



**ÇELİK YAPILARIN KAYNAKLI
BİRLEŞİMLERİNİN METASEZGİSEL
YÖNTEMLERLE OPTİMUM TASARIMI**

YÜKSEK LİSANS TEZİ

Soner SEZER

Danışman

Dr. Öğr. Üyesi Murat HİÇYILMAZ

İNŞAAT MÜHENDİSLİĞİ ANABİLİM DALI

Şubat 2021

AFYON KOCATEPE ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YÜKSEK LİSANS TEZİ

ÇELİK YAPILARIN KAYNAKLI BİRLEŞİMLERİNİN
METASEZGİSEL YÖNTEMLERLE OPTİMUM TASARIMI

Soner SEZER

Danışman
Dr. Öğr. Üyesi Murat HİÇYILMAZ

İNŞAAT MÜHENDİSLİĞİ ANABİLİM DALI

Şubat 2021

ÖZET
Yüksek Lisans Tezi

**ÇELİK YAPILARIN KAYNAKLI BİRLEŞİMLERİNİN METASEZGİSEL
YÖNTEMLERLE OPTİMUM TASARIMI**

Soner SEZER

Afyon Kocatepe Üniversitesi

Fen Bilimleri Enstitüsü

İnşaat Mühendisliği Anabilim Dalı

Danışman: Dr. Öğr. Üyesi Murat HİÇYILMAZ

Bu tez çalışmasında, çelik yapıların kaynaklı birleşimleri metasezgisel yapay zekâ algoritmaları kullanılarak optimize edilmiştir. Çalışmada kullanılmak üzere seçilen köşe kaynaklı birleşimden, yükleme koşullarına göre üç farklı sayısal model oluşturulmuştur. Bu modeller Çelik Yapıların Tasarım, Hesap ve Yapım Esaslarına Dair Yönetmelik – 2018 (ÇYTHYE-2018) uyarınca Yük ve Dayanım Katsayıları ile Tasarım (YDKT) ve Güvenlik Katsayıları ile Tasarım (GKT) prensipleri için ayrı ayrı kodlanmıştır. Kodlanan modeller MATLAB dilinde yazılmış Jaya, TLBO ve Sinüs-Kosinüs algoritmaları ile üç farklı popülasyon büyüklüğü kullanılarak optimize edilmiştir. Optimizasyonlar için amaç fonksiyonu olarak toplam kaynak alanının minimizasyonu seçilmiştir. Yükleme durumu, tasarım prensibi, popülasyon büyüklüğü ve kullanılan çözüm algoritması gibi farklı senaryolar altında yapılan 54 adet analizin sonuçları ilgili parametreler için karşılaştırılmıştır. Ayrıca bu çalışmada, kısıtlı optimizasyon yönteminde kullanılmak üzere yeni bir cezalandırma sistemi olan “Adil Ceza Yöntemi” geliştirilmiştir. Sonuç olarak, kullanılan algoritmaların çelik yapıların kaynaklı birleşimlerin optimizasyonu için uygun olduğu anlaşılmış olup algoritmaların birbirlerine göre avantaj ve dezavantajları da ortaya konulmuştur. Ayrıca, bu çalışmada sunulan adil ceza yönteminin kısıtlı optimizasyon problemlerinin çözümünde kullanılabilir olduğu görülmüştür.

2021, xi + 95 sayfa

Anahtar Kelimeler: Çelik Yapılar, Kaynaklı Birleşimler, Optimizasyon, Metasezgisel

ABSTRACT
M.Sc. Thesis

**OPTIMUM DESIGN OF WELDED CONNECTIONS OF STEEL STRUCTURES
USING METAHEURISTIC METHODS**

Soner SEZER

Afyon Kocatepe University

Graduate School of Natural and Applied Sciences

Department of Civil Engineering

Supervisor: Asst. Prof. Murat HİÇYILMAZ

In this thesis, welded connections of steel structures were optimized by using metaheuristic artificial intelligence algorithms. Three different numerical models were created from the fillet welded connection chosen to be used in the study, according to the loading conditions. These models were coded separately for Load and Resistance Factor Design (LRFD) and Allowable Strength Design (ASD) principles in ÇYTHYE-2018. The coded numerical models were optimized for three different population sizes with Jaya, TLBO and Sine-Cosine algorithms written in MATLAB language. The minimization of the total welding area were chosen as the objective function in optimizations. The results of 54 analyzes obtained for different scenarios such as loading status, design principle, population size and solution algorithm were compared for the relevant parameters. Furthermore, the “Fair Penalty Method”, which is a unique penalty method, was developed to be used in the constrained optimization method. As a result, it is understood that all of the algorithms used in the study were suitable for the optimization of welded connections of steel structures. The advantages and disadvantages of algorithms were understood by comparing them with each other. The fair penalty method proposed in this study was found to be usable for constrained optimization problems.

2021, xi + 95 pages

Keywords: Steel Structures, Welded Connections, Optimization, Metaheuristic

TEŞEKKÜR

Bu araştırmanın konusu, çalışmaların yönlendirilmesi, sonuçların değerlendirilmesi ve yazımı aşamasında yapmış olduğu büyük katkılarından dolayı tez danışmanım Sayın Dr. Öğr. Üyesi Murat HİÇYILMAZ'a, değerli fikirleri ile yol gösteren Sayın Prof. Dr. Mehmet Polat SAKA'ya, her konuda öneri ve eleştirileriyle yardımlarını gördüğüm hocalarıma ve arkadaşlarıma teşekkür ederim.

Eğitim hayatım boyunca maddi ve manevi destekleriyle her zaman yanımda olan aileme şükranlarımı sunarım.

Soner SEZER
AFYONKARAHİSAR, 2021

İÇİNDEKİLER DİZİNİ

	Sayfa
ÖZET	i
ABSTRACT	ii
TEŞEKKÜR	iii
İÇİNDEKİLER DİZİNİ.....	iv
KISALTMALAR DİZİNİ	vi
ŞEKİLLER DİZİNİ	vii
ÇİZELGELER DİZİNİ.....	ix
RESİMLER DİZİNİ	xi
1. GİRİŞ.....	1
1.1 Tezin Kapsamı	1
1.2 Tezin Amacı.....	2
1.3 Tezin Özgün Değeri.....	3
2. LİTERATÜR BİLGİLERİ	4
2.1 Kaynaklı Birleşimler.....	4
2.1.1 Kaynaklı Birleşimlerin Tarihçesi	4
2.1.2 Teknolojilerine Göre Kaynak Yöntemleri	7
2.1.2.1 Gaz Eritme Kaynağı	8
2.1.2.2 Elektrik Ark Kaynağı	9
2.1.3 Uygulanma Biçimlerine Göre Kaynak Çeşitleri	14
2.1.3.1 Küt Kaynak Dikişleri.....	14
2.1.3.2 Köşe Kaynak Dikişleri	16
2.1.3.3 Dairesel ve Oval Dolgu Kaynaklar	17
2.2 Metasezgisel Optimizasyon Yöntemi	18
2.2.1 Amaç Fonksiyonu	20
2.2.2 Tasarım Değişkenleri	21
2.2.3 Kısıtlar.....	21
2.2.4 Ceza Fonksiyonu.....	22
2.3 Yapısal Optimizasyon ve Önceki Çalışmalar	22
3. MATERYAL ve METOT	28
3.1 Çalışmada Kullanılan Metasezgisel Algoritmalar	28

3.1.1 Jaya Algoritması.....	28
3.1.2 Teaching and Learning Based Optimization (TLBO).....	32
3.1.3 Sinüs-Kosinüs Algoritması (SCA).....	39
3.2 ÇYTHYE 2018'e Göre Köşe Kaynaklı Birleşimlerin Tasarımı	43
3.2.1 Etkin Alan	43
3.2.2 Sınırlamalar	44
3.2.3 Kaynaklı Birleşimlerin Dayanımı	49
3.3 Problemlerin Tanıtımı ve Optimizasyon Modellerinin Oluşturulması	51
3.3.1 Giriş.....	51
3.3.2 Sayısal Modeller.....	52
3.3.3 Tasarım Değişkenleri	52
3.3.4 Popülasyonun Yapısı.....	53
3.3.5 Amaç Fonksiyonu	54
3.3.6 Kısıtlar.....	54
3.3.6.1 Kaynak Kalınlığı Kısıt Kontrolü	56
3.3.6.2 Etkin Kaynak Uzunluğu Kısıt Kontrolü.....	57
3.3.6.3 Başlık Dış Kaynak Grubunun Dış Ucuna Gelen Gerilmelerin Kısıt Kontrolleri (g_3 ve g_7)	58
3.3.6.4 Gövde Kaynak Grubunun Üst Ucuna Gelen Gerilmelerin Kısıt Kontrolleri (g_4 ve g_8)	59
3.3.6.5 Gövde Kaynak Grubunun Kesme Kısıt Kontrolleri (g_5 ve g_9)	60
3.3.6.6 Gövdeye Gelen Bileşke Gerilmelerin Kısıt Kontrolleri (g_6 ve g_{10}).....	61
3.3.7 Adil Ceza Yöntemi ve Ceza Fonksiyonu	62
3.3.8 Optimizasyon Algoritmaları İçin Belirlenen Parametreler	64
3.3.9 Algoritma Performanslarının Karşılaştırılması	67
4. BULGULAR	68
4.1 Sayısal Model – 1	68
4.2 Sayısal Model – 2	74
4.3 Sayısal Model – 3	80
5. TARTIŞMA ve SONUÇ	86
6. KAYNAKLAR.....	88
ÖZGEÇMİŞ.....	95

KISALTMALAR DİZİNİ

Kısaltmalar

AISC	American Institute of Steel Construction
ASD	Allowable stress design
ÇYTHYE	Çelik Yapıların Tasarım, Hesap ve Yapım Esaslarına Dair Yönetmelik
DOP	Doğrusal olmayan programlama
DP	Doğrusal programlama
GKT	Güvenlik katsayıları ile tasarım
LRFD	Load and resistance factor design
MAG	Metal aktif gaz
MIG	Metal inert gaz
SCA	Sin-Cosine Algorithm
SM	Sayısal model
TIG	Tungsten inert gaz
TLBO	Teaching and Learning Based Optimization
TP	Tamsayı programlama
YDKT	Yük dayanım katsayıları ile tasarım

