebilir. Bu yüzden problemin boyutu küçük olsa bile gerekli şartlar da oldukça nonlinear yapıda bulunabilir.
- c) Çoğu mühendislik uygulamalarında fonksiyonlar (maliyet ve sınırlama) tasarım değişkenleri açısından açık olamayabilir yani bağıntılar net şekilde kurulamayabilir.

Bu yüzden mühendislik sistemlerin optimum tasarımı için sistematik nümerik yaklaşımlar geliştirilmiştir. Bu tür yaklaşımlarda bir başlangıç çözümü alınır ve bu çözüm iteratif olarak optimallik şartları sağlanıncaya kadar geliştirilir.

- 3) Değişkenleri negatif olmayan tamsayı değerleri alan bir doğrusal programlama problemi tamsayı programlama problemi; değilse tamsayı olmayan programlama problemi şeklinde tanımlanır.
- 4) Diğer bir gruptandırma ise ayrık niceliklerin optimal olarak düzenlenmesi, grublanması, sıraya konulması veya seçilmesi problemi ayrık optimizasyon problemi olarak adlandırılır. Tasarım değişkenlerinin veya parametrelerinin alacağı değerler sürkli değerler ise bu tür problemlere sürekli optimizasyon problemi denir.
- 5) Yine, quadratik amaç fonksiyonuna ve doğrusal sınırlama fonksiyonlarına sahip bir problem quadratik programlama problemi olarak isimlendirilir.

2.1.5 Optimizasyon Yöntemlerinin Sınıflandırılması

En iyileme problemlerinde olduğu gibi, bu problemleri çözmek için geliştirilen yöntemler de çeşitli durumlara göre sınıflandırılmaktadır.

A) Genel olarak klasik en iyileme yöntemleri iki gruba ayrılır:

1. Optimalite kriterlerine dayalı dolaylı yöntemler:

Optimallik kriterleri, bir fonksiyonun minimum noktalarının sağlanması gereken şartlardır. Optimallik şartlarına göre çözüm arayan optimizasyon metotları genellikle dolaylı metotlar olarak adlandırılır. Önce gerek ve yeter şartlar yazılır, sonra bu şartlar bölgesel minimumluğa aday noktalar için çözülür.

2. Doğrudan yöntemler (araştırma metodları):

Araştırma metodları, biraz farklı bir fikre dayanır. Tahmini bir başlangıç çözümü ile araştırmaya başlanır ve başlangıç çözümü genellikle optimallik şartlarını sağlayamayacağından bu şartlar sağlanana kadar başlangıç çözümü. algoritma tarafından tekrarlı (iteratif) olarak geliştirilir. Yani bu tür yaklaşımda optimum çözümleri bulmak amacıyla çözüm uzayı araştırılır. Burada araştırma işlemi iki alt işlemden oluşur. Birincisi araştırma yönünün tayini ve bu belirlendikten sonra ikincisi adım uzunluğunun (büyüklüğünün) belirmesidir. Adım büyüklüğünün belirlenmesi için kullanılan metotlar en genel şekliyle iki gruba ayrılır:

1. Analitik Metotlar

2. Sayısal (Numerik) Metotlar

İteratif araştırma işlemi için yaygın olarak kullanılan nümerik metotlar:

- Eşit aralıklı araştırma
- Altın bölüm araştırma
- Polinom interpolasyonu
- En-dik iniş metodu
- Eşlenik gradyent metodu
- Değiştirilmiş Newton araştırma algoritması

İteratif Araştırma İşleminin Algoritması:

Adım 1: Bir başlangıç çözümünü ($x^{(0)}$) al. İterasyon sayacını sıfırla ($t = 0$).

Adım 2: Araştırma uzayında bu nokta için bir araştırma yönü ($d^{(t)}$) belirle. (Bu işlem, genellikle sınırlamasız optimizasyon problemlerinde maliyet fonksiyonu ile gradyentinin hesaplanmasına ihtiyaç duyarken sınırlamalı olanlarda sınırlama fonksiyonlarının da hesaplanmasını gerektirir).

Adım 3: Araştırmanın yakınsamasını kontrol et. Eğer, yakınsamış ise araştırma işlemini bitir. Yoksa, devam et.

Adım 4: Pozitif bir adım büyüklüğü ($\alpha^{(t)}$) hesapla.

Adım 5: Yeni çözümü (noktayı), aşağıdaki formülü kullanarak belirle.

$$x^{(t+1)} = x^{(t)} + \alpha^{(t)} d^{(t)}$$

Adım 6: İterasyon sayacını bir artır ($t = t+1$) ve 2. Adıma git.

Araştırma işlemi: $x^{(t-1)}$ önceki çözümü x^t mevcut çözümü, $x^{(t+1)}$ erişilecek bir sonraki çözümü ve $d^{(t)}$ de araştırma yönünü temsil etmektedir.

Mevcut çözümden bir sonraki çözüme geçiş işlemi $d^{(t)}$ araştırma yönündedir. Yukarıda verilen algoritmik yapıdan anlaşıldığı araştırma işleminin iki alt işlemden oluştuğu görülmektedir:

- 1) Araştırma yönünün ($d^{(t)}$) tayini ve
- 2) Adım büyüklüğünün ($\alpha^{(t)}$) belirlenmesi

Araştırma yönü gradyent bilgisine bağlı olarak belirlendikten sonra adım büyüklüğünün hesaplanması tek boyutlu bir minimizasyon problemi haline gelir. Adım büyüklüğünün belirlenmesi için yukarıda belirtilen yöntemler geliştirilmiştir.

B) Eniyileme algoritmaların başka bir sınıflandırma şekli de çözümleri geliştirmek için kullandıkları kuralların özelliklerine göre yapılmaktadır. Genelde geçiş kuralları, yani eski çözüm(ler)den yeni çözüm(ler) üretmek için kullandıkları kurallar:

- 1) deterministik veya
- 2) olasılık tabanlı ya da
- 3) her iki özelliği de sahip olmaktadır.

2.2 Sezgisel Algoritmalar

2.2.1 Giriş

Bu algoritmalar, çözüm uzayında optimum çözüme yakınsaması ispat edilemeyen algoritmalar. Bu tür algoritmalar yakınsama özelliğine sahiptir, ama kesin çözümü garanti edemezler ve sadece kesin çözüm yakınındaki bir çözümü garanti edebilirler.

Sezgisel algoritmaların gerek duyulmasının sebepleri:

- a) Optimizasyon problemi, kesin çözümü bulma işleminin tanımlanamadığı bir yapıya sahip olabilir.
- b) Anlaşılabilirlik açısından sezgisel algoritmalar karar verici için çok daha basit olabilir.
- c) Sezgisel algoritmalar, öğrenme amaçlı ve kesin çözümü bulma işleminin bir parçası olarak kullanılabilir.
- d) Matematik formülleriyle yapılan tanımlamalarda;
 - hangi amaçlar ve hangi sınırlamalar kullanılmalı,
 - hangi alternatifler test edilmeli,
 - problem verisi nasıl toplanmalı

gibi en zor tarafları genellikle ihmal edilir. Model parametrelerini belirleme aşamasında kullanılan verinin hatalı olması, sezgisel yaklaşımın üretebileceği alt optimal çözümden daha büyük hatalara sebep olabilir.

Sezgisel algoritmaların değerlendirilmesi için kriterler:

1. Çözüm kalitesi ve hesaplama zamanı:

İyi bir algoritma, ayarlanabilir parametreler setine sahip olmalı ve bu parametreler kullanıcıya önemlilik açısından hesaplama maliyeti ile çözüm kalitesi arasında bir vurgulamanın yapılabilmesine imkan vermelidir. Yani, çözüm kalitesi ve hesap zamanı arasındaki ilişki kontrol edilebilmelidir.

2. Kod basitliği ve gerçekleştirilebilirlik:

Algoritmanın yeni alanlara kolaylıkla uygulanabilmesi için algoritma prensipleri basit olmalıdır.

3. Esneklik:

Algoritmalar modelde, sınırlamalarda ve amaç fonksiyonlarında yapılabilecek değişiklikleri kolayca karşılayabilmelidir.

4. Dinçlik (Uygunluk):

Başlangıç çözümünün seçimine bağlı olmaksızın her zaman yüksek kaliteli, kabul edilebilir çözümleri üretebilme kabiliyetine sahip olmalıdır.

5. Basitlik ve Analiz Edilebilirlik:

Karmaşık algoritmalar, esneklik ve çözüm kalitesi açısından basit algoritmalarından daha zor analiz edilebilmektedir. Algoritma kolayca analiz edilebilir olmalıdır.

6. Etkileşimli Hesaplama ve Teknoloji Değişimleri

İyi bir kullanıcı ara-yüzü herhangi bir bilgisayar sistemini veya algoritmayı daha çekici yapmaktadır. Bunun en önemli avantajı çözümlerin grafiksel olarak sergilenebilmesidir.

2.2.2 Sezgisel Algoritmaların Sınıflandırılması

Klasik Sezgisel Algoritmalar ve Modern Sezgisel Algoritmalar olmak üzere ikiye ayrılır.

2.2.2.1 Klasik Sezgisel Algoritmalar

Bazı sezgisel-bölgesel araştırma algoritmaları başlangıç çözümüne bağlı olarak bölgesel optimum çözümler üretirler. Bu, genellikle iteratif gelişme yöntemlerinde karşılaşılan bir durumdur. Bölgesel optimal çözüm, küresel optimal çözümden çok uzak olabilir ve yine çoğu ayrık optimizasyon problemlerinde uygun bir başlangıç çözümünün seçimi için genel bilgiler mevcut olmayabilir.

Sezgisel-bölgesel araştırma metotlarının mahsurlarının bazılarını ortadan kaldırmak için basitlik ve genellik korunarak aşağıda belirtilen işlemler değerlendirilebilir:

- Çok sayıda başlangıç çözümleri ile algoritmanın tekrar tekrar koşulması. (Ancak bu durumda rastgele farklı başlangıç çözümleriyle hesaplama işlemi oldukça maliyetlidir ve hiçbir zaman optimal çözümün bulunması garanti edilemez.)
- Çok daha karmaşık bir komşuluk yapısının tanımlanması suretiyle çok daha iyi komşu çözümlerin üretilmesinin sağlanması.

- c) Algoritmaların koşılması esnasında bilgi toplanması sağlayan karmaşık öğrenme stratejilerinin kullanılması ve bu bilginin her bir koşma sonunda belirli bölgeleri veya çözümleri cezalandırmak amacıyla kullanılması.
- d) Bölgesel araştırma metotlarında amaç, sadece fonksiyon değerini azaltan çözümlerin kabul edilmesi olmasına rağmen, bölgesel optimallikten kaçmak için bazı gelişmeleri önleyen hareketlerin kabullenilmesidir.

2.2.2.1.1 Bölgesel Araştırma İniş Metotları

Bir eniyileme problemi “P”, mümkün olabilen çözümler seti ve her çözüm “S” ‘e nümerik bir değer “C(S)” atayan bir amaç fonksiyonu “C” ile tanımlanabilir. Minimizasyon probleminde optimum çözüm, mümkün olan minimum amaç değerli çözümdür.

- a) Basamak-basamak metodu:

Bölgesel araştırma iniş algoritması tanımlamak amacıyla çözümlerden yeni çözümleri elde etmek için kullanılan yöntemlerden biridir.

Verilen bir çözüm S’den bir adımla (bir hareket) elde edilebilecek çözümlerin seti S’in komşuluğu ve N(S) komşu üretme mekanizması olarak isimlendirilir. Komşu üretim mekanizmasına ilave olarak algoritma, komşular arasındaki alternatifleri araştırmalı ve seçim kriterlerini belirlemelidir. İki temel kriter vardır:

- 1) Hareket stratejisi: İlk hareket stratejisi, amaç fonksiyon değerini (F) düşüren ilk komşuyu (S) seçer.
- 2) Eniyi gelişim stratejisi: Mevcut çözümün tüm komşuları arasında amaç değerinde en büyük düşüşü sağlayan en iyi komşuyu (S’) seçer.

Bu tip sezgisel tabanlı bölgesel iniş algoritmasının temel adımları:

Adım 1: Bir başlangıç çözümü S al.

Adım 2: Test edilmemiş komşu $S' \in N(S)$ komşu üretme mekanizması var ise aşağıdaki adımları gerçekleştir. S’, S’in test edilecek bir komşusu olsun: $C(S') < C(S)$ ise o zaman S’ ‘nü S olarak kabul et. Yoksa, ikinci adımı tekrarla.

Adım 3: Dur ve bölgesel optimum S’yi bulunabilecek en iyi çözüm olarak ver.

2.2.2.2 Modern Sezgisel Algoritmalar(Yapay Zeka Yöntemleri)

Optimizasyon problemleri için geliştirilen çoğu yöntemler, çeşitli kabullerle elde edilen matematiksel bağıntılara dayalı belirgin yöntemlerdir. Klasik yöntemler olarak adlandırılan bu yöntemlerin daha ziyade problemlere özel olması, yani genel (esnek) olmaması, problemin kesinlikle matematiksel fonksiyonlarla tanımlama gerekliliği gibi bir çok mahsurları vardır.

Son zamanlarda doğada var olan sistemleri ve olayları temel alarak optimizasyon yöntemleri oluşturulmuştur. Bu optimizasyon yöntemleri Yapay Zeka Yöntemi veya Modern Sezgisel Yöntem ya da Zeki Yöntem olarak adlandırılmaktadır. Yapay Zeka Yöntemleriyle, daha esnek, daha genel ve performansı çok daha yüksek özellikleri içeren optimizasyon algoritmalarının geliştirilmesi amaçlanmıştır.

Yapay Zeka Dalları (Modern Sezgisel Yöntemler) :

- 1) Bulanık Mantık (Lütfi, 1965)
- 2) Genetik Algoritma (Holland, 1975)
- 3) Yapay Sinir Ağları (McCullogh ve Pits, 1943, Hebb, 1949, Rosenblatt, 1958)
- 4) Karınca Koloni Algoritması (Dorigo ve ark., 1991)
- 5) Tabu Araştırma Algoritması (Glover, 1989)
- 6) Yapay Bağışıklık Algoritması (Farmer ve ark., 1986)
- 7) Yapay Isıl İşlem Algoritması (Kirkpatrick, 1983, Aorts ve Korst, 1988)
- 8) Diferansiyel Gelişim Algoritması (Storn, 1997)

Yukarıda yapay zeka dallarından birincisi ve üçüncüsü dışındakiler optimizasyon amaçlı geliştirilmiştir.

Optimizasyon algoritmalarının çalıştırılmasında:

- 1) az kontrol parametresi kullanan,
- 2) problemin yapısından bağımsız,
- 3) az işlemle yeni çözüm üretebilen,
- 4) erken yakınsama problemi olmayan ve

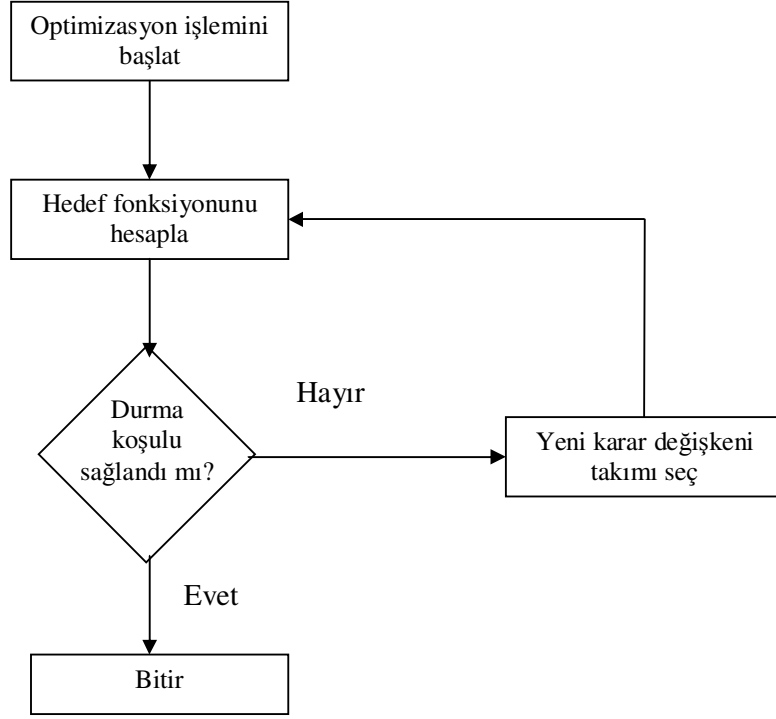
5) her başlangıç şartında en iyi çözümü veya en iyi çözüme oldukça yakın çözümler garanti eden

bir algoritma tasarlamak temel amaçtır.

Her algoritmanın doğal olarak, alınan kriterler açısından birbirlerine göre faydaları ve mahsurları mevcuttur. Bu yüzden genel eğilim, değişik algoritmaların olumlu özelliklerini bir araya getirmek suretiyle performansı daha yüksek melez algoritmaların tasarlanması şeklindedir.

Yapay Zeka Algoritmalarını değişik kriterlere göre karşılaştırmak mümkündür. Kullandıkları Araştırma Stratejilerine göre sınıflandırmak gerekirse:

- 1) Yapay ısıtma işlem ve tabu araştırma algoritmaları Komşuluk Araştırma Prensipleri'ne göre araştırma yapan algoritmalarlardır. Yani, bir başlangıç çözümü alıp onu iteratif olarak geliştirmeye çalışırlar. Bu algoritmalar, bölgesel-sezgisel komşuluk araştırmanın çok daha gelişmiş versiyonlarıdır. Yapay ısıtma işlem algoritması fizik biliminden (istatistiksel mekanik), Tabu araştırma kurallarından esinlenerek türetilmiştir.
- 2) Genetik, karınca koloni, yapay bağışıklık ve diferansiyel gelişim algoritmaları ise Paralel Araştırma Yapısına sahiptirler. Yani, tek bir çözüm alıp onu iteratif olarak geliştirmek yerine bir grup çözüm olarak bu çözümlerden daha iyi yeni bir çözüm veya çözümler üretmeye çalışırlar. Bu dört algortmada Komşuluk Prensibine dayanan bir araştırma özelliği doğrudan gözükme bile, karınca koloni, yapay bağışıklık ve diferansiyel gelişim algoritmalarında böyle bir özellik mevcuttur. Genetik algoritma ve yapay bağışıklık algoritması biyoloji biliminin prensiplerini temel alarak, karınca koloni algoritması ise gerçek karınca kolonilerinin davranışları örnek alınarak geliştirilmiştir.