ŞEKİLLER DİZİNİ

	Sayfa
Şekil 2.1 Oksi-asetilen kaynağı için donanım şeması (Tama 2009).	9
Şekil 2.2 Elektrot kaynağı için donanım şeması (Tama 2009).....	10
Şekil 2.3 Tozaltı kaynak için donanım şeması ve kaynak makinesi (Tama 2009).	11
Şekil 2.4 TIG kaynak için donanım şeması (Tama 2009).....	12
Şekil 2.5 MIG kaynak için donanım şeması (Ertürk 1987).....	13
Şekil 2.6 Küt kaynak.	15
Şekil 2.7 Köşe dikiş çeşitleri ve yükseklikleri (Gaylord vd. 1992).....	16
Şekil 2.8 Kuvvet yönüne göre köşe kaynaklar (Deren vd. 2008).....	16
Şekil 2.9 Dairesel ve dolgu oval kaynaklar.	18
Şekil 2.10 Kaynak dikişleri (Yorgun vd. 2017).	18
Şekil 3.1 Jaya algoritması için akış diyagramı.	30
Şekil 3.2 TLBO algoritması için akış şeması.	36
Şekil 3.3 SCA için arama modeli (Mirjalili 2015).	39
Şekil 3.4 r_1 'in çözüm üretmedeki etkisi (Mirjalili 2015).	40
Şekil 3.5 SCA için akış diyagramı.	41
Şekil 3.6 Tipik köşe kaynak uygulamaları (Yorgun vd. 2017).	43
Şekil 3.7 Köşe kaynakların etkin kalınlıkları (Gaylord vd. 1992).	44
Şekil 3.8 Maksimum kaynak kalınlıkları a. $t < 6\text{mm}$ b. $t > 6\text{mm}$ (Yorgun vd. 2017).	45
Şekil 3.9 Yalnızca boyuna kaynaklar ile yapılmış uç birleşim (ÇYTHYE 2018).	45
Şekil 3.10 Belirli aralıklarla düzenlenen süreksiz köşe kaynakların uygulama sınırları (ÇYTHYE 2018).	46
Şekil 3.11 Minimum bindirme uzunluğu (ÇYTHYE 2018).	47
Şekil 3.12 Çekme etkisindeki bir elemana köşe kaynaklı bağlantı (ÇYTHYE 2018). ..	47
Şekil 3.13 Kiriş ucu ile birleşim elemanı arasında dönme uyumunun sağlanabilmesi için kaynak uygulaması (ÇYTHYE 2018).	48
Şekil 3.14 Bir düzlemin farklı yüzeylerindeki köşe kaynaklar için uygulama detayı (ÇYTHYE 2018).	48
Şekil 3.15 Birleşim modelinin temel görünümü.	51
Şekil 3.16 Tasarım değişkenleri.	53
Şekil 3.17 Sinüs ve kosinüs aralığı için azalan model ($a=3$) (Mirjalili, 2015).	65
Şekil 3.18 Geniş aralıkta seçilen a parametrelerinin sonuçları.	67
Şekil 4.1 SM-1A için 10 çalıştırmanın ortalama değerleri. a. $n=250$, b. $n=500$, c. $n=1000$	71

Şekil 4.2 SM-1B için 10 çalıştırmanın ortalama değerleri. a. $n=250$, b. $n=500$, c. $n=1000$	73
Şekil 4.3 SM-2A için 10 çalıştırmanın ortalama değerleri. a. $n=250$, b. $n=500$, c. $n=1000$	76
Şekil 4.4 SM-2B için 10 çalıştırmanın ortalama değerleri. a. $n=250$, b. $n=500$, c. $n=1000$	79
Şekil 4.5 SM-3A için 10 çalıştırmanın ortalama değerleri. a. $n=250$, b. $n=500$, c. $n=1000$	82
Şekil 4.6 SM-3B için 10 çalıştırmanın ortalama değerleri. a. $n=250$, b. $n=500$, c. $n=1000$	85



ÇİZELGELER DİZİNİ

	Sayfa
Çizelge 2.1 Bazı küt kaynak dikişlerinin isimleri (Deren vd. 2008).....	15
Çizelge 3.1 Jaya algoritması için ilk üç adımının sonuçları.....	31
Çizelge 3.2 Denklem (3.2)'de kullanılacak r_1 ve r_2 rastsal sayıları.....	31
Çizelge 3.3 X' popülasyonu ve f' değerleri.	31
Çizelge 3.4 Güncellenmiş X popülasyonu ve f değerleri.	31
Çizelge 3.5 TLBO algoritması için ilk dört adım.....	37
Çizelge 3.6 T_F ve <i>Difference_Mean</i> değerleri.	37
Çizelge 3.7 Denklem (3.3) ile elde edilen geçici popülasyon.	37
Çizelge 3.8 Öğretmen aşamasının çıktısı olarak elde edilmiş nihai popülasyon.	37
Çizelge 3.9 Öğrenci aşaması için güncellenmiş geçici popülasyon.	38
Çizelge 3.10 Birinci iterasyonun sonucunda elde edilen değerler.	38
Çizelge 3.11 SCA'nın ilk iki adımının sonuçları.	42
Çizelge 3.12 Denklem (3.10) ile elde edilen geçici popülasyon.	42
Çizelge 3.13 SCA için birinci iterasyonun sonuçları.	43
Çizelge 3.14 Köşe kaynakların minimum kalınlıkları (ÇYTHYE 2018).....	44
Çizelge 3.15 Köşe kaynakların ϕ , Ω , F_{nBM} ve F_{nw} değerleri ve ÇYTHYE yönetmeliği içerisinde esas alınacak bölümler (ÇYTHYE 2018).....	49
Çizelge 3.16 Tasarım esasları ve yükleme durumlarına göre sayısal modeller.	52
Çizelge 3.17 Yükleme durumları için kuvvet dağılımları.	52
Çizelge 3.18 Tasarım değişkenleri.	53
Çizelge 3.19 GKT tasarım prensibi için kısıtlar.	55
Çizelge 3.20 YDKT tasarım prensibi için kısıtlar.	56
Çizelge 3.21 YDKT için rastlantısal olarak oluşturulmuş örnek kısıt hata sayaçları....	63
Çizelge 3.22 Algoritmalar için ortak çalıştırma parametreleri.	65
Çizelge 3.23 a parametresi için 10 çalıştırmanın sonuçları.	66
Çizelge 3.24 SCA için özel çalışma parametreleri.	67
Çizelge 4.1 SM-1A için en iyi sonuç değerleri.	68
Çizelge 4.2 SM-1A için algoritma karşılaştırma parametreleri.	69
Çizelge 4.3 SM-1B için en iyi sonuç değerleri.	71
Çizelge 4.4 SM-1B için algoritma karşılaştırma parametreleri.....	72
Çizelge 4.5 SM-2A için en iyi sonuç değerleri.	74
Çizelge 4.6 SM-2A için algoritma karşılaştırma parametreleri.	75

Çizelge 4.7 SM-2B için en iyi sonuç değerleri.	77
Çizelge 4.8 SM-2B için algoritma karşılaştırma parametreleri.....	78
Çizelge 4.9 SM-3A için en iyi sonuç değerleri.	80
Çizelge 4.10 SM-3A için algoritma karşılaştırma parametreleri.	81
Çizelge 4.11 SM-3B için en iyi sonuç değerleri.	83
Çizelge 4.12 SM-3B için algoritma karşılaştırma parametreleri.....	84



RESİMLER DİZİNİ

	Sayfa
Resim 2.1 Hindistan’da yapılmış kaynaklı demir sütun (İnt. Kyn. 1).	5
Resim 2.2 İlk Kaynaklı Yol Köprüsü: Maurzyce Köprüsü (İnt. Kyn. 1).....	6
Resim 2.3 Oksi-asetilen kaynağı ile metal kesimi işlemi (Tama 2009).....	8
Resim 2.4 Elektrot kaynağı imalatı (Tama 2009).	10
Resim 2.5 Kaynak bölgesine dökülen toz malzeme (Tama 2009).....	11
Resim 2.6 TIG kaynak uygulaması (Tama 2009).	13
Resim 2.7 MIG kaynak uygulaması (Tama 2009).....	14
Resim 2.8 Küt kaynağın giriş ekine uygulanması.....	15
Resim 2.9 Küt kaynağın kolon ekine uygulanması.....	16
Resim 2.10 Köşe kaynak uygulaması.	17
Resim 2.11 Açılı köşe kaynak uygulaması.	17

1. GİRİŞ

Artan dünya nüfusu ve azalan kaynakların bir sonucu olarak optimizasyon kavramı günümüzde popüler bir çalışma konusu haline gelmiştir. Optimizasyon, en genel anlamıyla çözüm uzayındaki en uygun çözümü bulma işine verilen isimdir. En uygun çözümü bulma işi; belirli sınırlamalar altında, amaç fonksiyonunu minimize veya maksimize etmek için, tasarım değişkenlerinin değerlerini değiştirmek suretiyle yapılmaktadır. Optimizasyonun amacı ise mühendislik yapılarında bir dizi tasarım kısıtlamaları karşılanırken, maliyet minimizasyonu gibi önceden belirlenmiş hedefler doğrultusunda en uygun tasarımın bulunmasını sağlamaktır. Optimizasyon, yüzyıllar boyunca çeşitli yöntemlerle yapılagelse de günümüzde bilgisayar teknolojileri ve yapay zekâ algoritmalarındaki gelişime paralel olarak daha hızlı ve daha verimli çözümler sunmaktadır. Bu sayede, kaynakların daha efektif kullanımı noktasında başarılı çözümler ortaya koyarak her geçen gün daha da popüler bir hale gelmektedir.

Pek çok disiplin gibi yapı mühendisliği de optimizasyon yöntemlerindeki gelişim ve sunduğu imkanlardan etkilenmiştir. Önceleri deneme-yanılma yöntemine başvurularak yapılan ve tasarımı gerçekleştiren mühendisin tecrübesi ile çözüm bulma süresinin doğrudan ilintili olduğu problem çözümleri, gelişen optimizasyon yöntemleri sayesinde optimizasyon algoritmalarına ve uzman sistemlere devredilmeye başlanmıştır.

1.1 Tezin Kapsamı

Bu çalışmada çelik yapıların kaynaklı birleşimlerinin metasezgisel algoritmalar yardımı ile optimum tasarımı incelenmiştir. Bu bağlamda, köşe kaynaklar ile birbirine bağlanmış profillerden oluşturulmuş birleşim modelinden yükleme durumlarına göre üç adet sayısal model elde edilmiştir. Elde edilen sayısal modellerin çözümünde Çelik Yapıların Tasarım, Hesap ve Yapım Esaslarına Dair Yönetmelik (ÇYTHYE) (2018)'da tanımlanan Güvenlik Katsayılarına ile Tasarım (GKT) ve Yük ve Dayanım Katsayıları ile Tasarım (YDKT) prensipleri kullanılmıştır. Optimizasyon için Jaya Algoritması, Teaching and Learning Based Optimization (Öğrenme ve Öğretme Tabanlı Optimizasyon) (TLBO) ve Sinüs – Kosinüs Algoritması (SCA) kullanılmıştır. Kısıtlı optimizasyon problemlerinin cezalandırılması için özgün bir yöntem olan “Adil Ceza Yöntemi” sunulmuştur. Analizler

çeşitlik kıyaslama işlevlerinin elde edilmesi ve algoritma tutarlılıklarının incelenmesi amacıyla üç farklı popülasyon büyüklüğü ile gerçekleştirilmiştir. Bütün kodlar MATLAB programlama dilinde yazılmış olup herhangi bir hazır kod ya da toolbox kullanılmamıştır. Çalışma kapsamında birbirinden farklı 54 adet analiz gerçekleştirilerek sonuçları irdelenmiştir. Algoritmaların çelik yapıların kaynaklı birleşimlerinin optimize edilmesinde kullanılabilirliğinin uygunluğu araştırılmış, elde edilen sonuçlar değerlendirilerek algoritmaların performansları karşılaştırılmıştır.

Tezin 2. bölümünde literatüre ilişkin genel bilgiler sunulmuştur. Kaynaklı birleşimler ve metasezgisel optimizasyona ilişkin genel bilgiler verildikten sonra, yapısal optimizasyon kavramı ile ilgili literatür çalışmalarına değinilmiştir.

Tezin 3. bölümünde çalışmada kullanılan metasezgisel algoritmalar sayısal örneklerle detaylı olarak açıklanmıştır. Çalışmada kullanılan yönetmelik ÇYTHYE (2018) için uyulması gereken esaslar belirtilmiştir. Sayısal modeller ve optimizasyon algoritmalarının ilgili modeller için kurulumları ile sonuçların değerlendirilmesi için izlenecek yöntemler detaylı olarak açıklanmıştır. Bu bölümde kısıtlı optimizasyon problemlerinin çözümü için özgün bir yöntem olan “adil ceza yöntemi” de tanıtılmıştır.

4. bölüm önceki bölümde detayları açıklanan 54 adet analizin bulgularını içermektedir. Elde edilen en iyi ve ortalama değerler, karşılaştırma parametreleri ve grafikler toplu olarak bu bölümde sunulmuştur. 5. Bölümde ise sonuç ve önerilere yer verilmiştir.

1.2 Tezin Amacı

Bu tez çalışmasının amacı, çelik yapıların kaynaklı birleşimlerinin optimum tasarımında Jaya, TLBO ve SCA metasezgisel optimizasyon algoritmalarının kullanılabilirliğini araştırmak, algoritmaların performanslarını değerlendirerek birbirlerine göre avantaj ve dezavantajlarını ortaya koymak, çalışma içerisinde önerilen adil ceza yönteminin kısıtlı optimizasyon problemlerinin çözümünde kullanılabilirliğini incelemektir.

1.3 Tezin Özgün Değeri

Yapısal optimizasyon literatürü incelendiğinde çalışmalardaki genel eğilimin yapı sistemlerinin genel hatları ile optimizasyonu olduğu gözlemlenmiştir. Bu bağlamda yapı elemanları için profil listelerinden uygun profillerin seçtirilmesi gibi yöntemlerle pek çok çalışma yapılmıştır. Bu tür çalışmalarda birleşim hesapları bazı kabullerle sınırlı kalarak bulonlu ya da kaynaklı birleşim detayları için optimizasyon yapılması yerine analizlerin genellikle rijit ya da yarı rijit kabul ekseninde ilerlediği görülmüştür. Bu çalışmalardan bazıları şunlardır: Kameshki ve Saka (2001), Hayalioğlu ve Değertekin (2005), Saka (2009), Toğan (2012). Bu çalışmada, kaynaklı birleşimlerin kaynak alanlarının optimize edilmesi amaçlanmıştır. Minimum kaynak miktarları için Jaya, TLBO ve Sinüs-Kosinüs algoritmaları ile optimum sonuçlar ve bu sonuçlara ait kaynak kesitleri araştırılmıştır. İlgili metasezgisel algoritmaların kaynaklı birleşimler için kullanılabilirliği araştırılarak 54 farklı analiz yapılmış ve sonuçlar karşılaştırılmıştır. Çalışmanın özgün değerini arttıran bir diğer faktör ise kısıtlı optimizasyon problemlerinin çözümü için bu çalışma kapsamında geliştirilen adil ceza yöntemidir.

2. LİTERATÜR BİLGİLERİ

2.1 Kaynaklı Birleşimler

Alaşımı belirli oranda benzer ya da aynı olan metallerin ısı ve/veya basınç etkisiyle birleştirilip yekpare hale getirilmesi işlemine kaynak denir. Çözümeyen bir bağlantı türü olan kaynak, belirli oranda malzeme tasarrufu sağlaması ve estetik kaygılara pratik çözümler sunması ile ön plana çıkmaktadır. Bazı kaynak türlerinde benzer alaşımlı ilave metaller (elektrot, tel vb.) kullanılsa da esas metalin eritilmediği lehimleme yöntemi ile karıştırılmamalıdır. Kaynak işlemi için pek çok farklı enerji kaynağı tercih edilebilir. Bunlardan en çok kullanılanları gaz alevi (kimyasal), elektrik arkı (elektrik), lazer, elektron ışını ve sürtünme olarak ifade edilebilir. Kaynak; genellikle fabrika koşullarındaki kontrollü ortamlarda yapılırsa da açık havada, su altında hatta uzayda dahi uygulamak mümkündür. Avantajlarına karşın kalifiye personel ihtiyacı ve kontrol hizmeti gerekliliği bulunmaktadır (Tama 2009).

2.1.1 Kaynaklı Birleşimlerin Tarihçesi

İnsanlık günümüzden birkaç bin yıl önce iki metal parçasını çekiçe döverek birbirine bağlamış ve birleştirmeyi başarmıştır. Bilinen en ilkel kaynak metodu olan bu tekniğe “demirci kaynağı” ya da “dövme kaynağı” adı verilmiştir. En eski örnekleri bronz ve demir çağlarında görülmektedir. MÖ 5. yüzyılda Antik Yunan tarihçisi Herodot, Sakız Adalı Glaucus’un demir kaynağının mucidi olduğunu belirtmiştir (Herodotus 1998). Hindistan’ın Delhi şehrinde MS 310 yılında inşa edilen 5,4 ton ağırlığındaki sütun, kaynağın yapısal anlamda kullanılmasının ilk örneklerinden birisi kabul edilmektedir (Carry ve Helzer 2005, İnt. Kyn. 1).

Orta çağa kadar demircilerin ısınan metali defalarca dövmesi ile gerçekleşen dövme kaynağı işlemi küçük gelişmelerle devam ettirilmiştir. Vannoccio Biringuccio’nun dövme kaynağının açıklamalarını içeren De La Pirotechnia’yı 1540 yılında yayınlamasıyla kaynak için yazılı literatür oluşmaya başlamıştır. Rönesans dönemiyle birlikte endüstri büyümeye devam etmiştir (Anonim 1994).



Resim 2.1 Hindistan’da yapılmış kaynaklı demir sütun (İnt. Kyn. 1).

Sir Humphry Davy 1800 yılında kısa darbeli elektrik arkını keşfederek sonuçlarını yayınlamıştır (Ayrton 1902, Anonim 1994, Anders 2003). 1802 yılında Vasily Petrov sürekli elektrik arkını gerçekleştirmeyi başararak 1803 yılında çalışmalarını yayınlamıştır (Lazarev 1999, Anders 2003). Bu çalışma kararlı bir ark deşarjını açıklayarak pek çok olası uygulama yöntemi için kapı aralamıştır. 1881-1882 yıllarında Nikolai Benardos ve Stanisław Olszewski karbon elektrotlar üreterek bu elektrotları kaynak işleminde kullanmışlardır. Bu sayede karbon ark kaynağı olarak isimlendirilen ilk elektrot kaynağı ortaya çıkmıştır. 1800’lerin sonlarında birbirlerinden bağımsız olarak Nikolai Slavyannov ve C.L. Coffin tarafından metal elektrotlar icat edilmiştir. 1900’lerin başında A.P. Strohmanger daha kararlı bir ark ortaya koyan kaplanmış metal elektrotu sunmuştur. 1905 yılında Vladimir Mitkevich tarafından üç fazlı elektrik arkı kullanılması önerilmiştir. Alternatif akım kaynağı C.J. Holslag tarafından 1919 yılında icat edilse de popülerleşmesi için on yıldan fazla zaman geçmiştir (Carry ve Helzer 2005).

19. yüzyılın sonlarında direnç kaynağı icat edilmiştir. Bu konudaki ilk patentler Elihu Thomson’a aittir. Termit kaynağı 1893’te icat edilirken bu süreçte başka bir yöntem olan oksî-gaz kaynağında ciddi gelişmeler sağlanmıştır ve kullanımı yaygınlaşmıştır. Oksî-gaz kaynağında ciddi bir öneme sahip olan asetilen 1836 yılında Edmund Davy tarafından keşfedilse de uygun torç geliştirilmesi ancak 1900’lü yıllarda mümkün olduğundan oksî-gaz kaynağının kullanımı pratik bulunmamıştır (Carry ve Helzer 2005). Oksî-gaz kaynağı

düşük maliyet ve mobilite avantajları sayesinde başlarda çok popüler olsa da 20. yüzyılda popülerliğini kaybetmiştir. Metal kaplama teknolojisindeki ilerlemelerin neticesi olarak oksijen-gaz kaynağının yerini ark kaynağı almaya başlamıştır (Weman 2007). Elektrotun dışındaki kaplama malzemeyi kirlilikten korurken aynı zamanda arkı stabilize etmektedir. Günümüzde kaplama teknolojisine geldiği son noktada farklı amaçlar için pek çok ürün bulmak mümkündür.

Birinci dünya savaşı atmosferindeki silahlanma yarışı kaynak teknolojisine yaygınlaşmasında ciddi bir fırsat oluşturmuştur. Ark kaynağının kullanıldığı ilk savaş makineleri Alman uçaklarının gövdeleri olmuştur (Anonim 1994). İngilizler “Fullagar” isimli savaş gemisinin inşasında ark kaynağı kullanmıştır (İnt. Kyn. 2). Kaynak teknolojisi askeri teknolojiye ilerlemesinin yanı sıra inşaat anlamında da ilerleme kaydetmiştir. 1927 yılında Stefan Bryla tarafından tasarlanan ve 1928 yılında Polonya’nın Łowicz yakınlarındaki Słudwia nehri üzerinde inşa edilen Maurzyce Köprüsü (Resim 2.2) dünyanın ilk kaynaklı yol köprüsü olarak kayıtlara geçmiştir (İnt. Kyn. 3).



Resim 2.2 İlk Kaynaklı Yol Köprüsü: Maurzyce Köprüsü (İnt. Kyn. 1).