Şekil 2.1: Optimizasyon Akış Şeması

2.3 Yapı Tasarımında Optimizasyon

2.3.1 Optimizasyon Probleminin Tanımı

Eniyileme problemi, eşitlik ya da eşitsizlikler şeklinde ifade edilen belirli kısıtlamalarla sınırlandırılmış x_i tasarım değişkenleri cinsinden yazılan bir $F(x_i)$ amaç fonksiyonunun maksimum veya minimum değerini arama problemidir. Bu fonksiyon, “Amaç” veya “Hedef” fonksiyonu olarak adlandırılır ve göz önünde bulundurulmuş kısıtlamalar da problemin “kısıtlamalar” ı olarak bilinir. Problemdaki bilinmeyenler ise “tasarım değişkenleri” dir.

Genel hallerde iken, optimizasyon sürecinin yer alabileceği bölgeyi sınırlayan kısıtlamalar vardır. Bu kısıtlamalar, o problemle ilgili pratik ve teorik koşulları yansıtır. Kısıtlamalar ile sınırlanan bölge ise “uygun bölge” veya çözüm kümesi (araştırma uzayı) olarak isimlendirilir. Böylece, bu bölge içinde seçilecek herhangi bir çözüm, optimum olmasa bile problemin çözümü için geçerli, yani uygun çözüm olacaktır.

Amaç fonksiyonları, genel durumlarda birden fazla değişkenlerle tanımlanır ve birden fazla tepe noktalarına sahiptir. Bir fonksiyon ile tanımlanan bir bölge içindeki

iki nokta arasında geçirecek bir doğru parçası eğer bölgenin dışına hiç geçmeyecekse bu bölge “dışbükey” olarak adlandırılır. Bu tanım optimum sonucun tek olup olmamasının belirlenmesinde yararlıdır. Kısıtlamalar yüzeyi, çok boyutlu uzayda dışbükey ise çözümden “mutlak (global) optimum” elde edilir. Bu durum sağlanamazsa “yerel (lokal=bölgesel) optimum” elde edilmiş demektir.

2.3.1.1 Amaç Fonksiyonu

Optimizasyon; problemin çözümünde, en iyiyi, en ideali bulma süreci olarak tanımlanabilir. Amaç fonksiyonu, bir optimizasyon probleminde ulaşılmak istenen hedefin matematiksel ifadesidir. Burada geçen “iyi” veya “ideal” kavramı, problemin türüne göre değişir. Bazen amaç fonksiyonunun minimum değerinde olması en iyi iken, bir başka problemde maksimum olması ideal bir çözüm olabilir. Bu tamamen optimize edilmek istenen problemde neyin amaçlandığına bağlıdır.

Optimize edilecek problem, bir maliyet problemi ise, en düşük maliyetle en ideal çözüme ulaşılmak istenir. Böyle bir durumda amaç fonksiyonunun minimize edilmesi, yani, en düşük değerinin bulunması gereklidir. İnşaat mühendisliğindeki optimum tasarım problemlerinin çoğunda minimizasyon problemi ortaya çıkmaktadır.

Amaç fonksiyonu, bir veya çok sayıda tasarım değişkenleri cinsinden ifade edilir. Bunlara değişkenler fonksiyonu denir. Yapı mühendisliğinde amaç fonksiyonu, yapının ağırlığını, yapı maliyetini veya diğer bazı şartları ifade edebilir.

F, maksimize veya minimize edilmek istenen amaç fonksiyonunu, x_i tasarım değişkenlerini göstermek üzere amaç fonksiyonunun genel formu

$$F = f(x_1, x_2, x_3, x_4, \dots, x_n) = f(x) \quad (2.6)$$

şeklinde dir. Maksimize edilecek amaç fonksiyonunun (-) değer ile çarpılması durumunda problem minimizasyon problemine dönüşür.

$$\max f(x_1, x_2, x_3, x_4, \dots, x_n) = -\min(-f(x_1, x_2, x_3, x_4, \dots, x_n)) \quad (2.7)$$

Bütün optimal tasarım yöntemlerinde, amaç fonksiyonunun seçimi en önemli kararlardan biridir. Bazı durumlarda amaç fonksiyonunu formüle etmek kolaydır. Örneğin, eğer yapı maliyetinin ağırlığıyla orantılı olduğu varsayılmışsa, amaç

fonksiyonu, yapının ağırlığı alınabilir. Bazı durumlarda ise, amaç fonksiyonunun matematiksel formülasyonunu oluşturmak çok zor olabilir.

Ağırlık minimizasyonu, kolayca formüle edilebilir olması nedeniyle literatürde en yaygın olarak kullanılan amaç fonksiyonudur. Yapının ağırlığı, önemli kriterlerden biridir, ancak ağırlığın minimum olması her zaman en ekonomik çözümü sağlamaz. Maliyet, çoğu zaman ağırlıktan daha fazla önem kazanır. Gerçek maliyet fonksiyonunu oluşturmak için yeterli veri elde etmek çoğu zaman oldukça zordur. Genel bir maliyet fonksiyonunda, malzeme, işçilik ve nakliye gibi birçok değişken bulunur. Buna ek olarak, maliyet fonksiyonu, tasarım ve yapımda gerekli olan işletme, bakım, onarım ve sigorta gibi diğer maliyet faktörlerini de içerebilir. Bununla beraber, yapı maliyetinin önemli bir kısmı malzeme ağırlığı ile orantılı olduğu için çoğu zaman yapı ağırlığının en küçüklenmesi yapı optimizasyonu için oldukça uygun bir yaklaşım olarak kabul edilmektedir.

2.3.1.2. Tasarım Değişkenleri

Optimizasyon işlemi yapılırken, yapı sistemini meydana getiren bazı parametrelerin başlangıçta belirlenmesi gerekir. Bu parametreler, sistem düzenlenirken, “önceden belirlenen parametreler” olarak adlandırılırlar ve optimizasyon algoritmasıyla değişmezler. Önceden belirlenemeyen parametrelere ise, “tasarım değişkenleri” denir. Tasarım değişkenleri ile önceden belirlenen parametreler, birlikte bir tasarımı tanımlayacaklardır. Tasarımcı, belirli parametrelerin seçiminde, özgür olmayabilir veya iyi bir sonuç meydana getirecek olan belirli parametreler ancak deneyim ile bulunabilir. Bazı etkenlerin sabit kabul edilmesiyle, yani optimizasyon probleminin çözümü sırasında bunların sabit alınmasıyla, problem, büyük bir oranda basitleştirilebilir.

Matematiksel bakış açısıyla, tasarım değişkenleri iki şekilde olabilir:

- a) Sürekli tasarım değişkenleri,
- b) Ayrık tasarım değişkenleri,

Bir optimizasyon probleminin çözümü sırasında, problemi ifade eden değişkenlerin, belli sınırlar (alt ve üst) içindeki her bir değeri alması durumu, o değişkenlerin “sürekli tasarım değişkenleri” olduğunu gösterir.

Ancak, yapısal optimizasyon problemlerinde çözümlü ifade eden tasarım değişkenleri her zaman sürekli değildir. Donatı alanı, çapı, ağırlığı, yapıyı oluşturan elemanların enkesit boyutları ve alanları gibi değişkenler sürekli değil, ayrık değerler alabilirler. Bu tür değişkenler “ayrık tasarım değişkenleri” denir.

Optimizasyon probleminin çözümü sırasında, sürekli veya ayrık tasarım değişkenleri arasında seçim yapmak önemlidir. Verilen bir aralıkta, büyük bir sayıda ayrık tasarım değerlerinin olması ve bu ayrık değerlerin birbirine çok yakın alınması durumunda bir sürekli değişken ifadesinin kullanılması daha uygun olur.

Fiziksel bakış açısıyla ise, optimizasyon yöntemiyle değişen tasarım değişkenleri, yapının aşağıda sıralanan bazı özellikleri olabilir:

- a) Malzemenin mekanik veya fiziksel özellikleri,
- b) Yapının topolojisi yani, bir yapıdaki eleman sayılarının veya elemanların birleşim noktalarının düzeni,
- c) Yapının konfigürasyonu veya geometrik planı,
- d) Eleman boyutları veya enkesit boyutları.

Bu özellikleri kısaca açıklar olursak:

- a) Malzeme Tasarım Değişkenleri:

Malzeme seçimi, ayrık özelliklere sahip, yani seçimin, değişkenleri ayrık bir yerleşiminden meydana getirilmiş olması gibi özel bir problemi ortaya koyar. Bu tür ayrık tasarımlar, optimizasyon yönteminde, hesaplama zamanında ve karmaşıklıkta önemli artışı da gerektirebilir. Uygun malzemenin az sayıda olması durumunda, her malzeme için ayrı ayrı optimizasyon işlemi yapmak ve sonuçları karşılaştırmak büyük olasılıkla oldukça hızlı ve verimli olacaktır. Yapıyı oluşturan elemanlarda yüksek performanslı kompozit malzemelerin kullanılması, tasarım değişkenleri arasında malzeme özelliklerinin alınmasının daha uygun olmasına neden olacaktır.

- b) Topolojik Tasarım Değişkenleri

Yapının topolojisi, bir yapıdaki eleman birleşim noktalarının düzeni ve/veya eleman sayısını ifade etmektedir. Tamsayı topolojik değişkene, bir köprüdeki açıklıkların sayısı, bir çatı sistemine mesnetlenmiş kolon sayısı veya ızgara sistemdeki elemanların sayısı da örnek olarak verilebilir.

c) Geometrik Tasarım Değişkenleri

Geometrik planlı değişkenlere örnek olarak bir çevredeki veya bir kafes sistemdeki birleşim noktalarının koordinatları verilebilir. Bu tip değişkenlere, bir köprüdeki mesnetlerin yeri, sürekli kirişdeki açıklıkların uzunlukları veya kabuk yapıların yüksekliği gibi örnekler de verilebilir. Genellikle, yapının geometrisi, sürekli değişkenlerle belirlenmektedir.

d) Enkesit Tasarım Değişkenleri

Kesit boyutları, en basit tasarım değişkenleridir. Bir kafes sistem elemanının enkesit alanı veya levhanın kalınlığı bu tip tasarım değişkenlerine verilebilecek bazı örneklerdir. Bazı durumlarda, tek bir tasarım değişkeni, kesiti anlatmak için yeterlidir, ancak çok detaylı tasarımlarda, her kesit için farklı tasarım değişkenlerinin alınması gerekebilir. Bu değişkenler, elemanın enkesit boyutları, alanı, atalet momenti ve mukavemet momenti olabilir.

Enkesit tasarım değişkenli bir problemde, tasarım değişkenlerinin alabileceği değerler sınırlı sayıda ve ayrık olabilir. Bu durumlarda, tasarım değişkenleri olarak, sadece uygun ayrık değerlerin bulunduğu bir listenin hazırlanması gerekir.

2.3.1.3 Kısıtlamalar

Bir optimizasyon probleminde amaç fonksiyonunun içerdiği tasarım değişkenlerinin veya tasarım değişkenlerini dolaylı olarak etkileyen parametrelerin sağlaması gereken koşullar olabilir. Bu koşullar “kısıtlama” olarak adlandırılır ve kısıtlamaların yer aldığı problemlere de “kısıtlamalı optimizasyon problemleri” denir. Optimum tasarım, problemdeki bütün kısıtlamaları sağlamalıdır. Eğer bir tasarım, bütün gereklilikleri bir arada toplamış ve bunları sağlıyorsa uygun tasarım olarak adlandırılır. Uygun bir tasarım elde etmek için sağlanması gereken sınırlayıcılar “kısıtlama”lardır.

Bu fonksiyonun sağlaması gereken şartlar şu şekilde yazılabilir:

$$F_i(x_1, x_2, x_3, \dots, x_n) \leq b_i \quad (3.8a)$$

ve/veya

$$F_k(x_1, x_2, x_3, \dots, x_n) = d_k \quad (3.8b)$$

ve/veya

$$\prod_j(x_1, x_2, x_3, \dots, x_n) \geq h_j \quad (3.8c)$$

Buradaki F , f , ve \prod , x_i tasarım değişkenleri cinsinden yazılmış kısıtlama fonksiyonlarıdır. Tipik bir optimizasyon probleminde n tane değişken vardır. b_i , d_k ve h_j ise bu değişken fonksiyonlarının sınır değerleridir.

“=” durumunun yer aldığı kısıtlamalara eşitlik kısıtlamaları (aktif kısıtlamalar),

“ \leq veya \geq ” durumlarının bulunduğu kısıtlamalara ise eşitsizlik kısıtlamaları (pasif kısıtlamalar) denir. Eşitlik kısıtlamaları optimumda mutlaka aktiftirler. Pasif kısıtlamalar ancak, eşitsizliğin eşitlik durumuna dönüşmesi durumunda aktif olabilirler. Özet olarak, optimum tasarım, çok boyutlu uzayda, n değişken varsa n -boyutlu uzayda, eşitlikleri gösteren çok boyutlu yüzeylerin kesişme noktalarından birinde bulunur.

Fiziksel bakış açısıyla kısıtlamaların iki farklı tanımı verilebilir:

- a) Malzemenin enkesit büyüklükleri veya boyutları ile ilgili sınırlamalarına “alan kısıtlamaları” veya “tasarım kısıtlamaları” denir. Önceden belirli olan bu kısıtlamalar yönetmelik sınırlamaları, işlevsellik, işçilik veya mimari gerekler gibi çeşitli durumlardan kaynaklanmış olabilir. Böylece söz konusu kısıtlamalar, bir tasarım değişkeninde veya bir tasarım değişkeni grubunun bağıl değerini saptayan bir bağıntıda sınırlamalarla (alt ve üst sınır) belirlenmiştir. Bu tür kısıtlamalar o problemin çözümünde kullanılan yönetmelik şartları olarak da düşünülebilir. Bir çatının minimum eğimi, bir kolonun minimum boyutları veya bir levhanın minimum kalınlığı tasarım kısıtlamalarının başlıca örnekleridir.
- b) Davranış gerekliliklerinden ortaya çıkan kısıtlamalar ise “davranış kısıtlamaları” olarak isimlendirilir. Maksimum gerilme, deplasman veya burkulma mukavemeti sınırlamaları davranış kısıtlamalarının tipik örnekleridir.

1) Gerilme Kısıtlamaları

Yapı elemanlarında meydana gelen gerilmelerin belirli sınırları aşmama koşulu olarak tanımlanmıştır.

$$\sigma_i \leq \sigma_{\text{mak}} \text{ (Normal)} \quad (3.9)$$

$$\tau_i \leq \tau_{\max} \text{ (Kayma)} \quad (3.10)$$

2) Yerdeğiştirme Kısıtlamaları

Yapı tasarımı optimizasyonunda yaygın olarak kullanılan üç tip yerdeğiştirme kısıtlaması vardır. Bunlar:

- j düğüm noktasında u_j üst sınır değeri ile sınırlanmış olan x_j yerdeğiştirme değeri. Bu yerdeğiştirme kısıtlaması şöyle yazılabilir:

$$x_j \leq u_j \quad (3.11)$$

- Birçok tasarım şartnamesinde, verilen bir yerdeğiştirme değeri için üst sınır belirlenemez. Bunun yerine, üst sınır değerleri olarak, ötelenmeler, yani bir elemandaki rölatif yerdeğiştirmeler alınır.
- Bir çok yapısal tasarım probleminde, yapının düğüm noktalarında dönmeler meydana gelmektedir. Bunların tamamının bazı sınırlarla kısıtlanmış olması gerekir. $\theta_{j\max}$ ise, dönmenin maksimum değeri olmak üzere, yerdeğiştirme kısıtlamasının şu şekilde olduğu kabul edilir.

$$\theta_j \leq \theta_{j\max} \quad (3.12)$$

2.3.2 Yapı Optimizasyonunda Kullanılan Yöntemler

2.3.2.1 Genel Bilgi

Genel olarak, yapı mühendisliğinde, mühendisin tecrübe ve sezgisine dayanan ve ülkemizde uygulamada yaygın olarak kullanılmakta olan deneme-yanılma yöntemi, mühendisin benzer tasarımlardan elde ettiği bilgilere dayanır ve basit optimizasyon problemlerinde iyi sonuçlar verebilir. Ancak bu yöntem, karmaşık problemlerin çözümünün üstesinden gelemez. Yapısal tasarımda, büyük değişim ve gelişimin yaşandığı günümüzde çok daha yeni yöntemlerin kullanımına başlanmıştır.

Yapı optimizasyonunda kullanılan yöntemlerin başlıcaları:

1) Matematik Programlama Yöntemleri

Bu yöntemler, bir optimizasyon probleminde, amaç fonksiyonunu, verilen kısıtlamalar çerçevesinde, matematiksel çözüm algoritmaları ile minimize veya maksimize etmeyi öngörürler. Tasarım ile ilgili değişkenler çok boyutlu tasarım

uzayının koordinatları olarak ele alınırlar. Kısıtlamalar ise bu uzayı uygun ve uygun olmayan bölgelere ayırırlar. Bu yöntemlerin bazıları:

1.1) Doğrusal Programlama Yöntemleri

a) Simplex Yöntemi

b) Tam Sayılı Programlama Yöntemi

1.2) Doğrusal Olmayan Programlama Yöntemleri

a) Kesen Düzlem Yöntemi

b) Geometrik Programlama Yöntemi

b.1. Prototip Geometrik Programlama Yöntemi

b.2. Signomlar İle Geometrik Programlama Yöntemi

c) İkinci Derece Programlama Yöntemi

d) Ceza-Fonksiyon Yöntemleri

d.1. Dış Ceza Yöntemi

d.2. İç Ceza Yöntemi

e) Ardı sıra Doğrusal Programlama Yöntemleri

1.3) Ayrık Programlama Yöntemi

2) Optimumluk Kriteri Yöntemleri

Matematiksel programlama yöntemlerinin optimizasyon probleminin çözümü için kısıtlamalar çerçevesinde doğrudan doğruya amaç fonksiyonunu ele almasına karşın, optimumluk kriterleri yöntemleri, yapı teorisi ve ilgili enerji bağıntılarını kullanarak, berilmiş kısıtlamalar bağlı kalmak koşuluyla ardışık yaklaşımla çözüme götürecek bağıntılar üretmeyi amaçlar.