1920’li yıllarda kaynak teknolojisinde elektrot telinin sürekli beslendiği otomatik sistemler de dahil olmak üzere pek çok büyük gelişim yaşanmıştır. Kaynakların atmosferdeki oksijen ve nitrojenin olumsuz etkilerinden korunması konusundaki çalışmalar neticesinde koruyucu gaz popüler bir konu haline gelmiştir. Gözeneklilik ve kırılabilirlik sorunlarına hidrojen, argon ve helyum ile koruyucu atmosfer oluşturulan yöntemler ile çözüm bulunmaya çalışılmıştır (Carry ve Helzer 2005). Otomatik kaynak

sistemleri, alternatif akımdaki gelişmeler ve kaplama teknolojisinin gelişmesi ile 2. Dünya Savaşı yılları ark kaynağının yayılmasında büyük etki göstermiştir (Anonim 1994).

20. yüzyılın ortalarında birçok yeni kaynak yöntemi bulunmuştur. 1930 yılında bugün hala popüler olan toz altı ark kaynağı ve sonradan gemi inşaat sektöründe popülerleşecek saplama kaynağı geliştirilmiştir. 1932 yılında Konstantin Khrenov tarafından ilk defa su altı ark kaynağı yapılmıştır. 1941 yılında gaz tungsten ark kaynağı gelişimini tamamlayarak mükemmel hale gelmiştir. Peşi sıra 1948’de pahalı koruyucu gazlara ihtiyaç duysa da gaz metal ark kaynağı gelişmiştir. Bu sayede demir içermeyen malzemelerin hızlı kaynaklanabilmesi de mümkün olmuştur. 1950’li yıllarda korumalı metal ark kaynağı kaplanmış elektrot kullanılarak gerçekleştirildikten sonra en popüler metal ark kaynağı olmuştur. İlerleyen yıllarda kendiliğinden korumalı tel elektrotun otomatik cihazlar ile kullanılabilir hale gelmesi kaynak imalat hızını oldukça yükseltmiştir. 1957 yılında Robert Gage tarafından plazma ark kaynağı icat edilmiştir. 1957 ve 1958 yıllarında birbirine oldukça yakın yöntemler olan Elektroslag ve Elektrogaz kaynağı tanıtılmıştır (Carry ve Helzer 2005). 1953 yılında N. F. Kazakov tarafından difüzyon bağlama yöntemi önerilmiştir (Kazakov 1985).

1958 yılında elektron ışını kaynağı ve 1960 yılında lazerin icat edilmesi ile takip eden on yıl içerisinde lazer ışını kaynağı 1967’de manyetik darbeli kaynak ve 1991 de sürtünmeli karıştırma kaynağı icat edilmiştir (Schwartz 2011). Bu dört yöntem pek çok ihtiyaca cevap verdiği için tüm dünyada yüksek kaliteli uygulamalar için kendisine yer bulsa da gerekli ekipmanların yüksek maliyetli olması yaygınlıklarını kısıtlamaktadır (Anonim 1994, İnt. Kyn. 1).

2.1.2 Teknolojilerine Göre Kaynak Yöntemleri

Kaynak yöntemlerini belirli özelliklerine göre sınıflandırmak mümkündür. Yapılış amacına göre birleştirme ya da dolgu kaynağı; uygulanma biçimine göre el kaynağı, otomatik kaynak, yarı mekanize kaynak; yapılacak işlemin cinsine göre basınç kaynağı ya da eritme kaynağı olarak gruplandırmak mümkündür. Metal kaynağı; ısı ve/veya

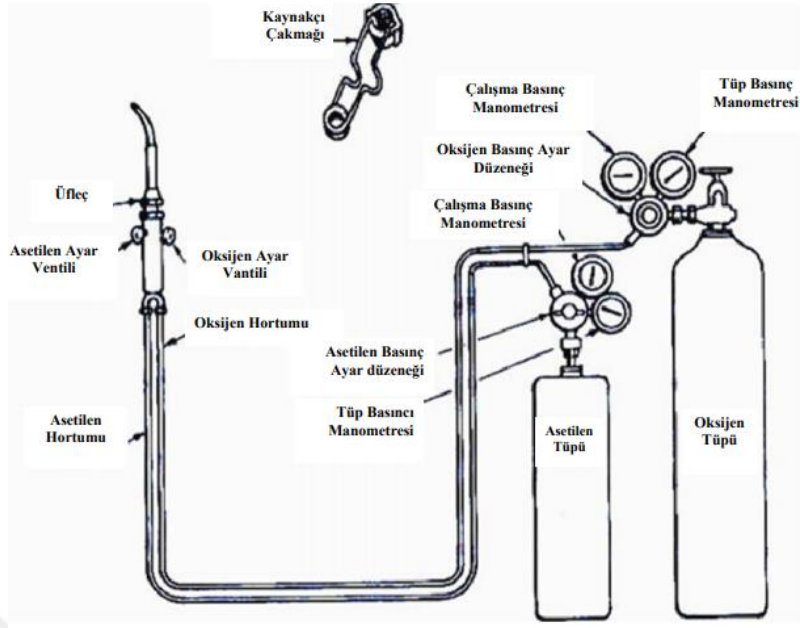
basınç etkisi ile, aynı ya da benzer erime aralığına sahip bir malzeme kullanarak ya da kullanmadan birleştirme işlemine verilen addır. Esas metaldeki ısı kaynak esnasında ergime noktasına kadar yükseliyorsa yapılan işleme Ergitme kaynağı, plastik kıvam oluşturacağı kadarsa basınç kaynağı adı verilmektedir. Çelik yapılarda kullanılan kaynaklar eritme kaynağıdır. Bu yöntemin en yaygın temsilcileri ise “gaz eritme kaynağı” ve “elektrik ark kaynağı” yöntemleridir (Tama 2009). Çelik yapı uygulamalarında yaygın olarak kullanılan kaynak yöntemlerinden bazıları aşağıda açıklanmıştır.

2.1.2.1 Gaz Eritme Kaynağı

Bir eritme kaynağı yöntemidir. Bu kaynak teknolojisinde biri yanıcı diğeri yakıcı özellikte olan iki gazın birlikte yakılması reaksiyonundan elde edilen ısı kullanılmaktadır. Yakıcı gaz olarak genelde oksijen; yanıcı gaz olarak da asetilen, hidrojen, metan, propan, bütan, LPG, benzin, benzol buharı gibi gazlar tercih edilir. İmalatta yaygın olarak kullanılan yanıcı gaz asetilendir. Bunun temel sebebi asetilen gazının tutuşma hızı, ısı değeri ve alev sıcaklığının diğeri gazlardan daha yüksek olmasıdır. Bütün bu avantajların doğal sonucu olarak, gaz eritme kaynağının en yaygın uygulaması oksii-asetilen kaynağıdır. Oksii-asetilen kaynağı için gerekli teçhizat Şekil 2.1’de gösterilmiştir. Oksii-asetilen kaynağı ile yapılan kaynak dikişlerinin mukavemeti diğeri yöntemlere göre düşük olduğu için çelik konstrüksiyonun taşıyıcı yani kuvvet aktaran birleşimlerinde tercih edilmemektedir. Yaygın kullanımı şantiye ya da fabrika ortamında levha ve sac kesimidir. Sac kesim işlemi Resim 2.3’de gösterilmiştir. Bilgisayar desteği ile otomatik kesim yapan CNC plazma teknolojisinin temelinde oksii-asetilen kaynağı bulunmaktadır (Tama 2009).



Resim 2.3 Oksii-asetilen kaynağı ile metal kesimi işlemi (Tama 2009).

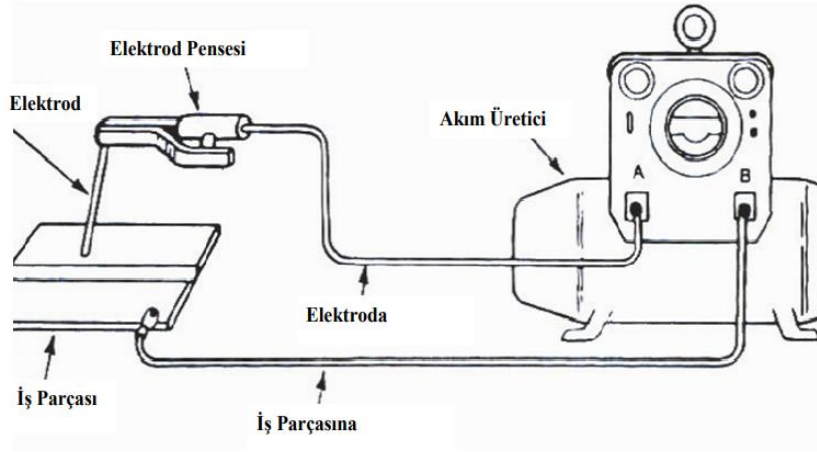


Şekil 2.1 Oksi-asetilen kaynağı için donanım şeması (Tama 2009).

2.1.2.2 Elektrik Ark Kaynağı

Kaynak işlemi için gerekli enerji elektrik arkı ile elde edilmektedir. Günümüzde çelik yapılarda en çok tercih edilen yöntemdir. Ek kaynak metalinin yani elektrotun içeriğine, uygulanma biçimine ve kaynağı korumak için kullanılan ilave koruyucu malzemelere göre çeşitli sınıflara ayrılmaktadır. Çelik yapılarda en çok kullanılan kaynak yöntemleri elektrot kaynağı, tozaltı kaynağı ve gazaltı kaynağıdır.

Elektrot kaynağı, kaynak metali olarak genellikle örtülü elektrotların kullanıldığı kaynak teknolojisidir. Örtülü elektrotla kaynak yöntemi, elektrikli ark kaynağı teknolojisi içerisinde en basit ve en çok uygulanan yöntemdir. 1,2 mm'den daha kalın parçalara her kaynak pozisyonunda uygulamak mümkündür. Kaynak için gereken akım bir akım üretici trafo tarafından sağlanır. Akım kablolar vasıtası ile iş parçası ve kaynak pensine iletilir. Pense elektrot takılarak iş parçasına elektrot ucu değdirilir ve elektrik arkı oluşturulur (Şekil 2.2, Resim 2.4). Ark oluşumunun başlaması, yanma, kaynak boyutunun ayarlanması, kaynak hızı, söndürme, kaynak ağzının doldurulması gibi işlemler tamamen kaynakçı tarafından yapılmaktadır. İnsan unsurunun bu kadar fazla olması personel tecrübesi ile kalite arasında yadsınamaz bir bağ kurmaktadır.