Optimumluk kriteri yöntemi, hesaplama açısından hızlı olmasına rağmen bunu sağlamak için yapının davranışı konusunda uzmanlaşmış olmak gerekir ve optimuma yakınsamak her zaman garanti değildir. Optimumluk kriterleri yöntemlerinin hızlı yakınsaklığı, sağlanması istenen kriterlerin problemin başında belirtilmesinden dolayıdır. Matematiksel programlamada ise optimum, amaç fonksiyonunun değerinde yeni bir değişikliğe olanak kalmadığında tanımlana bilmektedir.

Bu yöntem, matematiksel programlama yöntemleri gibi genel bir yöntem değildir. Her problem için ayrı bir kriter türetme güçlüğüne rağmen, daha hızlı sonuç verdiği için, bazı arařtırmacılar, optimumluk kriteri yöntemini kullanmayı tercih etmişlerdir. Bunun nedeni ise, matematik programlama yöntemlerinde deęişken sayısı arttıkça yakınsama hızının bazen çok yavaş olmasıdır.

3) Yapay Zeka Yöntemleri

Son zamanlarda geliştirilmiş olan optimizasyon yöntemlerinden yapay zekanın yapısal optimizasyonda kullanılan alt dallarından bazıları şunlardır:

- a) Genetik Algoritma
- b) Yapay Sinir Ağları
- c) Tabu Arařtırma Algoritması
- d) Yapay Isıl İşlem Algoritması
- e) Karınca Koloni Algoritması
- f) Uzman Sistemler
- g) Evrimsel Gelişime Dayalı Yapısal Optimizasyon

Bu yöntemler hakkında literatürde yeterli düzeyde bilgi olduğundan dolayı burada Genetik Algoritma hakkında bilgi verilecek ve bir sonraki bölümde ise ayrıntılarıyla anlatılacaktır.

2.3.2.2. Genetik Algoritma

GA'lar rastgele arama yöntemlerinden farklı olarak, ihtimal ilkelerini karar deęişkeni uzayında genetik işlemler yapmada araç olarak kullanır. GA'lardaki rastgele yaklaşımlar sonuç çözümün yerel en iyi çözümlere takılıp kalmamasını sağlar. **Adeli ve Hung (1995)** çalışmasına göre GA'da kodlama, başlangıç şartları, dinçlik (uygunluk) ölçüsü, evrim performansı ölçümü ve çalışma parametreleri olmak üzere beş tane adım vardır. GA'lar doğal olayların gelişmesindeki genetik mekanik ilkelere göre çalışırlar. Başlangıçta çözüm atamalarının belirli bir kurallar dizisi altında rastgele genetik olarak gelişmesi ile en iyi çözüme doğru ilerleyerek sonuca varılabilir (**Goldberg, 1989**). GA'lar klasik yöntemlerden aşağıdaki noktalarda farklıdır (**Buckles ve Petry, 1994**):

- a) GA' lar, gelişmesi sırasında karar uzayında o zamana kadar mevcut olan bilgilerden yararlanarak en iyi çözüme ulaşmaya çalışır.
- b) GA' lar, yapısında saklı paralel hesaplama yöntemine sahiptir. Bu sayede aynı zamanda birçok noktadan hareketle çözüm uzayındaki en iyi noktaya doğru adım adım gelişerek yaklaşırlar.
- c) GA' lar rastgele algoritmalarıdır. Sonuçlarına ulaşmak için ihtimal ilkelerinden yararlanır. Buna göre çalışmalar sırasında bir rastgele sayı üreticisine gerek vardır.

GA' lar, birçok çözüm adayı üzerinde aynı aynı zamanlı olarak işleyişlerini sürdürürler. Bu süreç esnasında komşu ve önceki noktalardan topladığı bilgiler ve rastgele işlemler sayesinde daha da iyi çözümlere doğru ilerleyen bir mekanizmaya sahiptir.

3. GENETİK ALGORİTMALAR

3.1. Evrimsel Hesaplama

Evrimsel Hesaplama, evrim prensibine dayalı algoritmaların hepsini temsilen kullanılan bir terimdir. Bu tekniğe giren algoritmalarından bazıları genetik algoritmalar, evrimsel programlama, evrimsel stratejiler, genetik programlama ve diferansiyel gelişim algoritmalarıdır. Ayrıca bu yöntemlerin önemli özelliklerinin birlikte kullanılmasıyla oluşan kuvvetli yapılar içeren melez sistemler de vardır.

Bir problemi çözmeye kullanılacak herhangi bir evrimsel algoritmanın temel ifadeleri:

- a) Problem için çözümlerin genetik temsili
- b) Çözümlerin başlangıç popülasyonunu oluşturacak bir yöntem,
- c) Çözümleri uygunluk açısından değerlendirecek değerlendirme fonksiyonu (çevre),
- d) Genetik yapıyı değiştirecek operatörler,
- e) Kontrol parametrelerinin değerleri

Bir evrimsel algoritmanın temel adımları:

1. Popülasyon büyüklüğü $P(t)$ ' yi düzenle.
2. $P(t)$ ' yi değerlendir.
3. Durdurma koşulu sağlanıncaya kadar aşağıdaki işlemleri tekrarla.
 - Önceki popülasyonda bulunan bireylere seçme işlemi uygulayarak yeni popülasyon oluştur.
 - Yeni popülasyonu değiştir.
 - Değiştirilmiş yeni popülasyonu değerlendir.

Evrimsel hesaplama teknikleri arasında çeşitli benzerliklerin olmasına rağmen bir çok da farklılıklar vardır. Bu algoritmaların genel yapıları:

1. Genetik algoritma, adaptasyon işlemini modellemek için düzenlenmiştir ve temelinde ikili dizi üzerinde çalışmaktadır. Temel operatörleri mutasyonla birlikte yeniden oluşumdur (çaprazlama).
2. Evrimsel stratejiler, yalnız Monte Carlo yöntemine benzer olan sayısal optimizasyon problemlerinde kullanılır. Özellikle parametre optimizasyon problemlerini çözmek için geliştirilmiştir. Bu yaklaşımda popülasyondaki her birey iki tane vektörden oluşmaktadır ($z = x + y$). İlk vektör x , araştırma uzayındaki bir noktayı, ikinci vektör y ise standart sapmaları ifade eder. Burada yalnız mutasyon işlemi uygulanır. Problem verilerinin kodlanmasına gerek duyulmamaktadır.
3. Temel Evrimsel Programlama teknikleri bireylerin kromozom yapısı ile temsil edilme prensibini kullanmış ve yeni bireyleri önceki bireylerin rastgele mutasyona uğratılması yoluyla oluşturulmuştur. 1960'lı yıllarda I. Rechenberg'in "Evrimsel stratejileri" adlı çalışmasıyla gündeme gelmiştir.
4. Genetik Programlama teknikleri, bir problemi çözmek için kullanılacak muhtemel bilgisayar yazılımlarından oluşan çözüm uzayındaki en uygun programı bulabilecek bir yöntem sağlamaktadır. Bu yöntemde her çözüm ağaç formunda temsil edilir. Genetik algoritmaların programlara uygulanmasıdır.
5. Diferansiyel gelişim algoritması, özellikle sayısal optimizasyon problemlerinin çözümü için geliştirilmiş ve yeni çözümlerin üretilmesinde çözümler arasındaki farklardan faydalanan oldukça yeni bir algoritmadır.

3.2 Genetik Bilgi

Genetik algoritmanın nasıl çalıştığını anlamak için ön genetik bilginin bilinmesi önemlidir.

1865'te G. Mendel 'karışım yoluyla kalıtım' kuralını reddederek kendi adıyla anılan yasaları belirlemiştir. Kalıtsal nitelikler ve nitel çeşitlilik gösteren karakterlerin kalıtımında Mendel yasalarının geçerli olduğu görülmüştür. Bu yasalar:

- a) Dölün tek biçimliliği,
- b) Genetik ayrılma yasası ve
- c) Karakterlerin bağımsız ayrılma yasası.

XX. yüzyılın başlarında; Amerikalı biyolog W.S. Sutton ve Alman R.Boveri, kromozomların bir kuşaktan diğerine, tıpkı Mendel'in sözünü ettiği kalıtsal etmenler (veya genler) gibi dağıldığını fark ettiler. Bu gözleme dayanarak genlerin kromozomlar üzerinde yer aldığı sonucuna vardılar. 1910'da ise T.Morgan ve grubu çalışmalar/deneyler ile birçok genin birbirine bağlı olarak soydan geçtiğini gözlemlediler.

Her türe özgü kromozom sayısı kesinlikle bellidir. Doğru sayıda kromozom içermeyen (anöploitlik) canlılarda genellikle doğuştan bozukluklar söz konusudur. Karakterleri belirleyen etmenler ALEL denilen farklı biçimleri olan genlerdir; çeşitli biçimlerde ortaya çıkan karakterler (fenotipler) aleller yoluyla açıklanmaktadır. Genler, bir bireyin tüm kalıtsal bilgilerini temsil eder ve her gen kromozomlar üzerinde özgün ve belirli bir yerde (lokus) bulunur. İki farklı karakteri belirleyen genin kuşaktan kuşağa birlikte aktarılması durumunda, genler arası bağlantıdan (linkaj) söz edilebilir. Fakat bazı durumlarda iki karakteri belirleyen aleller farklı biçimde bir araya gelebilir. Genler arasındaki bağlantı kuralına uymayan bu durumu Morgan, her iki kromozomdan olan genler arasında yer alan bir noktada kırık meydana gelmesiyle açıklanmıştır. Kırılan parçaların yer değiştirip diğer kromozomla birleşmiş olması, genlerin yer değiştirmesine yol açar. Böylece aleller arasında yeni bir düzenleme meydana gelir. Bir çift kromozomu oluşturan her kromozomun kendini eşlemesiyle iki kromatik ortaya çıkar. Bir kromozomun kromatiklerinden biri ile diğer kromozomun kromatiği arasında çaprazlaşma (crossing-over) olur. Bu durumda iki gen birbirinden ne kadar uzaksa, aralarında çaprazlama olasılığı o kadar fazladır. Kromozomlar arasında bu değişim sonucunda, normalden farklı bir genetik yapıya sahip kromozom meydana gelir buna bağlı olarak da gametler (eşey hücreleri) farklı olur.

Kalıtımın canlı hücrenin içindeki kromozomlarda yer alan ve başlı başına birer biyolojik varlık kabul edilen genlere bağlı olduğu, XX. yüzyılın ortalarına doğru anlaşılmıştır. Bu durumda genler kendi kopyalarını nasıl bir mekanizma sonucu çoğaltıyor? Genler hiçbir değişikliğe uğramadan kuşaktan kuşağa nasıl aktarılır?

Genlere bu özelliği sağlayan nasıl bir kimyasal yapıydı? 1944 senesinde, Amerikalı biyologlar O.T.Avery, C.M.Macleod ve M.McCarthy, kalıtsal maddenin kimyasal yapısını belirlediler (Kalıtsal madde bir protein molekülü değil, DNA molekülüydü.). DNA'nın işlevi nasıl yerine getirdiği ancak 1953 senesinde J.Watson ve F.Crick adlı iki bilim adamının, DNA'nın ikili sarmal modelini tanımlamasından sonra mümkün olmuştur. G. Gramov ise, DNA'nın doğrusal yapısını oluşturan baz dizileriyle, proteinleri oluşturan aminoasit zincirleri arasındaki ilişkiyi kurmuştur; Gramov'un varsayımına göre bir geni meydana getiren nükleotit dizisi, proteinlerin yapılması için gerekli bilgileri içeren şifreyi belirliyordu. Genetik şifre ise yalnızca 1966'da bir ekip çalışması sonucu çözülmüştür. DNA'nın kendini eksiksiz olarak eşlemesini engelleyen bazı etmenlerin olabilir. Bu koşullar altında bazen genetik bir değişim meydana gelmektedir. Bu değişimler 1) Kromozomlar ve 2) Genler üzerinde olarak ikiye ayrılır. Hollandalı H. De Vries tarafından önerilen ve mutasyon adı verilen bu değişimlerden sorumlu olan fiziksel ve kimyasal etmenler sonucunda kalıtsal maddede depolanmış olan genetik bilgide bazı değişimler görülür. Bu durumda ya hücre ölür ya da işlevsel açıdan değişikliğe uğrar ama çoğu zaman hücre mekanizmaları genetik şifrenin yeniden düzelmesini sağlayacak güçtedir. Hücre DNA'da meydana gelen yanlışlıkları düzeltmek için birtakım araçlara sahiptir. Onarım mekanizmalarının hareketi sonucunda hasarlı bölgeler yeniden oluşturulabilmektedir. Mutasyonlar kaotik olarak oluşmaktadır. Onların rastlanma sıklığı çok azdır ve tahmin edilmesi çok güçtür. Mutasyonlar doğal olarak meydana geldiğinde yeni genotipler oluşturabileceğinden bilimsel olarak büyük önem taşır. Dolayısıyla mutasyonlar topluluk genetiğinin evrimi ve yeni türlerin doğuşu bakımından temel bir etmendir.

Genetik algoritmalarda kullanılan önemli kavramların biyolojik yönden özeti:

- 1) Kromozomlar: Bütün canlı organizmalar hücrelerden oluşur. Her hücrede, aynı yapıda kromozomlar bulunur. Kromozomlar DNA dizileridir ve bütün organizmalar için model oluştururlar. Her kromozom DNA bloklarından, yani genlerden oluşur. Her gen, belli bir proteini çözer. Her gen kromozomda kendi pozisyonuna sahiptir. Kromozomların tümüne genom adı verilir. Genomdaki gen setlerine de genotip adı verilir.
- 2) Üreme: Üreme sırasında, ilk çaprazlama (crossover) meydana gelir. Ebeveynden gelen genler yeni kromozomu oluşturur. Yeni üretilen döl daha sonra

dönüştürülebilir. Mutasyon, DNA'nın elemanlarının biraz değiştirilmesi demektir. Bu değişiklikler, ebeveynden gelen genlerin kopyalanmasındaki hatalar yüzünden olabilmektedir. Bir organizmanın uygunluğu (fitness) organizmanın yaşamdaki başarısıyla ölçülmektedir.

3.3. Genetik Algoritmalar

3.3.1 Genetik Algoritmanın Tarihçesi

Makine öğrenmesi konusunda çalışan Michigan Üniversitesinde psikoloji ve bilgisayar uzmanı olan John Holland bu konuda ilk çalışmaları yapan kişidir (1975). Başlangıçta pratik bir faydası olmadığı düşünülen GA'lara, Holland'ın öğrencisi David E. Goldberg adlı İnşaat Mühendisi 'in gaz boru hatlarının denetimi üzerine yaptığı doktora teziyle National Science Foundation tarafından verilen Genç Araştırmacı ödülünü kazandı (1985). 1989 da konusunda bir klasik sayılan yayınlanan kitabında genetik algoritmalara dayalı tam 83 uygulamaya yer vererek GA'nın dünyanın her yerinde çeşitli konularda kullanılmakta olduğunu gösterdi

3.3.2. Genetik Algoritmanın Tanımı

GA, doğadaki evrim mekanizmasını örnek alan bir arama yöntemidir ve bir veri grubundan özel bir veriyi bulmak için kullanılır. Doğada geçerli olan en iyinin yaşaması kuralına dayanarak sürekli iyileşen çözümler üretir. Bunun için:

1. İyi'nin ne olduğunu belirleyen bir uygunluk (fitness) fonksiyonu
2. Yeni çözümler üretmek için yeniden kopyalama (recombanition)
3. Değiştirme (mutation)

gibi operatörleri kullanır.

GA'yı diğer algoritmalarından ayıran en önemli özelliklerden biri de seçmedir. Çözümün uygunluğu onun seçilme şansını artırır ancak bunu garanti etmez. Seçimde ilk grubun oluşturulması için rastgeledir ancak bu rastgele seçimde seçilme olasılıklarını çözümlerin uygunluğu belirler. En iyi çözüm veya çözümleri arayarak bulmak için karar değişkeni uzayındaki birçok başlangıç noktasından başlayarak, paralel işlemler dizisi ile en iyi yöne doğru topluca gelişerek yapar. Rastgele arama yöntemlerinden farklı olarak, ihtimal ilkelerini karar değişkeni uzayında genetik işlemler yapmada araç olarak kullanır.

GA'lar geleneksel yöntemlerden farkları (Buckles ve Petry, 1994):

- a) GA'lar, gelişmesi sırasında karar uzayında o zamana kadar mevcut olan bilgilerden yararlanarak en iyi çözüme ulaşmaya çalışır.
- b) GA'lar, yapısında saklı paralel hesaplama yöntemine sahiptir. Bu sayede aynı zamanda birçok noktadan hareketle çözüm uzayındaki en iyi noktaya doğru adım adım gelişerek yaklaşır.
- c) GA'lar rastgele algoritmalarıdır. Sonuçlarına ulaşmak için ihtimal ilkelerinden yararlanır. Buna göre çalışmalar sırasında bir rastgele sayı üreticisine gerek vardır.
- d) GA'lar, birçok çözüm adayı üzerinde aynı zamanlı olarak işleyişlerini sürdürürler. Bu süreç esnasında komşu ve önceki noktalardan topladığı bilgiler ve rastgele işlemler sayesinde daha da iyi çözümlere doğru ilerleyen bir mekanizmaya sahiptir.

Basit bir genetik algoritmanın temel adımları:

Adım 1: Olası çözümlerin kodlandığı (kromozom) bir çözüm grubu (popülasyon) oluştur.

Adım 2: Popülasyondaki her çözümün uygunluk değerini hesapla.

Adım 3: Durdurma koşulu sağlanıyorsa araştırmayı durdur.

Yoksa, aşağıdaki adımları gerçekleştir.

Tabii seleksiyon işlemini uygula (uygunluk değerleri daha yüksek olan çözümler yeni popülasyonda daha fazla temsil edilirler).

3.2. Çaprazlama işlemini uygula (Mevcut iki çözümden yeni iki yapı üretilir).

3.3. Mutasyon işlemini uygula (Çözümlerde rastgele değişim meydana getirilir).

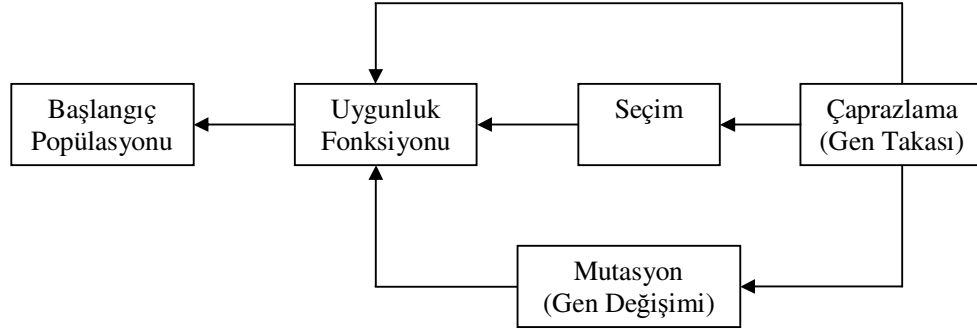
Adım 4: Adım 2 ye git.

Basit Genetik Algoritma Yordamı:

1. Olası çözümlerin kodlandığı bir çözüm grubu oluşturulur. (Burada geçen çözüm grubu, biyolojideki benzerliği nedeniyle, toplum (= population), çözümlerin kodları da (= kromozom) olarak isimlendirilir.)
 - Toplumda bulunacak birey sayısı için bir standart yoktur. Genel olarak önerilen 100-300 aralığında bir büyüklüktür. Büyüklük seçiminde yapılan işlemlerin karmaşıklığı ve aramanın derinliği önemlidir.
 - Toplum, birey sayısını belirleme işleminden sonra rastgele oluşturulur.
 - Birey sayısı belirlendikten sonra probleme bağlı olarak kromozomların kodlanması gerekmektedir. Değişik kodlama biçimleri vardır.
2. Toplumdaki her kromozomun ne kadar iyi olduğu bulunur.
 - Bu amaçla kullanılan fonksiyona “ uygunluk fonksiyonu” denir.
 - Bu fonksiyon yardımıyla kromozomların uygunluklarının bulunması ise “evrimleşme” adı verilir.
 - Uygunluk fonksiyonu genetik algoritmanın beynini oluşturmaktadır. Genetik algortmada probleme özel çalışan tek kısım bu fonksiyondur.
 - Uygunluk fonksiyonu, kromozomları problemin parametreleri haline getirerek onların bir bakıma şifresini çözmekte (decoding) ve sonra bu parametrelere göre hesaplamayı yaparak kromozomların uygunluğunu bulmaktır.
 - Genellikle genetik algoritmanın başarısı bu fonksiyonun verimli ve hassas olmasına bağlıdır.
3. Seçilen kromozomları eşleyerek yeniden kopyalama ve değiştirme operatörleri uygulanır.
 - Sonuçta yeni bir toplum oluşturulur.
 - Bu eşleme kromozomların uygunluk değerlerine göre yapılır.
 - Bu seçimi yapmak için rulet tekerliği seçimi, Boltzman seçimi, turnuva seçimi gibi seçme yöntemleri vardır.

- Yeniden kopyalama (recombination) genlerdeki genetik bilginin birinden diğerine geçmesi işlemine benzediği için “çaprazlama (=crossover)” olarak isimlendirilir. Bu işlem, toplumda çeşitliliği sağlar.
 - Değişirme'nin (mutation) etkisi, yalnızca bir çözüm üzerinde olmaktadır.
4. Yeni kromozomlara yer açmak için eski kromozomlar çıkartılarak sabit büyüklükte bir toplum sağlanır.
 5. Tüm kromozomların uygunlukları tekrar hesaplanır ve yeni toplumun başarısı bulunur.
 6. İşlemler tekrarlanarak verilmiş zaman içerisinde daha iyi olan yeni nesillerin oluşturulması gerçekleştirilir.
 7. Sonuçta toplumların hesaplanması sırasında en iyi bireyler bulunduğundan çözüm elde edilmiş olur.

Genetik Algoritmanın aşamalarından yöntemin tekrarlı karakter taşıdığı görülmektedir. Her tekrarlama (iterasyon) üretme olarak isimlendirilir. Standart Genetik Algoritmada üretme sayısı 50-500 aralığında değişmektedir (Mitchell, 1996).



Şekil 3.1: Genetik Algoritmalarda Evrimleşme Döngüleri

3.3.3. Genetik Algoritmanın Faydaları ve Mahsurları

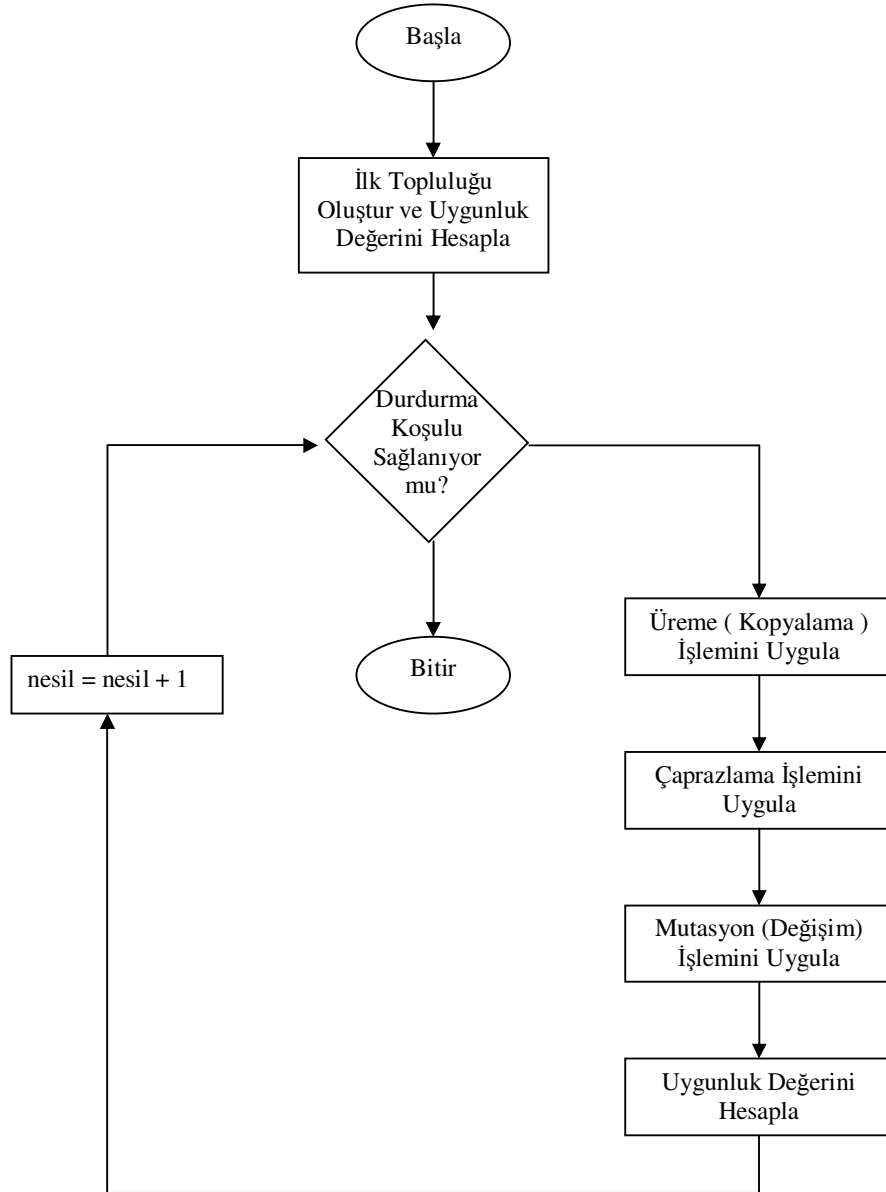
Genetik algoritma ile en iyileme (Eİ) sorunlarının çözümlesindeki bazı faydaları şunlardır:

1. Kesikli veya sürekli değişkenlerle en iyileme yapılabilir,
2. Türev alma işlemine gerek yoktur,
3. Çözüm uzayında aynı anda geniş bir alanda çok sayıda noktadan araştırmaya başlanır,
4. Çok fazla sayıda değişkenle eniyileme işlemleri yapılabilir,
5. Paralel hesaplamalara çok uygundur,
6. Çok fazla uç (enbüyük ve enküçük, ekstrem) değeri olan hedef fonksiyonları durumunda bile eniyileme yapabilir,
7. Yerel (lokal) en küçüklemeleri sıçrayarak aşabilir,
8. Sadece mutlak (global) en iyi çözümü değil en iyi çözümlerin listesini bile verebilir,
9. Karar değişkenlerini kodlayarak eniyilemeyi kodlama dünyasında yapar,
10. Genetik sayı sistemine göre üretilen sayılarla çalışır. Bunlar deney verileri veya analitik fonksiyonlar olabilir,

Genetik algoritmanın bazı mahsurları da şunlardır:

1. GA'lerde belirsizlik yöntemleri kullanılır ve algoritmanın işleyişinde belirginlik yoktur. Bunun sonucunda genetik algoritma modeli değişik defalarca kullanılınca birbirinden biraz farklı olan sonuçlar verebilir. Halbuki doğrusal, dinamik veya eğrisel (doğrusamaz) örgün en iyileme yöntemlerinin hepsi belirgin yöntemlerdir. İşleyişlerinde rastgelelik yoktur ve vardıkları sonuçlar aynı sorun ve şartlar altında hep aynıdır.
2. En iyi çözümün bilinen çözümler arasında göreceli olmasıdır. Ulaşılan çözümün en iyi olup olmadığının kontrol edilmesine izin vermeyebilir. Bu nedenle, genetik algoritmalar çözümün kesinlikle ne olabileceğinin bilinmemesi durumunda kullanılır. Doğal olarak evrim algoritmaları en iyi çözüm için ne zaman duracaklarını bilemezler. Bir durma kriteri belirlenmelidir.

3. Klasik eniyileme yöntemlerinde bir yön veya sistematik izin takip edilmesi gerekmesine karşı, genetik algoritma karar uzayının tamamen rastgele olarak tarar.



Şekil 3.2: Genetik Algoritma Akış Şeması

3.3.4 Genetik Algoritmanın Operatörleri

Basit bir genetik algoritma belirli aşamalardan gelmektedir. Sorunun çözümünde karşılaşılan ilk sorun; kromozomların oluşturulması ve şifreleme türünün seçimi ile ilgilidir. Bunun için, şifreleme (encoding) işlemi ile birlikte çaprazlama (crossover) ve mutasyon (değişim) gibi genetik algoritmanın temel operatörlerinin açıklanması

gerekmektedir. Çaprazlama ve mutasyon GA'nın en önemli kısımlarıdır. Performans bu iki operatörlerden etkilenir. Diğer soru ise çaprazlama için ebeveynlerin nasıl seçileceğidir. Bu birçok şekilde yapılabilir, ancak temel düşünce daha iyi olan ebeveynlerin seçilmesidir. Böylece daha iyi birey üretilebilir. Yeni dölle oluşturulan popülasyonda, son popülasyondaki en iyi kromozomun kaybolacağı da düşünülebilir. Bunun önlenmesi için seçkinlik (elitizm) kullanılır. Bu durumda, en azından iyi bir sonuç, hiçbir değişiklik olmadan yeni popülasyona kopyalanır, böylece bulunan en iyi sonuç korunmuş olur. Standart bir genetik algoritmanın bu aşamaları aşağıda anlatılmıştır:

3.3.4.1 Kromozomun Şifrelenmesi

Bir problemi GA ile çözmeye başladığımızda karşılaşacağımız problemlerden bir tanesi, kromozomların kodlanmasıdır. Kromozomların kodlanması probleme göre değişmektedir. Kullanılan kodlama yöntemleri:

1. İkili kodlama

Bu yöntem ilk GA uygulamalarında kullanıldığı için hâlâ en çok kullanılan yöntemlerdendir. Burada her kromozom, 0 ve 1'lerden oluşan bit dizisidir ve ikili diziyle ifade edilir. Bu dizideki her bit, çözümün bir özelliğini taşır. Dizinin tümü ise bir sayıya karşılık gelir.

İki Kodlama için kromozom örneği;

Kromozom A 101110010110

Kromozom B 010110100000

İkili kodlama, hatta küçük boyutlu problemler için çok büyük kromozom vektörü gerektirmektedir. Çözülecek probleme bağlı olarak çeşitli şifreleme yöntemleri vardır.

2. Permutasyon kodlama

Bu kodlama, Gezgin Satıcı Problemi (GSP) ve iş sıralama problemleri gibi permutasyon problemlerinde kullanılır. Burada her kromozom bir numaralar dizisidir.

Permutasyon kodlama için kromozom örneği;

Kromozom A 4528557956325

Kromozom B 7523654921354

Bu problemlerde olan ikili düzende kodlama kullanılabilir. Örneğin, her onluk sayı için ikili düzendeki sayılar yazılır.(Yapısına göre bilgisayarlar verileri ikili biçimde eşlemektedir.)

0 için 000

1 için 001

2 için 010

... ..

7 için 111

Bu durumda 8 genden (örneğin 8 değişken) oluşan permütasyon kodlamalı kromozomlarımız 24 genden oluşacaktır ve her üç gen bir değişkeni ifade edecektir.

3. Değer kodlama

Bu kodlama, gerçel gibi kompleks sayıların yer aldığı problemlerde kullanılır. Bu tür problemler için ikili kodlama çok zordur. Burada her kromozom bazı değerler dizisidir. Bu değerler ise problemle ilişkilidir; örneğin gerçel sayı, karakter veya kompleks nesnelere olabilir. Bu kodlama bazı özel problemler (örneğin bir yapay sinir ağının ağırlık katsayılarının bulunması) için çok uygundur.

Değer kodlama için kromozom örneği;

Kromozom A 2.872 5.921 3.612 1.847

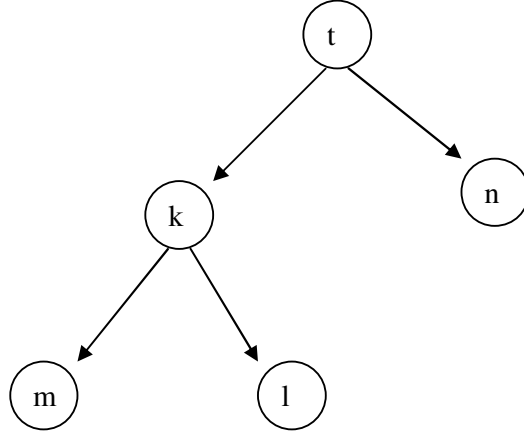
Kromozom B ABFİRFYJTALKTAILAIEE

Kromozom C (geri), (sol), (ileri), (sağ)

4. Ağaç kodlama

Ağaç kodlama genellikle genetik programlamada programlar ve ifadeler oluşturmak için kullanılır. Ağaç kodlamada her kromozom, adından da anlaşıldığı gibi nesnelere ve nesnelere arası işlemleri içeren bir ağaç yapısından oluşur. Ağaç kodlama, program geliştirmek için uygundur. Örneğin, LISP ve Prolog gibi programlama dillerinde

ağaçtan yapısal olarak sık bir biçimde kullanılır. Ağaç kodlamada çaprazlama ve mutasyon çok kolay bir şekilde uygulanabilir. Ağaç kodlama için kromozom örneği;



Şekil 3.3: Ağaç Kodlama İçin Kromozom Örneği

3.3.4.2 Eşleşme Havuzu Oluşturma

Eşleşme havuzu, bireylerin uygunlukların değerlendirildiği ve uygun olmayan bireylerin yerine uygun olanların kopyalandığı havuzdur. Burası eşlerin birleştirildiği alandır. Bir nesli oluşturan topluluktaki bireylerin çift sayıda olması, genetik algoritma işlemlerinin uygulanabilirliği açısından önemlidir.

3.3.4.3 Kopyalama (Üretim)

Bireylerin uyum faktörleri hesaplandıktan sonra, bu değeri 0.5 'in altında olan bireyler, o topluluk için güçsüz kabul edildiğinden bu bireyler topluluktan atılır. Havuzdan atılan her bir zayıf bireye eşit sayıda en güçlü bireylerden birer adet kopyalanır. Bu durumda, bir topluluğu meydana getiren toplam birey sayısı değişmez. Eğer, bir toplulukta uyum faktörü değeri 0.5 'in altında birey yoksa, bu durumda o topluluk için en zayıf olan, yani uyum faktörü değeri en düşük olan birey topluluktan çıkartılır. Bu bireyin yerine yine en güçlü, yani uyum faktörü en yüksek olan birey kopyalanır. Kopyalama işlemi tamamlandıktan sonra eşleşme havuzunda toplanan bu bireyler rastgele eşleştirilir. Bir sonraki adımda, eşleşen bireyler arasında "çaprazlama" ve "değiştirme" işlemleri yapılarak bireylerin genleri değiştirilir ve böylece farklı bireyler elde edilir.

3.3.4.4 Çaprazlama (Gen Takası)

Yaşamına devam eden ve eşleşme havuzunda toplanan bireyler rastgele eşleştirildikten sonra, çaprazlama operatörü kullanılarak yeni nesil oluşturulur. Çaprazlama işlemi üç adımda gerçekleştirilir. Önce toplumdan ikişer birey rastgele seçilir. Daha sonra seçilen bireylerin genleri arasında çaprazlama işlemi yapabilmek için rastgele bir konum belirlenir. Son olarak belirlenen konumdaki genetik bilgiler değiştirilerek çaprazlama işlemi tamamlanır.