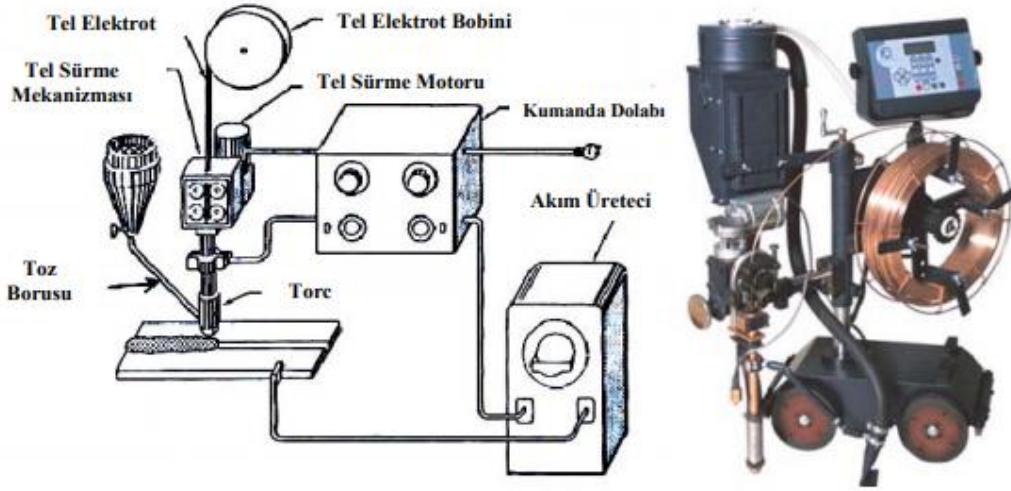
Şekil 2.2 Elektrot kaynağı için donanım şeması (Tama 2009).

Ark iş parçası ve elektrot arasında oluşur. Eriyen elektrot kaynak metalini oluşturur. Elektrot örtüsü de aynı anda eriyerek çıkan gaz ark bölgesini, cüruf ise kaynak dikişini örterek koruma sağlar. Kaynak makinelerinin diğer yöntemlerdeki makinelere göre görece ucuz olması, işlemin basit olması, kaynakçının büyük oranda hareket serbestliğine sahip olması ve aynı kaynak makinesinin elektrotları değiştirilerek farklı metallerin kaynağı için kullanılabilmesi gibi avantajları sayesinde çok yaygın bir kullanım ağına sahip olmuştur (Tama 2009).



Resim 2.4 Elektrot kaynağı imalatı (Tama 2009).

Tozaltı kaynağında ark, otomatik bir mekanizma ile iş parçasına değen çıplak tel ile iş parçası arasında oluşmaktadır. Kaynak bölgesine sürekli olarak toz dökülmektedir (Resim 2.5). Ark, dökülen bu tozun altında yanmaktadır. Toz örtüsü kaynağı atmosferin olumsuz etkilerinden korurken ısınım ile enerji kaybının da önüne geçtiği için ark enerjisinin yaklaşık %64'ü kaynak için harcanır (Şekil 2.3) (Tama 2009).



Şekil 2.3 Tozaltı kaynak için donanım şeması ve kaynak makinesi (Tama 2009).

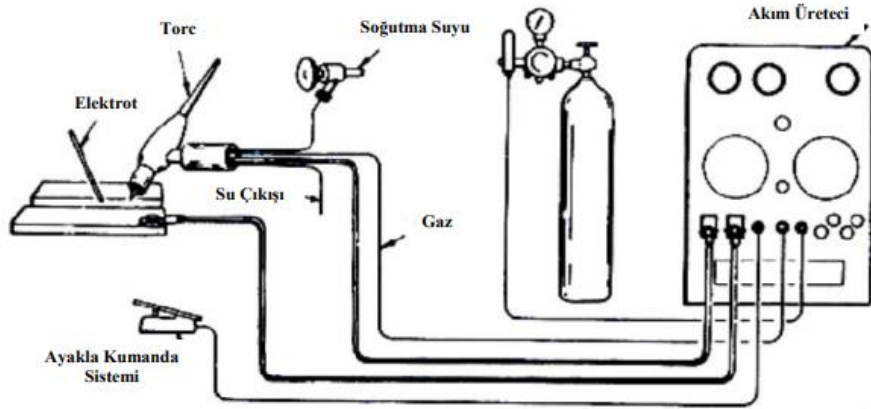
İlave metal ihtiyacı düşüktür. Özellikle uzun kaynakların imalatı için ($l > 90$ cm) uygun bir yöntemdir. Et kalınlığı farketmeksizin yüksek kaliteli köşe, yarım ve tam nüfuziyetli küt kaynak imalatında uygulanabilir. Avantajlarının yanı sıra yalnızca yatay pozisyonlar için kullanılabilir. Hareket kabiliyeti ve uygulama yüzeyine adaptasyonu düşüktür. Koruyucu tabaka vazifesi göre toz malzeme nemlenmeye meyilli bir yapıdadır. Nemli toz ile yapılan tozaltı kaynaklar gözeneklilik problemiyle karşı karşıya kalmaktadır. Bundan ötürü ilave kurutucu ekipman kullanılarak tozun devamlı kuru kalması sağlanmalıdır. Bu durum da yöntemin maliyetli oluşunun bir etmenidir (Kahraman ve Gülenç 2016).



Resim 2.5 Kaynak bölgesine dökülen toz malzeme (Tama 2009).

Gazaltı kaynağı, kaynak bölgesinin dış etkilere karşı gazla korunduğu özel bir ark kaynağı türüdür. Gazaltı kaynak makinelerinin ilk ayarları kaynakçı tarafından yapılırsa da genellikle devamındaki süreci otomatik olarak işletirler. Kaynak donanımları torç, kablolar, elektrot besleme ünitesi ve gaz ünitesidir. Kaynak torcunun tetiğine basıldığında

zaman elektrik akımı, elektrot ve koruyucu gaz iş parçasına aynı anda iletilmektedir. Kaynak kalitesi pek çok değişkene bağlıdır. Bunlardan bazıları koruyucu gazın cinsi, gaz debisi, ark boyu, elektrot çapı ve besleme hızı, kaynak hızı, serbest elektrot uzunluğu, kaynak pozisyonu olarak sayılabilir. Kaliteli bir kaynak için bütün bu parametreler üzerinde kontrol sahibi olmak gerekmektedir. Bu sebeple gazaltı kaynağının tecrübeli ve donanımlı personel ihtiyacı yüksektir. Gazaltı kaynağı ile, sürekli ve uzun kaynakların elde edilmesi kolaydır. Tüm pozisyonlarda kaynak yapmak mümkündür. Elektrot beslemesi sürekli ve otomatik, metal yığılma hızı da yüksek olduğundan işlem süresi kısadır. Kaynak bölgesi koruması gaz ile sağlandığından cüruf oluşumu görülmez bundan dolayı temizlik için harcanacak zamandan da tasarruf edilmiş olur. Otomasyon uygulamalarında başarılı sonuçlar vermektedir. Bu sayede üretim hatlarında sürekli olarak kullanılabilir. Bütün bu avantajlarına karşın, elektrotlu ark kaynağı gibi basit yöntemlere göre daha maliyetlidir. Daha karmaşık ve büyük teçhizatlar ile yapıldığından taşınması zordur. Spesifik olarak ulaşılması gereken bölgelerde kullanımı zordur. Koruyucu gaz kaynak arkının olduğu bölgede muhafaza edilmelidir. Gazı uzaklaştıracak hava akımlarına engel olunmalıdır. Bu sebeple açık alanda kullanımı kısıtlıdır. Kullanılmak istenirse hava akımlarına karşı önlem alınmalıdır. Kullanılan elektrot ve koruyucu gazlara göre sınıflandırılabilirler. Elektrot olarak tungsten tercih edildiğinde TIG (Tungsten Inert Gas) kaynağı, koruyucu gaz olarak argon veya helyum kullanıldığında MIG (Metal Inert Gas) kaynağı, aktif bir gaz olan karbon monoksit koruyucu gaz olarak kullanıldığında ise MAG (Metal Active Gas) kaynağı olarak isimlendirilirler (Tama 2009).



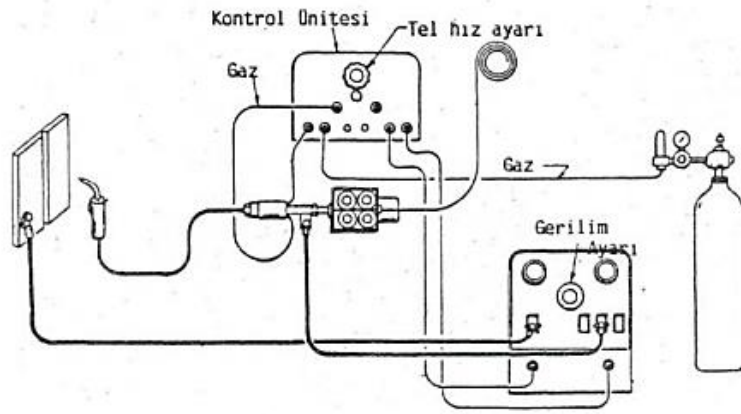
Şekil 2.4 TIG kaynak için donanım şeması (Tama 2009).

TIG kaynağında koruyucu gaz olarak helyum ya da argon kullanılır. Kaynak arkı erimeyen bir tungsten ya da tungsten alaşımı bir elektrot ile iş parçası arasında oluşmaktadır. Kaynak metaline ihtiyaç duyulur (Şekil 2.4, Resim 2.6) (Tama 2009).



Resim 2.6 TIG kaynak uygulaması (Tama 2009).

MIG kaynağında ark asal gaz (argon ve helyum) atmosferi ile korunan bir ortamda oluşmaktadır. Bu kaynağı TIG kaynağından ayıran en büyük özellik, arkın sürekli eriyen bir elektrot ile iş parçası arasında oluşmasıdır (Ertürk, 1987). MIG kaynağının donanımları Şekil 2.5’de gösterilmiştir.



Şekil 2.5 MIG kaynak için donanım şeması (Ertürk 1987).

MAG kaynağı 1950’li yıllarda geliştirilerek otomotiv endüstrisinde büyük yer bulmuştur. Başlangıçta yatay pozisyonda kaynak yapılabilirken sonraki yıllarda ince çaplı elektrotlar kullanılabilir hale geldiğinden pozisyon fark etmeksizin kaynak yapabilme kabiliyeti kazanmıştır. Koruyucu gaz olarak karbondioksit kullanılmıştır. İhtiyaç halinde arkı yumuşatmak ve sıçramaları minimize etmek için argon ile karbondioksit karışık

kullanılabilmektedir. MIG/MAG kaynak teknolojisi oldukça fazla ayar imkânı ile ince ve oldukça kalın kesitli pek çok parçanın birleştirilmesinde kullanılan yaygın bir yöntemdir. MIG uygulaması Resim 2.7’de gösterilmiştir (Tama 2009).