Çaprazlama, genetik algoritmanın motoru kabul edilir. Basitçe olay, iki ebeveyn kromozom arasında belirlenen parçaların takasıdır. Gen takası toplumda çeşitliliği yani iyi özelliklerin bir araya gelmesini kolaylaştırarak en iyiye yaklaşmayı sağlar. Gen takası olasılığı 0.5-0.1 aralığında tavsiye edilir (**Michalewicz, 1996**).

Değişik çaprazlama türleri aşağıda verilmiştir:

a) Tek kesimli çaprazlama (tek noktalı gen takası)

b) Çift kesimli çaprazlama (çift noktalı gen takası)

c) Çok kesimli çaprazlama

d) Tekdüze çaprazlama (üniform çaprazlama)

e) Tersleme

f) Karıştırmalı çaprazlama

g) Ara birleşmeli çaprazlama

h) Doğrusal birleşmeli çaprazlama

i) Aritmetik çaprazlama

1) İkili kodlanmış kromozomlarda

a) Tek noktalı gen takası: Bir çaprazlama noktası seçilir. Bu noktaya kadar olan bitler (genler) 1. ebeveynden, geriye kalanlar ise diğerinden alınarak bir kromozom oluşturulur.

Kromozom-1: 11011 | 00100110110

Kromozom-2: 11011 | 11000011110

Döl-1: 11011 | 11000011110

Döl-2: 11011 | 00100110110

b)Çift noktalı gen takası: Burada iki kırılma noktası seçilir. İlk noktaya kadar olan bitler 1. ebeveynden, iki nokta arasındaki bitler 2. ebeveynden, kalanlar ise tekrar 1. ebeveynden yeni kromozoma kopyalanır.

$$11001011 + 11011111 = 11011111$$

c)Tek biçimli (uniform) gen takası : Bu çaprazlama biçiminde bitler rastgele şekilde her iki ebeveynden yeni kromozoma kopyalanmaktadır.

$$11001011 + 11011111 = 1011111$$

d)Aritmetik gen takası: Yeni bir kromozom oluşturmak için değişik aritmetik işlemler uygulanır.

$$11001011 + 11011111 = 11001001 \text{ (AND) VEleme}$$

2) Permütasyon kodlanmış kromozomlarda

Çaprazlama: Bir çaprazlama noktası seçilir. Bu noktaya kadar olan ifade birinci ebeveynden, kalan ise diğer ebeveynden kopyalanarak yeni kromozom oluşturulur.

$$(123456789) + (453689721) = (123459721)$$

3) Değer kodlanmış kromozomlarda

Çaprazlama: İkili kodlanmış kromozomlarda yer alan tüm takas türleri burada da uygulanabilir.(1. Tek noktalı gen tabakası, 2. Çift noktalı gen tabakası, 3. Tek biçimli takas, 4. Aritmetik takas.)

4) Ağaç kodlanmış kromozomlarda

Çaprazlama: Her iki ebeveyn de birer takas noktası seçerek parçalanır. Bu noktaların altındaki kısımlar, aralarında değiştirilerek yeni bireyler oluşturulur.

3.3.4.5 Mutasyon (Gen Değişimi = Rakam Değişimi)

Değişim, bir birey üzerinde rastgele belirlenen genlerin karşıt genlerle yer değiştirilmesi ile yapılan işlemdir. Amaç, verilen belli bir değişim değeriyle 0 olan genleri 1, 1 olan genleri de 0 yaparak yerel en iyilerin dışında mümkün olabilecek en iyileri de araştırmaktır. Rakam değişimleri ile GA'lar ikinci türden hedef fonksiyonu yüzeyini araştırır. Değişim değerleri, olaya rastgeleliğin hakim olmasını önlemek için değişime tabi tutulacak genlerin sayısı küçük tutulur.

Değiştirme görünüşte genetik algoritmanın dayanak noktasıdır, ancak etkisi tek bir çözüm üzerindedir. Çok düşük bir değiştirme olasılığı toplumda bazı özelliklerin kaybolmasına neden olabilir. Bu da en iyi sonuçların bulunmasını engeller. Ancak yüksek bir değiştirme olasılığı da eldeki çözümleri bozarak sonuca ulaşmayı zorlaştırır. Değiştirmenin olasılıkları için kesin bir sayı yoktur. Genelde, her iterasyonda bireylerin %1'i ile %0,1'i miktarında rakam değişimi yapılır. Değiştirme olasılığı 0.01-0.001 aralığında tavsiye edilir[13.4]. En son iterasyonda artık rakam değişimi yapılmaz. Genel olarak, en iyi çözümlerinde de rakam değişikliğine müsaade edilmez.

1) İkili kodlanmış kromozomlarda

Mutasyon işlemi, problemin popülasyondaki çözümlerinin yerel optimuma düşmesini engellemek için kullanılır. Mutasyon yeni bireyi rastgele değiştirir. İki şifreleme için, rastgele seçilen birkaç gen 0'dan 1'e veya 1'den 0'a değiştirilebilir, yani seçilen genler terslenir. Mutasyon da çaprazlama gibi şifrelemeye bağlıdır.

Orijinal birey : 1101111000011110

Mutasyona uğramış birey : 1100111000011110

2) Permütasyon kodlanmış kromozomlarda

Mutasyon, iki gen seçilir ve bunların yerleri değiştirilir.

(123456897) => (183456297)

3) Değer kodlanmış kromozomlarda

Mutasyon, seçilen değerlerden küçük bir sayı çıkartılarak veya eklenerek yeni bir kromozom oluşturulur.

(1.29 5.68 2.86 4.11 5.55) => (1.29 5.68 2.73 4.22 5.55)

4) Ağaç kodlanmış kromozomlarda

Mutasyon, seçilen düğümlerdeki numaralar veya işlemler değiştirilir.

3.3.5. Genetik Algoritmanın Parametreleri

Genetik Algoritmalarda

-Çaprazlama ve mutasyon operatörlerinin oranının (veya olasılığının),

-Aynı zamanda popülasyon büyüklüğü ve

-Maksimum jenerasyon sayısının büyüklüğü

algoritmanın çalışmasını güçlü şekilde etkilemektedir. Bu parametreler aşağıda kısaca açıklanmaktadır:

3.3.5.1 Çaprazlama Olasılığı :

* Gen Takasının hangi sıklıkta olacağını belirler.

* Çaprazlama yapılmadığında, kromozom ebeveynlerin tam bir kopyasıdır. Çaprazlama gerçekleştiğinde ise kromozom, ebeveynlerin kromozomların parçalarından oluşmaktadır.

* Çaprazlama oranı %100 olduğunda bütün döller (kromozom=birey) takas yolu ile elde edilir. Bu oran %0 ise yeni neslin (jenerasyonun) tümü eski toplumdaki (popülasyondaki) kromozomların kopyası olmaktadır.(Ama bu yeni neslin aynı olduğu anlamına gelmez.)

* Çaprazlama oranının yüksek olduğu durumlarda problem uzayı çok hızlı bir şekilde araştırılır.

* Artışın devam ettiği durumlarda, iyi bireylerin yapılarının bozulmasından dolayı algoritmanın performansı düşmektedir.

* Düşük çaprazlama oranının da ise az miktarda yeni bireyler nesle dahil olduğundan tekrar üreme operatörü etkili olmakta ve çözüme erişim hızında güçlü düşüş görülmektedir.

* Çaprazlama işlemi, yeni kromozomların eskilerin parçalarından oluşması ve yenilerin daha iyi olması umuduyla yapılır. Bununla beraber, toplumun belli kısımlarını bırakmak bir sonraki nesil için iyidir.

3.3.5.2 Mutasyon Olasılığı:

- * Değişimin hangi sıklıkta olacağını belirler.
- * Mutasyona uğramayan kromozom (döl), çaprazlamadan ya da kopyalamada sonra hiçbir değişim olmadan alınmaktadır; aksi halde kromozomun bir kısmı değişir.
- * Mutasyon oranı %100 olduğunda, tüm kromozom değişikliğe uğrar. Bu oran %0 olduğunda ise kromozomların yapısında bir değişiklik görülmez.
- * Mutasyon oranının düşük olması problem uzayının tamamının araştırılmasını engellediğinden yerel çözümlerde takımlara neden olabilir.
- * Değişim oranının yükselmesi ise probleme rastgelelik kazandırır ve bireyleri sürekli değişir.
- * Mutasyon aslında çözümün yerel sonuçlara takılmasını önlemektedir.
- * Gerçek yaşamda çok az rastlanan mutasyon, Genetik Algoritmalarda genellikle 0.001-0.01 aralığındaki sıklıkla kullanılması önerilmektedir.
- * Mutasyon GA'nın yerel ekstremuma (tepelere, sonuçlara, optimumlara) takılmasını engellemek için gerçekleştirilir. Fakat çok sık kullanıldığında GA, körüne aramaya dönüşebilir.

3.3.5.3 Popülasyonun (Topluluğun) Büyüklüğü:

Çaprazlama ve mutasyon oranlarının dışında diğer önemli parametrelerden biri de popülasyonun büyüklüğüdür.

- * Popülasyonun büyüklüğü, toplumda kaç kromozom olduğunu ifade eder.
- * Popülasyon büyüklüğünün de optimum değerlerden uzak olması algoritmanın performansını etkilemektedir.
- * Nesildeki bireyler sayısı az ise uzay yeterince araştırılmadığından yerel optimum noktaya yakalanma olasılığı yüksektir. Bu durumda çaprazlama işlemi için yalnız birkaç olasılık olduğundan problem alanının sadece küçük bir kısmı araştırılabilecektir. Bu sayı aşırı arttığında ise neslin evrimleşmesi oldukça uzun zaman içerisinde gerçekleştirildiğinden algoritma yavaşlayacaktır

Araştırmalar popülasyon boyutu için, şifrelemeye bağlı olarak, bir üst sınırın belirlenmesi gerektiğini göstermektedir. Bu değer üstüne çıkıldığında ise

problemin çözümünde bir iyileşme görülmeyecektir. Goldberg, ikili kodlamalı durumlarda (P) popülasyon büyüklüğü ile (ku) kromozom uzunluğu arasında aşağıdaki bağıntıyı önermektedir.

$$P=1,65 * 2^{0,2*ku}$$

Yapay Zekada kullanılan optimizasyon yöntemlerinden olan GA parametrelerinin kendilerinin de optimizasyona gerek duyduğu görülmektedir. Bu parametrelerle ilgili çalışmaların bazı değerleri tabloda verilmiştir.

Tablo 3.1: Genetik Algoritma İçin Önerilen Kontrol Parametre Değerleri

Parametreler	Negnevitsky	D.Jong (1975)	Mchalewicz	Schaffer (1989)	Grefenstette (1986)
Popülasyon Büyüküğü	50	50-100	50-100	20-30	30
Çaprazlama Oranı	0.7	0.6	0.5-0.1	0.75-0.90	0.95
Değişim Oranı	0.007-0.01	0.001	0.001-0.01	0.005-0.01	0.01

Dolayısıyla problemin niteliğine uygun olarak kontrol parametrelerinin uyarlanması gerekmektedir. Diğer yandan operatörlerin de nesile bağılı olarak adaptasyonu önemli etmenlerdir.

3.3.5.4 Seçim

Kromozomlar çaprazlamaya ebeveyn olmak için topluluktan (popülasyondan) seçilirler. Burada sorun, kromozomların seçimi ile ilgilidir. Darwin'in evrim teorisine göre, en iyi olanlar kurtulmalı ve yeni dölü oluşturmalıdır. En iyi kromozomları seçmenin birçok yolu vardır. Bunlar aşağıda verilmiştir:

- 1- Rulet tekeri (Turnike) seçimi (Roulette Wheel Selection)
- 2- Baltzman seçimi (Baltzman Selection)
- 3- Turnuva seçimi (Tournament Selection)

- 4- Sıralama seçimi (Rank Selection)
- 5- Sabit durum seçimi (Steady State Selection)
- 6- Rastgele seçimi
- 7- Ağırlıklı Rastgele seçimi
 - 7.1 Mertebe Ağırlıkları
 - 7.2 Hedef Ağırlıkları

Bunlardan bazılarıdır. Şimdi kısaca bunlardan bir kısmının açıklanması:

1) Rulet Teker (veya Turnike) Seçimi:

Bu yöntemde seçilme işlemi bireylerin uygunluk değerlerine göre yapılmaktadır. Fakat uygunluk değeri en büyük olanın seçileceği garanti edilmez, yalnız seçilme şansı daha fazla olacaktır. Bu yöntemde tüm bireylerin uygunluk değerleri bir tabloya yazılır ve toplanır. Sonra uygunluk değerleri, toplama bölünerek, bireylerin [0,1] aralığında seçilme olasılıkları belirlenir. Sayıların hepsi bir tabloda tutulur. Seçilme olasılıklarının tuttuğumuz tablodaki sayılar birbirine eklenerek rastgele bir sayıya kadar ilerlenir. Bu sayıya ulaşıldığında ya da geçildiğinde son eklenen sayının ait olduğu seçim çözülmüş olur. Bu yöntem rulet tekerleği seçimi ismi, bir dairenin dilimlenmesine benzediği için verilmiştir. Rulet tekeri seçimi çözümlerin uygunluk değerlerinin pozitif olmasını gerektirir. Çünkü olasılıkların negatif olması çözümün seçilme şansının olmadığını göstermektedir. Çoğunluğunun uygunluk değeri negatif olan bir toplumda yeni nesiller belli noktalara takılıp kalabilir. Bu nedenle uygunluk ölçeklendirme (fitness scaling) yöntemine başvurulur.

*Uygunluk fonksiyonunu f ile ifade edersek ölçeklendirmeden dönülecek k değeri;

$$k = mf + n$$

şeklinde verilebilir. Burada;

m : 1,0-2,0 aralığındaki değerler kullanılır

n : uygunluk fonksiyonunun tahmini minimum noktasını pozitif yapacak bir değerdir.

Ebeveynler uygunluklarına göre seçilirler. Kromozom daha iyi oldukça seçilme şansları da artmaktadır. Popülasyondaki tüm kromozomların bulunduğu ve her birinin uygunluk fonksiyonuna bağlı olarak yer aldığı bir rulet tekeri düşünülün.

Ardından, bir msket atılsın ve kromozom seilsin. Daha byk uygunluęa sahip olan kromozom daha fazla seilecektir.

2) Sıralama Seimi:

Rulet tekeri seim, uygunluklar ok farklıysa problemlere yol aacaktır. rneęin, en iyi kromozomun uygunluęu, rulet tekerinin %90'ında ise, dięer kromozomlar ok az seilme řansına sahip olacaklardır.

Sıralama seimi nc poplasyonu sıralamakta ve ardından her kromozomun bu sıralamada uygunluęunu aramaktadır. En kts 1 uygunlukta, ikinci kts 2 uygunlukta vb; en iyisi ise N uygunlukta olacaktır (Burada N, poplasyondaki kromozom sayısıdır). Bunun ardından btn kromozomlara seilme řansı doęacaktır. Fakat bu yntemde en iyi kromozomlar, dięerlerinden fazla farklı olmadığından zme yaklaşma yavaş olacaktır.

3) Sabit Durum Seimi:

Bu ynteme gre ebeveynlerin seimi iin kromozomların byk paraları bir sonraki nesile tařınmalıdır. GA daha sonra řu řekilde ilerler. Her nesilde yeni bir dl oluřturmak iin birkaç kromozom seilir (byk uygunlukla iyi olanlar). Daha sonra az uygunlukla kt olan bazı kromozomlar atılır ve yeni dl onların yerine getirilir. Poplasyonun geri kalanı deęiřtirilmeden yeni dle aktarılır.

3.3.5.5 Sekinlik (ELİTİSM)

aprazlama ve mutasyonla yeni poplasyon retildięi zaman, en iyi kromozom kaybedilecektir. Sekinlik, nce en iyi kromozomu (ya da birkaç en iyi kromozomu) yeni poplasyona kopyalar. Geri kalanı dięer yntemlerle devam ettirilir. Sekinlik, bulunan en iyi sonucun kaybını nledięinden GA'nın performansını oldukça hızlı bir řekilde artırır.

3.3.6. Hesapta İzlener Yol

Eniyileme sorununa Genetik Algoritma ynteminin uygulanmasında takip edilen iřlem adımları ařaęıda aıklanacaktır:

3.3.6.1 Gen Haritasının Hazırlanması

Genetik Algoritma, uygulama öncesinde on veya iki tabanlı sayı sistemlerinden birisi ile sorunun çözümü için genetik sayı sistemine karar verilir. GA sayı sisteminde özel isimleri olan dört tane sayı türü bulunmaktadır. Bunlar bit (gen veya hane), dizi, kromozom ve toplum sayılarıdır.

1. GA sayı sisteminde karar değişkeni sayılarının her bir rakamına bu çalışmada “bit (gen)” ifadesi kullanılacaktır. Bu durumda, GA sayı sisteminde en küçük eleman bit (gen)’ dir. Her gende ikili sayı sisteminde ya 1 ya da 0 , onluk sayı sisteminde ise 0 ile 9 arasında 10 rakamdan sadece bir tanesi bulunur. Böylece her karar değişkeni bir sayı sistemine göre kodlanır. Genellikle bilgisayarın çalışma sistemiyle aynı olması bakımından iki tabanlı sayı sistemine göre kodlama tercih edilir.
2. Belli sayıda bit’lerden (gen’lerden) meydana gelen baytlar “dizi” olarak isimlendirilir. Bu dizinin onluk sayı sistemindeki karşılığı bir tasarım değişkeninin set (araştırma uzayı) içindeki sıra numarasını ifade eder. Bir problemdeki tasarım değişkeninin sayısı kadar dizi oluşturulur.
3. Tasarım değişkeninin sayısı kadar oluşturulan bu diziler yan yana yazılarak meydana gelen diziler toplamına “birey” veya “kromozom” denir.
4. İki veya daha fazla kromozomun bir araya gelmesi ile oluşturulan sayıların hepsine birden (çözüm kümesi = araştırma uzayı) “toplum” ismi verilir.