Resim 2.7 MIG kaynak uygulaması (Tama 2009).

2.1.3 Uygulanma Biçimlerine Göre Kaynak Çeşitleri

Kaynaklar, uygulanma biçimlerine göre farklı yük taşıma mekanizmalarına sahiptirler. Bu uygulanma biçimlerini üç alt grupta incelemek mümkündür. Bunlar; küt kaynaklar, köşe kaynaklar ve dairesel ya da oval dolgu kaynaklardır. Birleşim içerisinde birlikte kullanılmaları söz konusu olduğunda her biri için ayrı hesap yapılması gerekmektedir. Kaynaklar ile ilgili teknik gerekler TS EN 1090-2 (2018)’de hesap esasları ise Çelik Yapıların Tasarım, Hesap ve Yapım Esaslarına Dair Yönetmelik (2018) (ÇYTHYE 2018)’de açıklanmıştır. Yapılarda kullanılan kaynakların %80’i köşe kaynak, %15’i küt kaynak, %5’i ise dolgu kaynakları ve diğer kaynakları ifade etmektedir (Jaspard ve Weynand 2017).

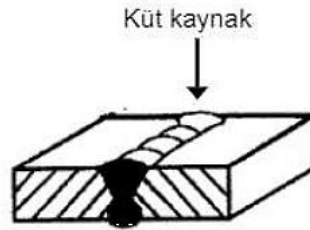
2.1.3.1 Küt Kaynak Dikişleri

Küt kaynaklar çoğunlukla aynı düzlemdeki iki elemanı birleştirmek amacıyla kullanılırlar da farklı düzlemlerdeki elemanlara uygulanması da mümkündür (Şekil 2.6). Çoğunlukla birleştirilecek levhaların işlenmesi ihtiyacı ortaya çıkmaktadır. Bu işleme kaynak ağzı açılması adı verilmektedir. İşlem sonucunda levha kenarlarında oluşturulan

geometrik şekle göre farklı isimlerle anılan küt kaynak çeşitleri vardır. Bunlardan bazıları I, V, Y ve U'dur (Deren vd. 2008) (Çizelge 2.1). Birleşen elemanlardan en incenin kalınlığı kaynak kalınlığını belirlemektedir.

Çizelge 2.1 Bazı küt kaynak dikişlerinin isimleri (Deren vd. 2008).

Dikiş İsmi	I Dikişi	V Dikişi	Y Dikişi	X Dikişi	U Dikişi
Levha Kenarları Şekli					
Kaynak İşareti					
Levha t kalınlığı	≤ 5	3~20	8~20	18~40	>16



Şekil 2.6 Küt kaynak.

Küt kaynaklar tüm levha kalınlığı boyunca yapılırsa tam penetrasyonlu küt kaynak, tüm levha kalınlığında olmayacak şekilde yapılırsa kısmi penetrasyonlu küt kaynak olarak iki alt grupta incelenirler. Özellikle ağız açma işleminin getirdiği işlemler dolayısıyla köşe kaynaklardan daha maliyetlidirler. Resim 2.8 ve Resim 2.9'da küt kaynak dikişinin sırasıyla kiriş ekine ve kolon ekine uygulandığı görülmektedir.



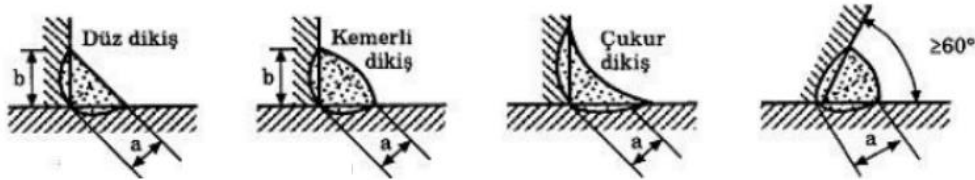
Resim 2.8 Küt kaynağın kiriş ekine uygulanması.



Resim 2.9 Küt kaynağın kolon ekine uygulanması.

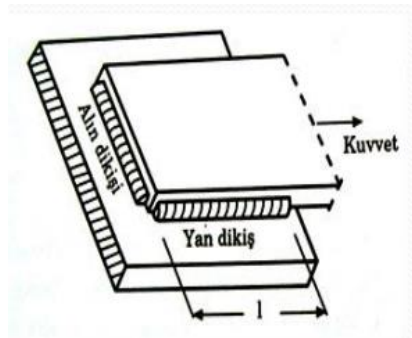
2.1.3.2 Köşe Kaynak Dikişleri

Birleştirilecek parçaların kesişen yüzeylerinin arasında kalan bölgenin kaynak metali ile doldurulması durumunda oluşan ve en kesit alanı üçgen olarak kabul edilen kaynaktır (ÇYTHYE 2018). Birleşen yüzeyler arasındaki açı minimum 60° olmalıdır. Daha küçük açılar için kaynağın yük taşımadığı kabul edilmektedir. Kaynak en kesiti içerisine çizilebilecek olan hayali üçgenin yüksekliği kaynak kalınlığı a 'ya eşittir (Şekil 2.7).



Şekil 2.7 Köşe dikiş çeşitleri ve yükseklikleri (Gaylord vd. 1992).

Sisteme uygulanan kuvvetin doğrultusu kaynak doğrultusu ile aynı ise ilgili kaynağa “yan dikiş”, farklı doğrultularda ise ilgili kaynağa “alın dikişi” ismi verilir (Şekil 2.8).



Şekil 2.8 Kuvvet yönüne göre köşe kaynaklar (Deren vd. 2008).

Üst üste gelen bütün levhaların birleştirilmesi için kullanılmaktadır. Küt kaynaklar gibi kaynak ağzı açma süreci olmadığı için daha ekonomik bir çözüm sunmaktadır. Sunduğu birleşim geometrisi ve ekonomi avantajı ile çelik yapılar için en çok kullanılan kaynak dikişi olmuştur. Resim 2.10 ve Resim 2.11’de köşe kaynak uygulamalarından görseller sunulmuştur. Resim 2.10’da verilen bağlantının kuvvet aktardığının kabul edilebilmesi için iki parça arasındaki açının 60°’den büyük olması gerekmektedir.



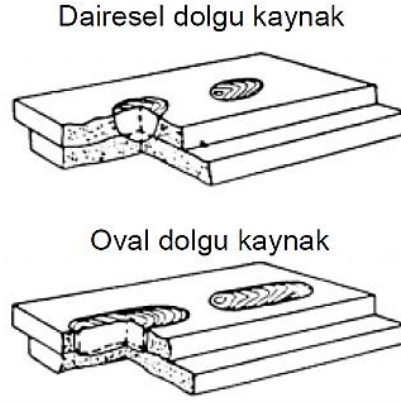
Resim 2.10 Köşe kaynak uygulaması.



Resim 2.11 Açılı köşe kaynak uygulaması.

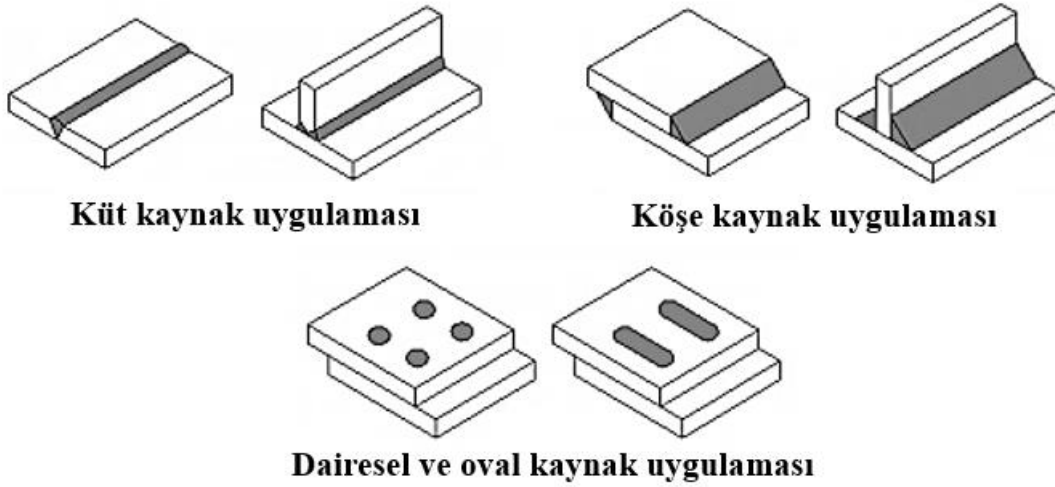
2.1.3.3 Dairesel ve Oval Dolgu Kaynaklar

Dairesel ya da oval olarak açılmış deliklerin kaynak metali ile doldurularak kapatılması için yapılan kaynağa verilen isimdir (Şekil 2.9). Bu tür kaynakların yapıldığı elemanlarda kaynak alanları net en kesit alanlarına eklenmemelidir.



Şekil 2.9 Dairesel ve dolgu oval kaynaklar.

Kaynak dikişlerinin tamamı Şekil 2.10’da gösterilmiştir. Şekilde de görülebileceği gibi benzer bağlantı türleri farklı dikişler ile gerçekleştirilebilmektedir. T bağlantı biçimi ağız açarak ve aynı düzlemdeki yüzeylerin arası kaynak metali ile doldurularak bağlanırsa küt kaynak, bağlanacak plakalar üst üste koyularak farklı düzlemlerdeki yüzeylerinin arası kaynak metali ile doldurulursa köşe kaynak olarak değerlendirilecektir.



Şekil 2.10 Kaynak dikişleri (Yorgun vd. 2017).

2.2 Metasezgisel Optimizasyon Yöntemi

Metasezgisel (metaheuristic) kelimesi, Yunanca yeni strateji bulma sanatı anlamına gelen “heuriskein” sözcüğünün üst seviye anlamındaki “meta” ön ekiyle birleşmesinden oluşmuştur. Spesifik optimizasyon problemlerinin çözümü için ortaya konmuş, belirli stratejileri temel alan üst düzey genel metodolojileri ifade etmektedir (Talbi 2009).