Bireyin uzunluğunun nasıl hesaplandığına örnek verecek olursak:

Tasarım değişkeni sayısı : tds

Bir dizinin uzunluğu (bir dizideki gen sayısı) : du

Bireyin uzunluğu : bu = tds * du’ olur.

Genetik Algoritmada araştırma uzayı, problemin muhtemel çözüm kümesi olup bu küme tasarımcı tarafından problemin çözümüne başlamadan önce belirlenmektedir. Araştırma uzayının küçük olması yani muhtemel çözüm kümesinin eleman sayısının az olması çözümün hızlı olmasını sağlar. Bu çözüm kümesinin eleman sayısının kaç tane olması tasarımcıya bağlıdır. Ayrıca bir dizinin uzunluğu da muhtemel çözüm kümesinin eleman sayısı ile bağlantılıdır. Araştırma uzayının elemanları sıraya konur

ve her birine sıra numarası verilir. Her bir eleman o sıra numarasına atanarak bireyler oluşturulur.

Muhtemel çözüm kümesinin eleman sayısına nasıl karar verildiğini görmek için örnek verecek olursak:

Adım 1 : Bir dizinin uzunluğu 4 olsun.

Adım 2 : Gen (bit = hane) sayısı 4 olan ve onluk sayı sistemine dönüştürüldüğünde en büyük sayıyı ifade eden $(1111)_2$ sayısı esas alınır. Burada $(1111)_2$ sayısına kadar olan bütün ikili sayı sistemindeki sayılar onluk tabanda yazılır. $(1111)_2$ sayısının onluk tabana göre karşılığı:

$$(1111)_2 = (1 * 2^0 + 1 * 2^1 + 1 * 2^2 + 1 * 2^3)_{10} = (15)_{10} \text{ 'dir.}$$

Adım 3 : Ancak, $(0000)_2$, onluk sayı sistemine göre "0" a eşit olmasına rağmen sonuç olarak bir sayıyı ifade ettiği, açıkçası hesap 1'den değil de 0'dan başladığı için gen haritası

$$A_{k(j)} = S_{e(i+1)} \quad (3.1)$$

olur. Buradaki A_k , ikili sayı sistemine göre dizi ve S_e muhtemel çözüm kümesindeki elemanların sayısıdır. "i" ise "j" grubu gösteren alt seriye karşılık gelen onluk taban değeridir. Sonuç olarak, $(1111)_2 = (15)_{10}$ ve $(0000)_2 = (0)_{10}$ olmasından dolayı muhtemel çözüm kümesindeki eleman sayısı $15 + 1 = 16$ 'dır. Burada belirtilen ifadelerin daha iyi anlaşılması için örnek aşağıda verildi:

Tablo 3.2: Araştırma Uzayının Belirlenmesi

ARAŞTIRMA UZAYININ BELİRLENMESİ			
İkili Sayılar (Ak(j))	Onluk Taban Karşılığı	Set'teki Sıra Numarası (Se(i+1))	Eleman Enkesitleri (Araştırma Uzayı) (cm2)
0000	0	1	13,20
0001	1	2	16,40
0010	2	3	20,10
0011	3	4	23,90
0100	4	5	28,50
0101	5	6	33,40
0110	6	7	39,10
0111	7	8	45,90
1000	8	9	53,80
1001	9	10	62,60
1010	10	11	72,70
1011	11	12	84,50
1100	12	13	98,80
1101	13	14	116,00
1110	14	15	134,00
1111	15	16	156,00

Bu örnekte, bir düzlem çelik çerçevenin tasarım değişkenleri için varsayılan uygun 16 adet kesit alanı değerleri ele alınrsa;

$S_e = \{ 13.20, 16.40, 20.10, 23.90, 28.50, 33.40, 39.10, 45.90, 53.80, 62.60, 72.70, 84.50, 98.50, 116, 134, 156 \} (cm^2)$

Burada (0000)₂ değeri S_e listesinin 1. sırasındaki elemanı, yani 13.20 cm²'yi ifade etmektedir. Genetik Algoritma ile optimizasyon işlemi için “dizi uzunluğu” ve buna bağlı olarak “muhtemel çözüm listesi” oluşturulduktan sonra “tasarım değişkenleri” bu liste içinden seçilir. Problemden kaç adet tasarım değişkeni varsa, kümeden o kadar sayıda rastgele eleman seçilir. Ve bu elemanları ifade eden ikilik tabandaki karşılıkları yan yana yazılır. Böylece birey oluşturulmuş olur.

3.3.6.2 Optimum Tasarım Probleminin Tanımlanması

Gen haritası oluşturulan ve bireyleri rastgele belirlenen bir optimizasyon probleminde bir sonraki adım, amaç fonksiyonunu ve bu amaç fonksiyonunun sağlanması gereken şartları oluşturmaktır. Bu işlemler:

a) Amaç fonksiyonunun oluşturulması

İnşaat Mühendisliği'nde kapsama alanına giren konular, genellikle belli kısıtlamalar altında yapının ağırlığının veya maliyetinin enküçüklemesini (minimizasyonunu) kapsar.

Çelik yapıların tek bir malzemeden yapılıyor olmaları nedeniyle belli kısıtlamalar altında ağırlığının veya hacminin minimize edilmesi yeterlidir. Çünkü, maliyet ağırlıklarıyla doğru orantılıdır ve bu nedenle maliyet hesabının yapılmasına gerek yoktur. Ağırlığı az olan yapı maliyeti daha düşük olan yapıdır.

Betonarme yapıların optimizasyonu için ağırlığın değil maliyetin enküçüklemesi gerekir. Betonarme, kompozit bir malzeme olduğundan, davranış özellikleri, ağırlıkları ve fiyatları birbirinden çok farklı malzemelerden oluşur. Bundan dolayı, betonarme yapının ağırlığının az olması o yapının maliyet açısından optimum olduğunu ifade etmez.

Sonuç olarak, sorunun yapısına ve o problemin çözümünden beklenilene göre amaç fonksiyonunun belirlenmesi gerekir. Bundan sonraki aşama ise kısıtlamaların belirlenmesidir.

b) Kısıtlamalar

İnşaat Mühendisliği'nde çözümü istenen eniyileme problemleri kısıtlamalı problemlerdir. Kısıtlamaların probleme katılması sorunu daha karmaşık hale getirmektedir. Genetik algoritma, kısıtlamasız eniyileme problemleri için oldukça uygundur. Genetik Algoritmayı optimizasyon probleminde kullanabilmek için kısıtlamalı problemleri kısıtlamasız problemlere dönüştürmek gerekir. Optimizasyon dönüştürme işlemi genellikle iç ve dış ceza fonksiyonlarının kullanılması ile yapılmaktadır. Ama böyle dönüştürmeler ardışık aramalar için elverişlidir. Genetik algoritma ise muhtemel çözüm kümesinde rastgele oluşturduğu nesillerden faydalanarak arama işlemi yapar. Bundan dolayı, ceza (engel) fonksiyonlarının kullanıldığı geleneksel dönüştürmeler Genetik algoritmalar için uygun değildir. Genetik algoritma, dönüştürme işlemi için genellikle “normalize” edilen kısıtlamaların ihlal edilmesi esasına dayanan bir ifade kullanmaktadır. Kısıtlamaların normalize edilmiş şekilleri aşağıdaki gibi ifade edilir:

$$\rho_{i,j}(x) = \frac{M_{i,j}}{M_{i,ju}} - 1 \leq 0, \quad i = 1, \dots, p \quad \text{ve } j = 1, \dots, nm \quad (3.2)$$

$$\rho_{i,j}(x) > 0 \quad \text{ise} \quad c_{i,j} = \rho_{i,j}(x) \quad (3.3a)$$

$$\rho_{i,j}(x) \leq 0 \quad \text{ise} \quad c_{i,j} = 0 \quad (3.3b)$$

alınmak üzere;

$$C = \sum_{i,j=1}^{p,nm} c_{i,j} \quad (3.4)$$

dir. Sonuç olarak, kısıtlamalar probleme bir C katsayısı olarak dahil edilmiş olur. Buradaki ifadelerde,

p : kısıtlama sayısı,

nm : eleman sayısı,

$M_{i,j}$: j. elemandaki i. kısıtlamanın aldığı değer,

$M_{i,ju}$: j. elemandaki i. kısıtlamanın üst sınır değeri,

C : zorlanma katsayısı, anlamındadır.

3.3.6.3 Genetik Algoritmada Tablo İşlemleri

- Eniyileme probleminin tasarım değişkenleri daha önce oluşturulmuş olan araştırma uzayından rastgele seçildikten sonra, amaç fonksiyonu bu tasarım değişkenleri ile çözülür.
- Elde edilen amaç fonksiyonu değeri çözüm tablosunda yerine yazıldıktan sonra bu amaç fonksiyonunun sağlaması gereken kısıtlamalar, bir önceki kısımda anlatıldığı şekilde, normalize edilerek bir C katsayısı olarak tabloda gösterilir.
- Bütün bu işlemler, bir tabloyu oluşturan bir neslin rastgele seçilen birey sayısı kadar tekrarlanır.

Genetik algoritma tarafından gerekli olan uygunluk kriterini elde etmek için kısıtlamalı amaç fonksiyonunun kısıtlamasız probleme dönüştürülmesi gerekmektedir. Bu durumdan sonra, kısıtlamasız amaç fonksiyonu, her bir birey için, C zorlanma katsayısı da hesaba katılarak, aşağıdaki şekilde ifade edebiliriz:

$$\Phi(x) = f(x) * (KC + 1) \quad (3.5)$$

Burada;

$\Phi(x)$: Kısıtlamasız fonksiyon

$f(x)$: Amaç fonksiyonu,

C : Zorlanma katsayısı,

K : Zorlanma katsayısının etkisini artırmak için kullanılan katsayı

anlamındadır.

Buradaki kısıtlamasız fonksiyon ($\Phi(x)$)'nin minimumu genetik algoritma tarafından bulunur. Bu yöntem, bireyler arasında seçim yapabilmek için bir koşulun kullanımını gerektirir. Kısıtlamasız amaç fonksiyonu $\Phi(x)$, her bir birey için hesapladıktan sonra uygun değerlerle değiştirilmesi gerekir. En fazla uygunluğa sahip olan birey en iyi demektir. Goldberg (1989)'e göre; minimizasyon problemleri için, $\Phi(x)$, büyük bir sabit değerden çıkarılmalıdır, böylece bütün uygun değerler pozitif olur ve etkili değerlerine göre bireylerin uygun değerleri elde edilir. Bu sabit değer, $\Phi(x)$ 'in bir

tablodaki enbüyük ve enküçük değerlerinin toplanmasıyla elde edilmektedir. Uyum derecesi için hesap şu şekilde yapılır:

$$F_{ud(i)} = [\Phi_i(x)_{\max} + \Phi_i(x)_{\min}] - \Phi_i(x) \quad (3.6)$$

Burada;

$F_{ud(i)}$: i bireyinin uyum derecesini,

i-indisi : bir nesildeki bireyi,

$\Phi_i(x)_{\max}$ ve $\Phi_i(x)_{\min}$: bütün bireylerden oluşan topluluk içinde kısıtlamasız (3.5) ifadesinin alacağı maksimum ve minimum değerleri gösterir.

Her bir bireyin uyum faktörü ise;

$$\frac{F_{ud(i)}}{F_{ort}} \quad (3.7)$$

ile hesaplanır.

Topluluğun ortalama uyum derecesi;

$$F_{ort} = \frac{\sum F_{ud(i)}}{n} \quad (3.8)$$

dir. Burada;

n : topluluktaki toplam birey sayısı

Topluluğun ortalama uyum derecesine (3.8) bakılarak her bir bireyin elenmesine veya bir sonraki nesil için eşleşme havuzuna kopyalanmasına karar verilmektedir.

Bu işlemlerden sonraki aşama, bir sonraki nesil için yeni bir topluluk oluşturmaktır. Oluşturulan bu topluluk bir önceki neslin çocukları olacaktır. Eşleşme havuzu oluşturulduktan sonra bireyler burada rastgele eşleşmekte ve çaprazlama ve değişim operatörleri uygulanmakta, böylece bir sonraki nesil meydana gelmektedir.

Genetik Algoritma ile eniyileme tasarım işleminin algoritmasının aşamaları:

a) Öncelikle, araştırma uzayından, problemin tasarım değişkeni sayısı kadar dizi rastgele seçilir. Sonra bu dizilerin bir araya getirilmesiyle (birleştirilmesiyle) bireyler

oluşturulur. Böylece n-bireyden oluşan başlangıç topluluğu (1.nesil) belirlenir. Genetik Algoritmanın daha sonraki aşamalarında yapılan eşleşme işleminin gerçekleştirilebilmesi için n-sayısının çift olması gerekir.

b) İkili tabanda kodlanan her bir tasarım değişkeni onluk sisteme dönüştürülerek listedeki sıra numarası bulunur ve o sıra numarasının ifade ettiği tasarım değişkeni değeri belirlenir.

c) Daha önce elde edilen tasarım değişkenleri yardımıyla kısıtlamalar hesaplanır ve amaç fonksiyonunun sayısal değeri bulunur.

d) Daha önceden belirtilen (3.3) ve (3.4) ifadeleri yardımıyla kısıtlamaların ihlal edilme katsayıları, (3.5) ifadesi kullanılarak kısıtlamasız amaç fonksiyonu her bir birey için hesaplanır. Daha sonra tabloda bu fonksiyonun enbüyük ve enküçük değerleri belirlenir.

e) (3.6) ifadesi kullanılarak bireylerin uyum dereceleri saptanır. (3.8) ifadesiyle ortalama uyum derecesi ve denklem (3.7) ile de her bir bireyin uyum faktörü hesaplanır.

f) Uyum faktörüne bağlı olarak bireyler eşleşme havuzuna kopyalanır.

g) Bireyler rastgele eşleştirilip çaprazlama işlemi uygulanarak yeni topluluk elde edilir.

h) Yeni topluluk, başlangıç topluluğunun yerine konularak b) adımından h) adımına kadar olan işlemler tekrarlanır.

Bu işlemlere yeni topluluğun %80 veya %85'i aynı bireyden teşkil olana kadar devam edilir. Bu oran arama işlemini sona erdirmek için yeterlidir ve elde edilen birey optimum çözümü temsil eder (Rajeev ve Krishnamoorthy, 1992).

4. DÜZLEMSEL ÇELİK ÇERÇEVELERİN OPTİMUM TASARIMI

Çelik yapıların tek bir malzemeden yapılıyor olmalarından dolayı genellikle belli kısıtlamalar altında yapının ağırlığının minimize edilmesi yeterlidir. Çünkü yapının maliyeti ağırlığıyla doğru orantılıdır ve bu nedenle maliyet hesabının yapılmasına ihtiyaç yoktur. Sonuç olarak, ağırlığı az olan yapı maliyeti daha düşük olan yapıdır. Bu çalışmada, düzlem çelik çerçevelerin optimum tasarımını gerçekleştirmek için kesit özelliklerini uygulamada hazır olan standart kesitlerden seçebilen bir basit genetik algoritma sunulmaktadır. Burada basit genetik algoritma kullanılarak yapı gerilme, stabilite ve deplasman sınırlayıcıları altında boyutlandırılmıştır.

4.1. Optimum Tasarım Problemi

4.1.1 Tasarım Değişkenleri

Çelik çerçeve sistemlerinin boyut optimizasyon problemlerinde elemanların kesit alanları tasarım değişkeni olarak düşünülür. Düğüm noktalarının koordinatları ve değişik elemanlar arasındaki bağlantılar sabit olarak alınır. Bundan dolayı boyutlandırma optimizasyon problemi, sadece önceden belirlenmiş standart kesitlerden alınarak eleman kesit alanlarının sınırlandırılmasıyla uygulamada faydalıdır.

4.1.2 Amaç Fonksiyonu

Düzlem çerçevelerin optimum tasarım problemi,

$$\min W(x) = \sum_{k=1}^{ns} A_k \sum_{i=1}^{mk} \rho_i L_i \quad (4.1)$$

olarak tanımlanabilir. Burada:

$W(x)$: amaç fonksiyonu,

ρ_i : i'inci çubuğun yoğunluğu,

L_i : i'inci çubuğun boyu,
 n_s : sistemdeki toplam grup sayısı,
 m_k : k grubundan toplam çubuk sayısı
olarak tanımlanır.

4.1.3 Kısıtlamalar

Çelik yapılarda stabilite sorunu önemli olduğu için bu çalışmada gerilme, burkulma ve yanal burkulmayı içeren bileşik gerilme durumu ele alınmıştır. Burada sınırlayıcılar **Rajeev, Krishnamoorthy, (1992)** tarafından önerilen ve kaynak **Goldberg (1989)** da kullanılan normalize edilen sınırlayıcıların ihlal edilmesi esasına dayanan bir ifade kullanılacaktır. Birleşik gerilme sınırlayıcısı TS 648'e göre hesaba katılmıştır. Sınırlayıcıların normalize edilmiş şekilleri aşağıda ifade edilmiştir.

Deplasman sınırlayıcılar,

$$g_j(x) = \frac{\delta_j}{\delta_{ju}} - 1 \leq 0 \quad j = 1, \dots, p \quad (4.2)$$

Birleşik gerilme sınırlayıcılar,

$$g_i(x) = \frac{\sigma_{eb}}{\sigma_{bem}} + \frac{C_m \sigma_b}{(1 - \frac{\sigma_{eb}}{\sigma_e}) \sigma_B} - 1 \leq 0 \quad (4.3)$$

ve

$$g_i(x) = \frac{\sigma_{eb}}{0.60 \sigma_a} + \frac{\sigma_b}{\sigma_B} - 1 \leq 0 \quad i = 1, \dots, nm \quad (4.4)$$

Yukardaki ifadelerden (4.3) ve (4.4) ifadeleri aksel basınç ve eğilme momenti etkisindeki çubuklar tarafından gerekmektedir. Aksel çekme kuvveti ve eğilme momentine maruz çubukların ise (4.4) ifadesini sağlamaları gerekmektedir. Eğer,

$$\frac{\sigma_{eb}}{\sigma_{bem}} \leq 0.15$$

ise (4.3) ve (4.4) ifadeler yerine

$$g_i(x) = \frac{\sigma_{eb}}{\sigma_{bem}} + \frac{\sigma_b}{\sigma_B} - 1 \leq 0 \quad (4.5)$$

ifadesi kullanılabilir.

Yukarıdaki formüllerde,

δ_j ve δ_{ju} : j'inci noktanın deplasmanı ve onun üst sınırı,

p : sınırlanmış deplasman sayısı,

σ_{eb} : yalnız aksenal basınç kuvveti altında hesaplanan gerilme,

σ_{bem} : yalnız basınç kuvveti etkisi altında müsaade edilecek basınç gerilmesi,

σ_B : yalnız eğilme momenti etkisi altında müsaade edilecek basınç eğilme gerilmesi

σ_e' : güvenlik katsayısına bölünmüş Euler gerilmesi,

σ_a : çeliğin akma gerilmesi,

nm : çerçevedeki eleman katsayısı,

C_m : aksenal basınç ve eğilmenin etkidiği sistemlerde, kolonun şeklini göz önüne alan bir katsayı. Bu uyum katsayısı yanal deplasmanın mümkün olduğu çerçevelerde 0.85'e eşittir(TS 648, 1980).

4.2 Hazırlanan Bilgisayar Programlarının Tanıtılması

Düzlem çelik çerçevelerin genetik algoritma yöntemi ile ağırlık optimizasyonunu yapılabilmesi amacıyla bir program hazırlanmıştır.

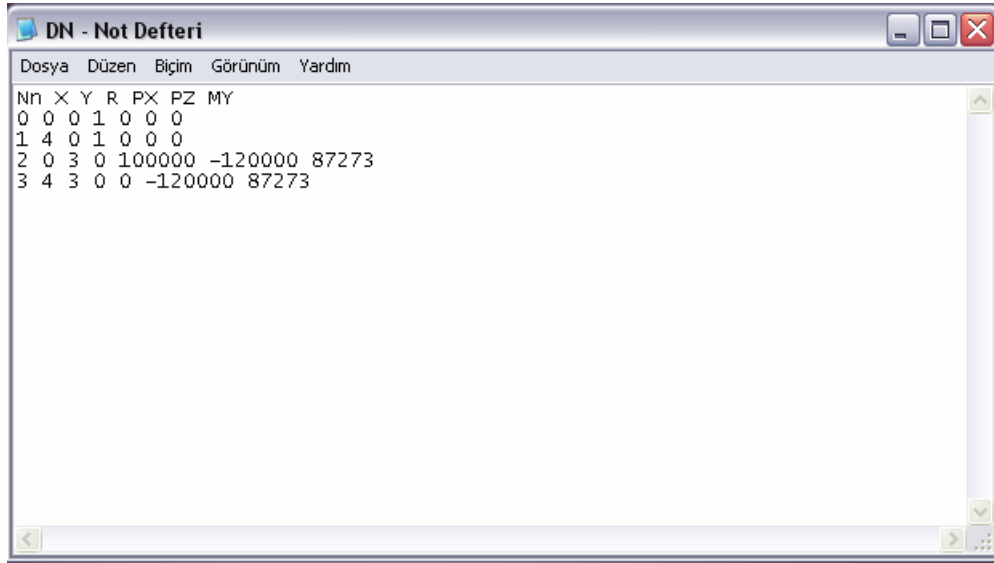
YZDOPT1 olarak isimlendirilen bu program, MATLAB programlama dili ile yazılmıştır. Çerçeve elemanlarının enkesit boyutları değiştirilerek sistem ağırlığını minimize eden optimum eleman kesitleri araştırılmıştır. Amaç fonksiyonu, çerçevenin ağırlık hesabını içermektedir. Burada TS 648 esaslarına göre birleşik gerilme kuralları kısıtlayıcı olarak belirlenmiştir.

YZDOPT1 Programı İçin Verilerin Hazırlanması

A) Veri Dosyalarının Hazırlanması

Hazırlanan Genetik Algoritma, analiz programı olarak **Yılmaz (2006)**, tarafından yapılmış olan Sonlu Elemanlar yöntemini kullanarak analiz yapan program kullanılmıştır. Dolayısıyla YZDOPT1 programının üreteceği veri dosyası modeli de bu analiz programına göre olacaktır. YZDOPT1'in başlangıç için ve diğer optimizasyon için gerekli bilgileri okuyacağı bazı veri dosyaları ilk olarak hazırlanır. Bunlar:

1. "DN.txt" dosyası: Düğüm noktalarının özelliklerini içeren veri dosyası Şekil 1.1'de gösterilmiştir.



Şekil 4.1: DN.txt Veri Dosyası

2. "EL.txt" dosyası: Eleman özelliklerini içeren veri dosyası Şekil 1.2'de gösterilmiştir.

Dosya	Düzen	Biçim	Görünüm	Yardım
hn	I	J	A1	IX E V Q
0	0	2	13.21	317.80 2100000.00 0.30 0
1	2	3	62.61	11770.00 2100000.00 0.30 0
2	3	1	13.21	317.80 2100000.00 0.30 0

Şekil 4.2: EL.txt Veri Dosyası

2. “EV.txt” dosyası: Yapı elemanları ile ilgili diğer özellikleri içeren veri dosyası Şekil 1.3’de gösterilmiştir.

Dosya	Düzen	Biçim	Görünüm	Yardım
EN	GN	B	YO	Sa
1	1	3.00	3.10	2400
2	2	4.00	3.10	2400
3	1	3.00	3.10	2400

Şekil 4.3: EV.txt Veri Dosyası

4. “PVERİ.txt” dosyası: Genel değişken özelliklerini ve yönetmeliklere göre hesaplanmış kısıtlamalarda kullanılacak bilgileri içeren veri dosyası Şekil 1.4’de gösterilmiştir.

pa	sett	ix	l1	l2	se	sbem	sb	wz	w
IPE120	13.21	317.80	300	400	2209.47	193.50	233.61	60.73	7.44
IPE140	16.43	541.20	300	400	3031.93	250.30	302.50	88.34	5.75
IPE160	20.09	869.30	300	400	3984.26	312.00	376.18	123.90	4.61
IPE180	23.95	1317	300	400	5066.45	383.70	466.94	166.40	3.7
IPE200	28.48	1943	300	400	6278.50	461.70	557.51	220.60	3.12
IPE220	33.37	2772	300	400	7637.18	564.70	683.38	285.40	2.55
IPE240	39.12	3892	300	400	9147.16	637.20	804.01	366.60	2.26
IPE270	45.94	5790	300	400	11605.28	741.50	968.45	484.00	1.94
IPE300	53.81	8356	300	400	14286.71	813.80	1086.75	628.40	1.77
IPE330	62.61	11770	300	400	17297.01	854.20	1142.95	804.30	1.69
IPE360	72.73	16270	300	400	20567.36	902.90	1199.00	1019.00	1.59
IPE400	84.46	23130	300	400	25205.31	927.40	1230.83	1307.00	1.55
IPE450	98.82	33740	300	400	31426.79	951.90	1260.67	1702.00	1.51
IPE500	115.50	48200	300	400	38408.98	976.60	1289.92	2194.00	1.47
IPE550	134.40	67120	300	400	45967.53	1001.40	1309.13	2787.00	1.44
IPE600	156.00	3387	300	400	54338.62	1018.00	1327.79	3512.00	1.41

Şekil 4.4: pveri.txt Veri Dosyası

A) Genetik Algoritma Parametrelerinin Hazırlanması

Veri dosyalarının yanında YZDOPT1 bir programının kullanacağı GA parametrelerini içeren bilgilerde programın ilk başlarında girilecektir. Bunlar:

topboy = 16; % Topluluğun büyüklüğü

degoran = 0.01; % Değişim (mutasyon) oranı

caporan = 0.5; % Caprazlama oranı

du = 4; % Dizi uzunluğu (Her bir değişkendeki gen sayısı veya dizideki gen sayısı)

tds = 2; % Tasarım değişken sayısı (dizi sayısı)

ku = du * tds; % Kromozom uzunluğu (Bir kromozomdaki toplam gen sayısı)

taban=2; % Kromozomun oluşturulacağı tabandır. Burada 2'li taban cinsindedir.

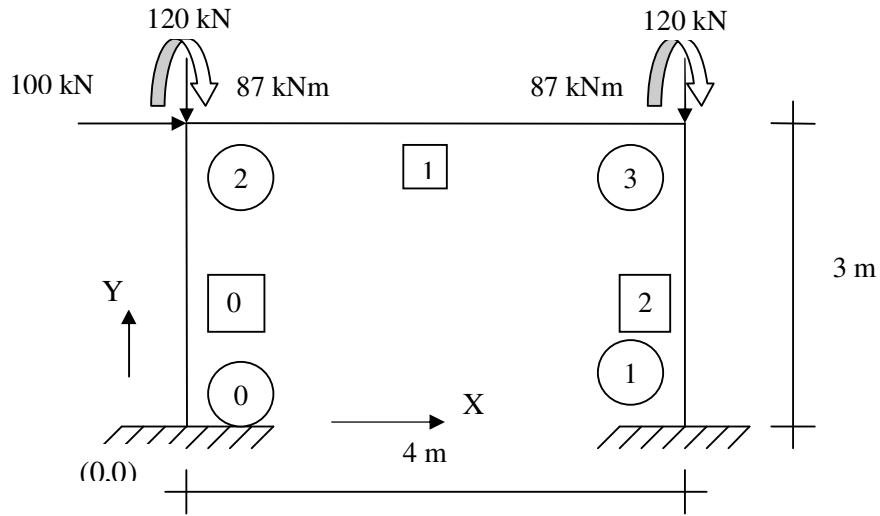
Program çalıştırıldığında, rastgele seçilen bireylerdeki tasarım değişkenleri değerleri daha önce hazırlanmış olan sistemde bulunan eleman kesit alanları ve diğer özellikler atanmakta ve sistem çözülerek ağırlık hesabı yapılmakta ve amaç fonksiyonu hesaplanmaktadır. Her nesil tamamlandığında genetik algoritma işlemleri uygulanmaktadır. Döngü işlemlerine son verilebilmesi için nesildeki bireylerin en az %80-85 oranında benzer olmaları gerekmektedir. Benzer olan bu bireyler optimum çözümü temsil etmektedir.

5. SAYISAL ÖRNEK

5.1 Örnek 1

Çalışmada örnek olarak önce üç elemanlı, basit bir çerçeve örneği ele alınarak, genetik algoritma yöntemi ile çözümü incelenmiştir.

Şekil 5.1’de örnek çerçevenin özellikleri aşağıda açıklanmıştır. Elemanların kesit alanları değişmektedir 1. ve 3. eleman A1 grubunu, diğer elemanda A2 grubunu oluşturur.



Şekil 5.1: Örnek 1 İçin Tek Açıklıklı Çelik Çerçeve

Adım 1: Yapının Malzeme ve Geometrik Özellikleri

Malzeme : St 37

Profil : IPE

Adım 2: Genetik Algoritma İle İlgili Kabuller

1. Genler (Genetik sayı sistemi) : 1,0

2. Dizi uzunluğu (bir dizideki toplam gen sayısı) (du) : 4

3. Tasarım değişkeni sayısı (tds) : 2

4. Popülasyon sayısı (bir nesildeki birey sayısı) (topboy): 16

Popülasyon sayısı topluluğun büyüklüğünü ifade eder.

Adım 3: Araştırma Uzayının Belirlenmesi

Problemdeki dizi uzunluğu (bir dizideki toplam gen sayısı) 4 seçilmiştir. İkili sayı sistemi “0” ve “1” ‘den oluştuğuna göre, ikili sayı sistemini oluşturan en büyük sayı “1” ‘dir. Dört hane uzunluğundaki bir dizinin (ikilik tabana göre dört basamaklı sayı) onluk tabandaki karşılığı en büyük sayıyı ifade eden değeri ise “(1111)₂” ‘dir. Dört basamaklı, en küçük değere karşılık gelen bu sayının onluk tabandaki karşılığı esas alınarak araştırma uzayı eleman sayısı belirlenir:

$$(1111)_2 \rightarrow (15)_{10}$$

(1111)₂ sayısı onluk tabandaki karşılığı (15)₁₀ ‘dur. Ancak (0000)₂ sayısı onluk tabana göre (0)₁₀ ‘a karşılık gelmesine rağmen o da bir sayıdır ve bir tasarım değişkenini ifade edebilir. Bu durumda problemdeki araştırma uzayının toplam eleman sayısı 15+1=16 olur. Diziler, M=0000 ve N=1111 arasında değişen 16 farklı değer almakta ve her biri araştırma uzayındaki bir uzunluk değerine karşılık gelmektedir. Yine araştırma uzayının ilk elemanı (0000)₂ olarak alınmakta ve değer araştırma uzayındaki ilk elemana karşılık gelmektedir. (0000)₂ değerinin onluk tabana göre karşılığı sıfır olduğuna göre hesap 1’den değil 0’dan başlayacağı için fonksiyon haritası $l_{w,j}=S_{i+1}$ olur. Buradaki i, j. Grubu gösteren diziyeye karşılık gelen onluk taban değeridir.

Araştırma uzayı diğer bir deyişle set 16 elemandan oluşturulur. Buradaki her bir değer ikilik tabandaki bir sayıya karşılık gelmektedir.

Adım 4: Bireylerin Oluşturulması

Bu örnekte, bir nesli oluşturan toplam birey sayısı 16 olarak kabul edilmiştir. Bireylerin uzunlukları, yani, bir bireyi oluşturan toplam gen sayısı ise şöyle belirlenir: Bir dizi, sadece bir tasarım değişkenini, yani, bir elemanın kesit alanı değerini ifade eder. Bu örnekte 2 adet tasarım değişkeni olduğuna göre iki tane tasarım değişkeninin, yani iki tane dizinin yan yana yazılması gerekir. Yan yana yazılan bu dizilerin toplamının oluşturulması olduğu kromozoma bir “birey” denir. Bu durumda bir bireyin uzunluğu (kromozom uzunluğu)

$$\text{Kromozom uzunluğu (ku)} = \text{tasarım değişkeni sayısı (tds)} * \text{dizi uzunluğu (du)}$$

$$\text{ku} = 4 * 2 = 8 \text{ (gen sayısı)}$$

G gen haritası; $G=H(A1,A2)$ ’dir. Buradan Genetik Algoritma kodlamasına geçilecek olursa;

$$G=H(1111,1111) \text{ elde edilir.}$$

Böylece bireyler, (0000,0000) ile (1111,1111) değerleri arasında değişmektedir. Bu durumda bir birey, toplam 8 adet gen’den, diğer bir deyişle 8 adet “1” ve “0” lardan oluşmaktadır.

Bireyin uzunluğu (kromozom uzunluğu) ve toplam birey sayısı belirlendikten sonra 1. nesildeki bireyler rastgele belirlenerek 16 adet bireyden oluşan 1. nesil oluşturulmuş olur.

Adım 5: Her Bir Birey İçin Nesil Tablosunun Hazırlanması ve Hesaplar

- 1) Amaç Fonksiyonu ve Kısıtlamaların Hesabı
- 2) Tablo İşlemleri

Adım 6: Topluluktaki tüm bireyler için Adım 4'ten Adım 6'nın tekrarlanması.
 Bu örnekte, bir düzlem çelik çerçevenin tasarım değişkenleri için varsayılan uygun
 16 adet kesit alanı değerleri ele alınırsa;

Tablo 5.1: Örnek 1 İçin Araştırma Uzayı

ARAŞTIRMA UZAYININ BELİRLENMESİ			
İkili Sayılar (Ak(j))	Onluk Taban Karşılığı	Set'teki Sıra Numarası (Se(i+1))	Eleman Enkesit Boyları (cm2)
0000	0	1	13,20
0001	1	2	16,40
0010	2	3	20,10
0011	3	4	23,90
0100	4	5	28,50
0101	5	6	33,40
0110	6	7	39,10
0111	7	8	45,90
1000	8	9	53,80
1001	9	10	62,60
1010	10	11	72,70
1011	11	12	84,50
1100	12	13	98,80
1101	13	14	116,00
1110	14	15	134,00
1111	15	16	156,00

Tablo 5.2: Araştırma Uzayını Oluşturan Profillerin Geometrik Özellikleri

Profil Adı	A (cm ²)	I _x (cm ⁴)	I _y (cm ⁴)	i _x (cm)	i _y (cm)	L (cm)
IPE120	13.21	317.8	27.67	4.9	1.45	300
IPE140	16.43	541.2	44.92	5.74	1.65	300
IPE160	20.09	869.3	68.31	6.58	1.84	300
IPE180	23.95	1317	100.9	7.42	2.05	300
IPE200	28.48	1943	142.4	8.26	2.24	300
IPE220	33.37	2772	204.9	9.11	2.48	300
IPE240	39.12	3892	283.6	9.97	2.69	300
IPE270	45.94	5790	419.9	11.23	3.02	300
IPE300	53.81	8356	603.8	12.46	3.35	300
IPE330	62.61	11770	788.1	13.71	3.55	300
IPE360	72.73	16270	1043	14.95	3.79	300
IPE400	84.46	23130	1318	16.55	3.95	300
IPE450	98.82	33740	1676	18.48	4.12	300
IPE500	115.5	48200	2142	20.43	4.31	300
IPE550	134.4	67120	2668	22.35	4.45	300
IPE600	156	92080	3387	24.3	4.6	300

Tablo 5.3: Araştırma Uzayını Oluşturan Profillerin Diğer Parametreleri

$\sigma_e(K=1)$ için)	$\lambda(K=1)$ için)	$\sigma_{bem}(K=1)$ için)	ω	σ_B	$wz(cm^3)$
2209	207	193.5	7.44	233.6	60.73
3032	182	250.3	5.75	302.5	88.34
3984	163	312	4.61	376.2	123.9
5066	146	383.7	3.7	466.9	166.4
6279	134	461.7	3.12	557.5	220.6
7637	121	564.7	2.55	683.4	285.4
9147	112	637.2	2.26	804.0	366.6
11605	99	741.5	1.94	968.4	484
14287	90	813.8	1.77	1086.7	628.4
17297	85	854.2	1.69	1142.9	804.3
20567	79	902.9	1.59	1199.0	1019
25205	76	927.4	1.55	1230.8	1307
31427	73	951.9	1.51	1260.7	1702
38409	70	976.6	1.47	1289.9	2194
45968	67	1001.4	1.44	1309.1	2787
54339	65	1018	1.41	1327.8	3512