Farklı karmaşıklığa sahip optimizasyon problemlerini çözmek için pek çok matematiksel programlama yöntemi ortaya atılmıştır. En sık kullanılan matematiksel programlama yöntemleri ise doğrusal programlama (DP), doğrusal olmayan programlama (DOP), ve tamsayı programlamadır (TP). DP, doğrusal bir amaç işleviyle ifade edilip doğrusal eşitlik ve eşitsizlik kısıtlarına sahip problemlerinin çözümü için geliştirilmiştir. TP, doğrusal programlamadaki bütün tasarım değişkenlerinin tam sayı olduğu özel durumu ifade etmektedir. Özellikle doğrusal amaç fonksiyonlarına sahip olan listeden eleman seçimi problemlerinde tercih edilmektedir. DP ve TP için tabiatı gereği yerel optimum (yerel minimum ya da yerel maksimum) noktalarından bahsedilemez. Çözümü neticesinde elde edilen sonuç daima global optimumu vermektedir. DOP, optimizasyon probleminde herhangi bir fonksiyonun doğrusal olmadığı durumlar için geliştirilmiştir. Bu tür problemlerin çözümü için gradyan yöntemi, Lagrange çarpanarı yöntemi, ceza fonksiyonu yöntemi gibi pek çok yöntem geliştirilmiştir (Haftka ve Gürdal 1991, Rao 1996).

Yukarıda bahsedilen klasik optimizasyon yöntemleri, nispeten küçük arama alanlarındaki optimizasyon problemleri için uygulanabilir olsalar da tasarım değişkeni sayısı yani problemin boyutu arttıkça büyüyen arama alanıyla birlikte performanslarını kaybetmektedirler. Problemin çözümü için geçen süre arttıkça tasarımcılar açısından pratikliğini yitiren yöntemler, yeni optimizasyon yöntemlerinin gelişmesine de zemin hazırlamıştır (Pham ve Karaboğa 2000). Metasezgisel algoritmalar bu noktadaki ihtiyaca pratik bir çözüm üretmektedir. Metasezgisel algoritmalar, doğadan, fizikten, bir canlının av ararken ya da eş seçerken sergilediği davranışlardan, matematikten ve pek çok diğer alanlardan belirli stratejileri temel alan algoritmalarlardır. İlham kaynakları neticesinde ortaya bir strateji çıkmaktadır. Bu strateji, tasarım değişkenlerinin sınırlarıyla oluşturulan arama alanındaki en iyi sonucu bulmaya çalışırken rehber niteliğinde görev yapmaktadır. Metasezgisel algoritmalar esin kaynaklarına göre; sürü tabanlı, evrimsel ve fiziksel algoritmalar olmak üzere üç ana kategoride değerlendirilebilirler (Yildiz vd. 2020). En çok bilinen sürü tabanlı algoritmalar şunlardır: Parçacık Sürü Optimizasyonu (Kennedy ve Eberhart 1995), Karınca Kolonisi (Dorigo vd. 2006), Yapay Arı Kolonisi (Karaboğa ve Baştürk 2007), Ateş Böceği (Yang 2010) ve Cuckoo Arama Algoritması (Gandomi vd. 2013). Evrimsel temelli algoritmalar en yaygın kullanılanlar ise Genetik

Algoritma (Holland 1975), Diferansiyel Evrim (Storn ve Price 1995), Evrim Stratejisi (Beyer ve Schwefel 2002) ve Genetik Programlama (Koza 1992)'dir. Fiziksel algoritmaların en bilinenleri ise Tavlama Benzetimi (Kirkpatrick 1983), Büyük Patlama Büyük Çöküş (Erol ve Eksin 2006), Kara Delik (Hatamlou 2013) ve Işın Optimizasyonu (Kaveh ve Khayatazad 2012)'dur.

Metasezgisel algoritmalar belirli bir probleme özel geliştirilmemiştir. Çoğu algoritma farklı optimizasyon problemlerinin çözümünde kullanılabilir. Metasezgisel algoritmaları diğer yöntemlerden ayıran en önemli özelliklerden birisi de amaç fonksiyonunun her zaman daha iyiye gitmek gibi bir zorunluluğun olmamasıdır. Arama yapılırken rastlantısallık da önemli bir parametre olarak etkililmektedir. Bu sayede arama alanının daha kolay taranması dolayısıyla çözüm süresinin kısaltılması sağlanırken yerel optimum noktadaki takılmanın da önüne geçilmeye çalışılır. Diğer bir deyişle, amaç fonksiyonunun daha kötüye gitmesi göze alınarak daha geniş bir alanı tarama imkânı yaratılmaktadır. Bütün bu avantajlarına karşın metasezgisel algoritmalar global optimumu bulmayı garanti etmezler. Bu zafiyetten kaçınmak amacıyla genellikle farklı metasezgisel algoritmalar aynı probleme uygulanarak sonuçlar karşılaştırılır.

2.2.1 Amaç Fonksiyonu

Optimizasyon probleminde hedeflenen amacı sembolize eden matematiksel ifadedir. Amaç fonksiyonu problemin amacını ne kadar iyi modellerse yapılan optimizasyon da o kadar gerçekçi sonuçlar verecektir. Her bir aday çözüm için hesaplanarak hangi çözümün daha başarılı olduğunun anlaşılmasını sağlar. Algoritmalar bu sayede daha verimsiz çözümden kaçınıp daha iyi çözüme doğru ilerlediği açgözlü seçim prosedürünü başarıyla tamamlar. Optimizasyon probleminin yapısına bağlı olarak tek veya birden fazla amaç fonksiyonu olabilir. Yapısal optimizasyon problemleri ele alındığında, bina toplam ağırlığı, bina maliyeti, düğüm noktalarındaki yer değiştirmeler, iç kuvvetler sonucu ortaya çıkan gerilmeler gibi parametrelerden biri ya da birkaçı amaç fonksiyonu olarak seçilebilir. Maliyet minimizasyonunun hedeflendiği bir yapısal optimizasyon probleminde, amaç fonksiyonu olarak yapı ağırlığı ya da yapı maliyeti seçilebilir. Amaç fonksiyonu yapı ağırlığı olarak seçilip algoritmanın işleyişi minimizasyon olarak

kurgulandığında algoritma doğrudan yapı ağırlığını azaltacak şekilde hareket edecektir. Azalan yapı ağırlığı neticesinde maliyet optimize edilmiş olur.

2.2.2 Tasarım Değişkenleri

Tasarım değişkenleri bir problemin amaç fonksiyonundaki girdilerini ifade eder. Her bir tasarım değişkeni amaç fonksiyonunu elde etmede doğrudan ya da dolaylı olarak kullanılmalıdır. Diğer bir deyişle problemin boyutu kadar yani bilinmeyen sayısı kadar tasarım değişkeni vardır. Başlangıçta tasarım değişkenleri kullanıcı tanımlı olarak belirlenmiş sınırlar arasında rastgele tanımlanır. Sonrasında ilgili metasezgisel algoritmanın işleyişine göre çeşitli değişikliklere uğratılarak en iyi amaç fonksiyonunu veren tasarım değişkenlerine ulaşılmaya çalışılır. Tasarım değişkenleri kesikli ya da sürekli değerler olarak tanımlanabilir. Amacı minimum en kesit alanını bulmak olan bir yapısal optimizasyon problemi ele alındığında, tasarım değişkenlerinin tamamı en kesit alanının hesabına etkiyen parametreler (ilgili kesitin ayrıtları gibi) olarak tanımlanırlar. Yapısal optimizasyon özelinde bakılacak olursa, ayrık tasarım değişkenleri genellikle hazır profil listelerinden en uygun boyuttaki yapı elemanını seçme problemlerinde tercih edilmektedir. Sürekli tasarım değişkenleriyle yapılacak optimizasyon ile elde edilen sonuçlardaki en kesit ölçüleri piyasadaki profillerle örtüşmeyeceği için en uygun çözüm mevcut profillerin listelenip her bir profil için sıra numarası verilerek ayrık tasarım değişkeni tanımlamaktır. Tasarım değişkenlerinin seçimi, tanım aralıkları ve amaç fonksiyonuna ne derece etki ettikleri, optimizasyondan elde edilecek sonucu doğrudan etkilediği için, probleme uygun parametreler tanımlamak oldukça önemlidir.

2.2.3 Kısıtlar

Kısıtlar, elde edilecek optimum sonucun sağlanması gereken kriterleri ifade eder. Yapısal optimizasyon problemleri için bu kısıtlar genellikle şartname ve yönetmeliklerde istenen gerekleri ifade etmektedir. Örneğin bir çelik elemanın basınç kuvveti taşıyabilmesi için narinlik şartını sağlanması gerekmektedir. En ekonomik çözüm olsa dahi bu şartın sağlanmadığı bir çözüm kabul edilebilir değildir. Kısıtlar kodlanarak bu şartlar

metasezgisel optimizasyon algoritmalarına tanıtılır. Her bir aday çözüm için kısıtın sağlanma durumu kontrol edilir. Kısıtların sağlanmadığı durumlarda ceza işlevleri devreye girer.

2.2.4 Ceza Fonksiyonu

Optimizasyon problemine tanımlanan kısıtlar sağlanmadığında amaç fonksiyonunun cezalandırılması gerekir. Ceza fonksiyonu, ilgili aday çözüm için amaç fonksiyonunu cazip olmaktan uzaklaştıracak şekilde kurgulanmalıdır. Ceza fonksiyonları optimizasyon probleminin yapısına göre değişiklik göstermektedir. Önemli olan nokta, uygunsuz çözümlerin uygunsuzlukları ölçütünde adaletli bir şekilde cezalandırılarak dışlanmasının sağlanmasıdır.

2.3 Yapısal Optimizasyon ve Önceki Çalışmalar

Yapı mühendisliği alanında metasezgisel yöntemlerden önce de çeşitli yöntemlerle optimizasyon çalışmaları yapılmıştır. 1970'li yıllarda bilgisayar teknolojisinde yaşanan gelişmelerin bir sonucu olarak, yapısal optimizasyon alanındaki problemlerin çözümü için de büyük bir alan açılmıştır. Yapısal optimizasyon kavramı ilk defa Schmit (1960) tarafından yapılan çalışma ile ortaya konulmuştur. Yeni bir yapı mühendisliği dalı olarak görülen bu alan, yapısal tasarım problemlerini bir karar verme problemi olarak yeniden kurgulamaktadır. Karar verme, kişinin mevcut seçenekler arasında en iyi olana yöneldiği bilişsel bir süreci ifade etmektedir. Karar verme problemleri, belirlenmiş sınırlar altında çözümün kalitesini temsil eden bir amaç fonksiyonunu en aza indirgeyecek ya da en yükseğe çıkaracak şekilde modellenir. Tasarım değişkenleri, problemdeki kaynağın miktarını ve bazı faaliyetlerin düzeylerini ifade eder. Karar verme probleminin optimum çözümü, tasarım değişkenlerinin amaç fonksiyonunu en iyiye getirdiği değeri bulmayı hedefler. Yapısal tasarım süreci matematiksel olarak modellenirken; tasarım değişkenleri genellikle yapısal elemanın kesitsel özelliklerini, kısıtlar ise genellikle meydana gelen yer değiştirme ve gerilmedeki sınırlamaları ifade etmektedir. Amaç fonksiyonu genellikle yapının maliyetini en aza indirecek şekilde seçilmektedir (Saka vd. 2016).