$S_e = \{ 13.20, 16.40, 20.10, 23.90, 28.50, 33.40, 39.10, 45.90, 53.80, 62.60, 72.70, 84.50, 98.50, 116, 134, 156 \} (cm^2)$

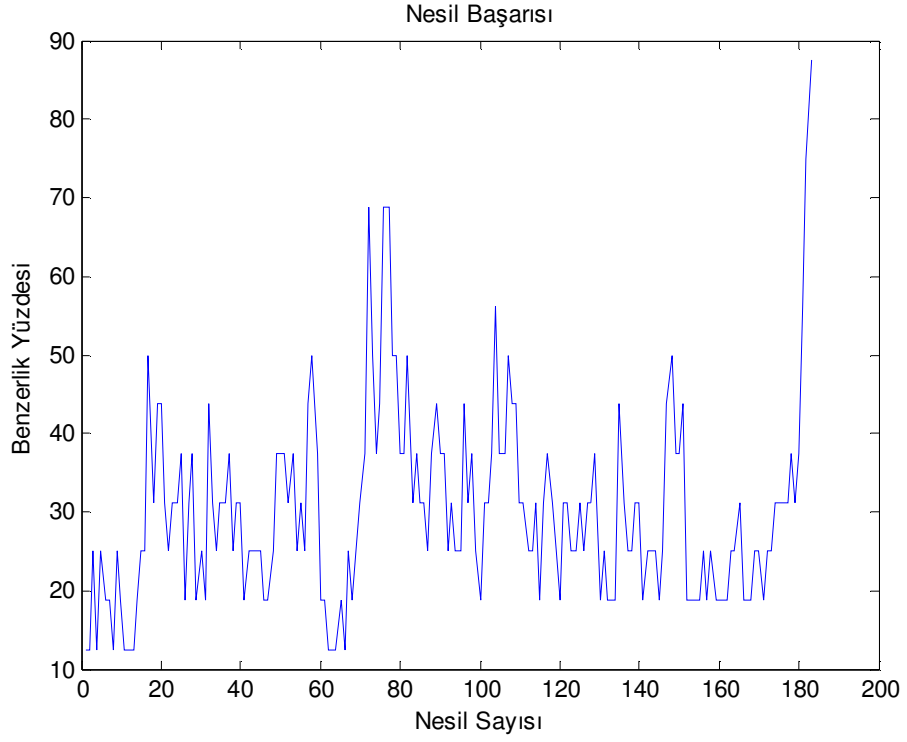
Geliştirilen algoritma verimli bir şekilde çalışmakta olup 242. nesilde optimum kesit alanları $A_1=33.37 cm^2$ ve $A_2=28.48 cm^2$ olarak bulunmuştur.

Bu çalışmada Genetik Algoritma, basit bir çelik çerçevenin ağırlık optimizasyonunda kullanılmıştır. Kullanılan metodun çözüm hızı geleneksel yöntemlere kıyasla oldukça yüksektir. Sunulan çalışma, Genetik Algoritma'nın temel adımlarını ve felsefesini anlatmaktadır.

Tablo 5.4: Örnek 1 için Son Uygun Nesil

ÖRNEK 1 İÇİN 183.NESİL																							
Birey No	Topluluk								W(x)	C	Fi	Fi/For	Eşleme Havuzu						Eşleşme	Kesim Yerleri	Yeni Topluluk		
1	0	1	0	1	0	1	0	0	973.80	0	2501.5	1.131	0	1	0	1	0	1	0	0	15	2	1010100
2	0	1	0	1	0	1	0	0	973.80	0	2501.5	1.131	0	1	0	1	0	1	0	0	4	2	1010100
3	0	1	0	1	0	1	0	0	973.80	0	2501.5	1.131	0	1	0	1	0	1	0	0	7	3	1010100
4	0	1	0	1	0	1	0	0	973.80	0	2501.5	1.131	0	1	0	1	0	1	0	0	2	3	1010100
5	0	1	0	1	0	1	0	0	973.80	0	2501.5	1.131	0	1	0	1	0	1	0	0	16	2	1010100
6	0	1	0	1	0	1	0	0	973.80	0	2501.5	1.131	0	1	0	1	0	1	0	0	3	2	1010100
7	0	1	0	1	0	1	0	0	1034.50	0	2440.8	1.104	0	1	0	1	0	1	0	0	9	2	1010100
8	0	1	0	1	0	1	0	0	2501.50	0	973.8	0.440	0	1	0	1	0	1	0	0	7	2	1010100
9	0	1	0	1	0	1	0	0	973.80	0	2501.5	1.131	0	1	0	1	0	1	0	0	11	6	1010100
10	0	1	0	1	0	1	0	1	2501.50	0	973.8	0.440	0	1	0	1	0	1	0	1	9	6	1010101
11	0	1	0	1	0	1	0	0	973.80	0	2501.5	1.131	0	1	0	1	0	1	0	0	3	3	1010100
12	0	1	0	1	0	1	0	0	2501.50	0	973.8	0.440	0	1	0	1	0	1	0	0	8	3	1010100
13	0	1	0	1	0	1	0	0	973.80	0	2501.5	1.131	0	1	0	1	0	1	0	0	9	5	1010100
14	0	1	0	1	0	1	0	0	973.80	0	2501.5	1.131	0	1	0	1	0	1	0	0	7	5	1010100
15	0	1	1	1	0	1	0	0	973.80	0	2501.5	1.131	0	1	1	1	0	1	0	0	10	1	1110100
16	0	1	0	1	0	1	0	0	973.80	0	2501.5	1.131	0	1	0	1	0	1	0	0	2	1	1010100
Toplam hesaplama zamanı:47.9531 sn																							
Benzerlik yuzdesi = % 87.5																							
Mutasyon sayısı = 2																							
A1=33.37 (IPE220), A2=28.48 (IPE200)																							

Bu basit GA'da çaprazlama olarak tek noktadan kesim yöntemi ve mutasyon operatöründe de tek gen'in değişimi yapılmıştır. Eşleşme havuzunun seçiminde zayıf bireylerin elenmesinden sonra bunların yerine en güçlü birey kopyalanmıştır.



Şekil 5.2 : Nesil Başarısı

6. SONUÇLAR VE ÖNERİLER

Bu çalışmada, öncelikle tek açıklıklı basit çerçevelerin statik yükler altında tasarımı ve genetik algoritma yöntemi yardımıyla ağırlık optimizasyonu incelenmiş ve sayısal örneklerde kullanılmak üzere bilgisayar programları geliştirilmiştir. Bu konuda daha ileri aşamada araştırma yapmak isteyenler için öneriler aşağıda sıralanmıştır.

Genetik Algoritma, matematiksel olarak çözümü mümkün olmayan veya çok karmaşık olan problemleri çözebilecek kapasitededir.

Genetik Algoritma'ya dayanan optimum tasarım algoritması ayrık değişkeni kullanılan problemlerin çözümü için oldukça uygundur.

Genetik Algoritmanın kullanılabilmesi için problemin yapısı hakkında bilgi sahibi olunması gerekir. Problemin çözüm kümesi yani araştırma uzayı tasarımcı tarafından belirlenmektedir. Optimum çözüme ulaşılabilecek tasarım değişkenlerinin araştırma uzayının dışında kalması durumunda mutlak optimum yerine bölgesel optimum bulunmasına neden olur.

Çözüme ulaşma hızı, verilen tasarım değişkeni sayısı, araştırma uzayının eleman sayısı ve sistemin büyüklüğü ile ters orantılıdır.

GA' rastgele yöntemleri kullanmasından dolayı her çözümde matematiksel yöntemler gibi aynı sonuçları veremeyebilir. Ama verdiği sonuçlar optimum sonuç olarak kabul edilebilir seviyededir.

İnşaat mühendisliği problemlerinde GA'yı daha etkin kullanabilmek için:

- a) Çözüm kümesi ile topluluğun büyüklüğü arasında bağıntının,
 - b) GA parametreleri arasında performans analizinin,
 - c) Topluluğu oluştururken seçilen rastgele üretme motorları arasındaki performans durumunun,
 - d) Uygunluk kriterinin hassasiyetinin,
- üzerinde dikkatlerin bu noktalara çekilmesi gerekir.

KAYNAKLAR

- Altay, A.**, 2008. Kişisel görüşme.
- Arifoğlu, U.**,2005. MATLAB 7.04 Simulink ve Mühendislik Uygulamaları, Alfa Basım-Yayın, İstanbul.
- Atabay, Ş. ve Gülay, F.G.**, 2004. Genetik algoritmalar ile perdeli yapı sisteminin maliyet optimizasyonu, *itüdergisi/d mühendislik*, **3**, 6, 71-81.
- Atabay, Ş.**, 2008. Kişisel görüşme.
- Atabay, Ş.**,2004. Perdeli betonarme yapı sistemlerinin genetik algoritma ile algoritma ile optimum tasarımı, *Doktora Tezi*, İ.T.Ü. Fen Bilimleri Enstitüsü, İstanbul.
- Bekiroğlu, S., Ayvaz, Y. ve Dede, T.**, 2003. Değer kodlaması kullanılarak kafes sistemlerin genetik algoritma ile minimum ağırlıklı boyutlandırılması, *III. Uluslar arası İleri Teknolojiler Sempozyumu*, Gazi Üniversitesi, Ankara.
- Çağdaş, İ.U.**, 2001. Betonarme döşemelerin akma çizgileri yöntemiyle analizinde akma çizgisi deseninin genetik algoritma ile belirlenmesi, *Doktora Tezi*, Gazi Üniversitesi, Fen Bilimleri Enstitüsü, Ankara.
- Çağlayan, B.Ö.**, 2008. Kişisel görüşme.
- Daloğlu, A., Armutçu, M.**, 1998. Genetik algoritma ile düzlem çelik çerçevelerin optimum tasarımı, *Teknik Dergi*, Cilt **9**, Sayı 2.
- Değertekin, S.Ö., Ülker, M. ve Hayalioğlu, M.S.**, 2006. Uzay çelik çerçevelerin tabu ve genetik algoritma yöntemleriyle optimum tasarımı, *İMO Teknik Dergi*, Yazı 259, 3917-3934.
- Deren, H., Uzgider E. ve Piroğlu F.**, 2002 Çelik Yapılar, Çağlayan Kitapevi, İstanbul.
- Dominguez, A., Stiharu, I. and Sedaghati, R.**, 2005. Practical desing optimization of truss structures using the genetic algorithms, *Res Eng Design*, 10 August 2006 (online).
- Fairbairn, E.M.R., Silvos, M.M., Filho, R.D.T., Alves, J.L.D. and Ebecken, N.F.F.**, 2004. Optimization of mass concrete construction using genetic algorithms, *Computers and Structures*, **82**, 281-299.
- Goldberg, D.E.**, 1989. Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley Press, Canada.

- Govindaraj, V., and Ramasamy, J.V.,** 2005. Optimum detailed design of reinforced concrete continuous beams using Genetic Algorithms, *Computers and Structures* **84**, 34-48.
- Güneş, A. ve Yıldız, K.,**1997. MATLAB for Windows, Türkmen Kitabevi, İstanbul.
- Haupt, R. and Haupt, S.E.,** 1998. Practical Genetic Algorithms, Wiley-Interscience Publication, ABD.
- Hernandez, S., Sayed, M.E. and Brebbia, C.A.,** 1995. Structural Optimization, Computer Aided Optimum Design of Structures IV, Computational Mechanics Publications, UK
- İnan, A.,** 2007. MATLAB Klavuzu, Papatya Yayıncılık, İstanbul.
- Karaboğa, D.,** 2004. Yapay Zeka Optimizasyon Algoritmaları, Atlas Yayın Dağıtım, İstanbul.
- Karaduman, M.,** 2005. Çelik Yapılar, Atlas Yayın Dağıtım, İstanbul.
- Kaveh, A. and Rahami, H.,** 2005. Analysis, design and optimization of structures using force method and genetic algorithm, *International Journal for Numerical Methods in Engineering*, **65**, 1570-1584.
- Kaya, M.,** 2001. Betonarme yüksek kiriş tasarımında genetik algoritmaların kullanılması, *Yüksek Lisans Tezi*, Gazi Üniversitesi, Fen Bilimleri Enstitüsü, Ankara.
- Kömür, M. ve Altan, M.,** 2006. Betonarme bir kiriş ve kolonun genetik algoritma ile optimum boyutlandırılması, *GAP V. Mühendislik Kongresi*, Harran Üniversitesi, Şanlıurfa.
- Lagaros, N.D., Papadrakakis, M., and Kokossalakis, G.,** 2002. Structural optimization using evolutionary algorithms, *Computers and Structures*, **80**, 571-589.
- Mathias, J.D., Balandraud, X. and Grediac, M.,** 2006. Applying a genetic algorithm to the optimization of composite patches, *Computers & Structures*, **84**, 12, 823-834.
- McMahon, M.T., Watson, L.T., Soremekun, G.A., Gtirdal, Z. and Haftka, R.T.,**1998. A fortran 90 genetic algorithm module for composite laminate structure design, *Engineering with Computers*, **14**, 260-273.
- Michalewicz, Z.,** 1996. Genetik Algoritmalar + Data Structures = Evolution Programs', Third Edition Springer-Verlag Berlin Heidelberg, Newyork.
- Nabiyev, V.V.,** 2005. Yapay Zeka, Seçkin Yayıncılık, Ankara.
- Pern, Y. H., Juan, Y. and Hsu, H. S.,** 2005. Genetic algorithm-based decision support for the restoration budget allocation of historical buildings, *Building and Environment*, 11 September 2005, (In pres)

- Polat, A.**, 2006. Kafes sistemlerin genetik algoritma ile çok amaçlı optimizasyonu, *Yüksek Lisans Tezi*, Fırat Üniversitesi, Fen Bilimleri Enstitüsü, Ankara.
- Raich, A. and Liszkai, T.**, 2003. Benefits of implicit redundant genetic algorithms for structural damage detection in noisy environments, *GECCO*, LNCS **2724**, 2418-2419.
- Raich, A.M. and Ghaboussi, J.**,2000. Evolving structural design solutions using an implicit redundant genetic algorithm, *Struct Multidisc Optim*, **20**, 222-231.
- Rajeev, S. and Krishnamoorthy**, 1992. Discrete optimization of structures using genetic algorithms, *J.of Structural Engineering,ASCE*, **118**, 5, 1233-1250.
- Sahab, M.G., Ashour, A.F. and Toropov, V.V.**, 2005. Cost optimisation of reinforced concrete flat slab buildings, *Engineering Structures* **27**, 313-322.
- Sahab, M.G., Ashour, A.F. and Toropov, V.V.**, 2005.. A hybrid genetic algorithm for reinforced concrete flat slab buildings, *Computers and Structures* **83**, 551–559.
- Şen, Z.**, 2004. Genetik Algoritma ve En İyileme Yöntemleri, Su Vakfı Yayınları, İstanbul.
- Toğan, V. ve Daloğlu, A.**, 2003. Genetik algoritma ile paralel başlıklı düzlem kafes sistemlerin şekil ve boyut optimizasyonu, *Deprem Sempozyumu*, Kocaeli, s. 508.
- Toğan, V., ve Daloğlu, A.**, 2003. Genetik algoritma ile çatı makasların şekil ve boyut optimizasyonu, *III. Uluslar arası İleri Teknolojiler Sempozyumu*, Gazi Üniversitesi, Ankara.
- Torregosa, R.F. and Kanok-Nukulchai, W.**, 2002. Weight optimization of steel frames using genetic algorithm, *Advances in Structural Engineering*, **5**, 2, 99-111(13).
- TS-648**, 1980. Çelik yapıların hesap ve yapım kuralları, *Türk Standartları Enstitüsü*, Ankara.
- Turgut, P. ve Arslan, A.**, 2001. Sürekli bir kirişte maksimum momentlerin genetik algoritmalar ile belirlenmesi, *DEÜ Mühendislik Fakültesi Fen ve Mühendislik Dergisi*, **3**, 3, 1-9.
- Turgut, P., Gürel, M.A. ve Arslan, A.**, 1997. Genetik algoritma ile betonarme bir kirişin optimum tasarımı, *Prof. Dr. A. Rifat Yarar Sempozyumu*, İ.T.Ü., İstanbul.

- Uzunođlu, M., Kızıl, A. ve Onar, Ö.Ç.,** 2003. Her Yönü ile MATLAB, Türkmen Kitabevi, İstanbul.
- Wang, W., Zmeureanu, R. and Rivard, H.,** 2005. Applying multi-objective genetic algorithms in green building design optimization, *Building and Environment* **40**,1512-1525.
- Yılmaz, M.,** 2008. Kişisel görüşme.
- Yurtcu, Ş., İçağa, Y.,**2006. Evrimsel algoritmaların inşaat mühendisliği sistemlerinde kullanımı, *Yapı Teknolojileri Elektronik Dergisi*, **2006(1)**, 51-59.
- Zolfaghari, A.R., Heath, A.C. and Mccombie, P.F.,** 2005. Simple genetic algorithm search for critical non-circular failure surface in slope stability analysis, *Computers and Geotechnics* **32**, 139-152.

ÖZGEÇMİŞ

Tacettin ŞEKER, 1981 yılında Gaziantep'in Şehitkamil İlçesine bağlı Arıl Kasabası'nda doğmuştur. İlk, orta ve lise eğitimini aynı kasabada yapmıştır. 2000 yılında Mustafa Kemal Üniversitesi, Mühendislik-Mimarlık Fakültesi, İnşaat Mühendisliği Bölümü'nde lisans eğitimine başlamış ve 2004 yılında bölüm birincisi ve fakülte üçüncüsü olarak mezun olmuştur. 2006 yılında İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, İnşaat Mühendisliği Anabilim Dalı, Deprem Mühendisliği Programı'nda master öğrenimine başlamıştır.