Yapısal optimizasyon genellikle boyut, şekil ve topoloji optimizasyonu olmak üzere üç alt kategoride incelenir (Christensen ve Klarbring 2009). Örneğin bir çelik kafesin optimum tasarım süreci incelendiğinde boyut optimizasyonu elemanların en uygun kesit alanlarını bulmayı hedeflerken şekil optimizasyonu düğüm noktalarının konumlarını değiştirmeyi hedefler (Kaveh ve Talatahari 2009). Topoloji optimizasyonunda ise elemanların ve düğümlerin eklenip çıkarılmasıyla en uygun sonuca ulaşmaya çalışılmaktadır (Tejani vd. 2018). Yapısal optimizasyon yöntemleri ortaya konulmadan önce tasarımcılar tarafından deneme yanılma esaslı yöntemler kullanılmıştır. Deneme-yanılma esasına dayanan tasarımlarda, problemin kabul edilebilir bir çözüme ulaşma süresi doğrudan tasarımcının tecrübesiyle ilişkilidir. Denenmesi gereken seçenek sayısı arttıkça tasarımcının optimum sonucu bulma ihtimali de azalmaktadır. Bu durumda genellikle ilgili yönetmeliklerin kriterlerini sağlayan ve birkaç denemeyle elde edilen en iyi çözüm, problemin optimum sonucu olarak kabul edilmektedir. Bu kabul, ilgili yönetmeliklerin kriterlerini sağlamaktadır ancak ekonomik olarak optimumdan uzaktır.

Yapısal optimizasyonun tek faydası maliyetleri azaltmak değildir. Minimum maliyet hedeflenirken azalan yapı malzemesi kullanımının çevreye olan pozitif etkileri de bulunmaktadır. Bu bağlamda, Lagaros (2018) tarafından yapılan çalışmada, gelecekteki bina projelerinde yapısal optimizasyonun kullanılması neticesinde beton ve donatı gibi yapı malzemelerinin kullanımındaki azalmanın çevreye ve sürdürülebilirliğe olan etkileri incelenmiştir. Çalışmada yapısal optimizasyonun projeye özgü tasarruf miktarları %10 ile %20 arasında bulunmuştur. İlgili tasarrufun çevresel faktörlere olan etkileri incelenerek inşaat sektörünün sera gazı emisyonları üzerindeki etkisinin azaltılması için yapısal optimizasyon uygulamalarının yaygınlaşması ihtiyacına vurgu yapılmıştır.

Saka (1991) tarafından yapılan çalışmada, matematik tabanlı temel bir algoritma kullanılarak AISC'deki gerilme ve yer değiştirme kısıtlarını esas alan model Basic programlama dilinde oluşturulmuştur. Bu model ile muhtelif çelik çerçeve sistemlerin boyutlandırılması gerçekleştirilmiştir. Oluşturulan optimizasyon modeli, amaç fonksiyonunun oluşturulması ve yönetmelik kısıtlarının algoritmalara tanıtılması gibi konularda pek çok araştırmaya yol göstermiştir.

Daloğlu ve Armutçu (1998) tarafından yapılan bir çalışmada, düzlem çelik çerçeve sistemlerin Genetik Algoritma ile boyutlandırılması incelenmiştir. Çalışmada gerilme ve yer değiştirme sınırlayıcıları için TS 648 esas alınmıştır. Çalışmanın sonucunda Genetik Algoritma düzlem çerçevelerin optimizasyonu için oldukça başarılı bulunmuştur.

Daloğlu ve Aydın (1999) tarafından yapılan çalışmada Fortran yazılım dilinde oluşturulan Genetik Algoritma ile düzlem kafes sistemleri optimize edilmiştir. Optimizasyon problemi olarak yaygın olarak kullanılan çatı makasları seçilmiştir. Amaç fonksiyonu çatı makaslarının minimum ağırlığını bulmaya yöneliktir. Algoritmaya yer değiştirme, gerilme ve burkulma kısıtları uygulanmıştır.

Kameshki ve Saka (2001) tarafından, yarı rijit bağlanmış çerçeveler için P- Δ etkisini de dikkate alarak lineer olmayan bir yaklaşım ile boyut optimizasyonu yapılmıştır. Çalışmada optimizasyon algoritması olarak Genetik Algoritma tercih edilmiştir.

Hayalioğlu ve Değertekin (2005) tarafından yapılan çalışmada yarı rijit bağlı doğrusal olmayan çelik çerçeveler genetik algoritma ile optimize edilmiştir. Çalışmada AISC geniş flanş (W) kesitlerden çerçeve elemanları için optimum seçim yaptırılmıştır. AISC için LRFD prensibi ile gerilim, yer değiştirme ve elemanlar için boyut kısıtları uygulanmıştır.

Kaveh ve Talatahari (2007) tarafından birçok algoritma birlikte kullanılarak, düzlem çelik çerçeveler AISC yönetmeliğindeki LRFD tasarım prensibi için optimize edilmiştir. Boyutlandırmalar gerilme ve yer değiştirme kısıtları altında yapılmıştır. Algoritmalarından elde edilen optimum sonuçlar kıyaslanmıştır. Benzer şekilde Saka (2009) tarafından yapılan çalışmada Harmoni Arama Algoritması ile BS-5950 yönetmeliğini kullanarak düzlem çelik çerçeve optimizasyonu gerçekleştirmiştir. Oluşturulan algoritma İngiliz yönetmeliğinde sunulmuş 64 kiriş ve 32 kolon kesitinden uygun olanları seçerek çerçeveyi optimize etmektedir. Benzer bir çalışma da Toğan (2012) tarafından yapılmıştır. Düzlem çerçeve için AISC yönetmeliğinin LRFD tasarım prensibi esas alınarak AISC tarafından belirlenmiş W profiller içerisinde uygun seçimler ile optimum tasarım araştırılmıştır. Çalışmada TLBO algoritması kullanılarak literatürdeki çeşitli çerçeve örnekleri incelenmiştir. TLBO algoritması ile bulunan sonuçlar Genetik

Algoritma, Karınca Kolonisi Optimizasyonu, Harmoni Arama Algoritması, Geliştirilmiş Karınca Kolonisi Optimizasyonu yöntemleri ile karşılaştırılarak güçlü bir algoritma olduğu ortaya konmuştur.

Dede vd. (2011) tarafından yapılan çalışmada, muhtelif çubuk eleman sayılarında düzlem kafes sistemler için, önceden oluşturulmuş listeden uygun kesitlerin seçtirilmesi ile minimum sistem ağırlığı aranmıştır. Çalışmada Genetik Algoritma kullanılmıştır. Düzlem kafeslerin optimizasyonu konusunda pek çok çalışma yapılmıştır. Bunlardan bazıları: Toğan vd. (2011) tarafından Harmoni arama algoritması ile, Kaveh ve Talatahari (2012) tarafından CBO (Colliding Bodies Optimisation) ve PSO (Parçacık Sürü Optimizasyonu)'nun hibriti ile, Dede ve Ayvaz (2013) tarafından ise TLBO algoritması ile gerçekleştirilmiştir.

Saka vd. (2015) tarafından yapılan çalışmada AISC yönetmeliği ile LRFD tasarım prensibi esas alınarak uzay çelik çerçeve optimizasyonu gerçekleştirilmiştir. Çalışmada; Karınca Kolonisi Optimizasyon Yöntemi, Parçacık Sürü Optimizasyonu, Yapay Arı Kolonisi Algoritması, Ateş Böceği Algoritması ve Guguk Kuşu Algoritması kullanılarak iki farklı uzay çerçeve optimize edilmiştir. Amaç fonksiyonu olarak sistemin toplam ağırlığı esas alınmıştır. Farklı algoritmalar ile elde edilmiş sonuçlar karşılaştırılarak, sürü zekasını temel alan algoritmaların yapısal optimizasyon problemlerinin çözülmesinde başarılı yöntemler olduğunun altı çizilmiştir.

Artar ve Daloğlu (2020) tarafından yapılan çalışmada, çelik uzay çerçevelerin optimum tasarımını yapı-zemin etkileşimi de dikkate alınarak gerçekleştirilmiştir. Normalde çok uzun sürede tamamlanan analizler, Jaya algoritması vasıtasıyla oldukça kısa sürelerde gerçekleştirilmiştir. Optimizasyon için MATLAB dilinde oluşturulmuş kod ile SAP2000 – OAPI ortak çalıştırılmıştır. Çalışmada Jaya Algoritmasının pratikliği ile sağladığı zaman tasarrufu olumlu bulunmuştur.

Yapısal optimizasyon literatürü incelendiğinde çalışmalardaki genel eğilimin sistemlerin ana hatları ile optimize edilmesi şeklinde olduğu görülmüştür. Bu bağlamda pek çok çalışmanın, profil listelerinden uygun profillerin seçtirilmesi gibi yöntemler

benimsenerek yapıldığı görülmektedir. Birleşim hesapları bazı kabuller ile sınırlı kalarak, bulonlu ya da kaynaklı birleşim detayları için optimizasyon yapılması yerine analizler genellikle yarı rijit kabul ile sürdürülmektedir. Yarı rijit birleşimler ilgili düğüm noktasına hem dönebilen hem de moment aktarabilen yaylar modellenerek sağlanmaktadır (Lui ve Lopes 1997).

Kaynaklı birleşimler ile ilgili literatür sınırlıdır. Bunun temel sebebi Northridge (1994) ve Kobe (1995) depremleri sırasında meydana gelen beklenmedik hasarlardan önce, çelik yapıların kaynaklı birleşimlerinin davranışları için yeterince araştırma bulunmamasıdır. Bu depremlerin neticesi olarak, kaynaklı birleşimlerin davranışlarını etkileyen parametreleri inceleyen çalışmalar, doksanlı yılların ikinci yarısından sonra özellikle ABD ve Japonya’da yoğunlaşmaktadır (Mele 2002). Bu çalışmalardan bazıları şunlardır: Mahin vd. (1996), Engelhardt ve Sabol, (1997), Tanaka vd. (1997) ve Castiglioni vd. (2000).

Kaynaklı birleşimlerin optimizasyonuna dair literatürde en çok bilinen problem Ragsdell ve Phillips (1976) tarafından ortaya atılmıştır. Problemden kurulum, kaynak işçiliği ve malzeme maliyetlerinin toplamından oluşan amaç fonksiyonunun minimize edilmesi hedeflenmiştir. Çalışmada ortaya atılan kaynaklı kiriş problemi, Geometrik Programlama yöntemi ile optimize edilmiştir. Bu problem diğer araştırmacılar tarafından da yaygın olarak kullanıldığından, yeni geliştirilen algoritmaların eski algoritmalar ile mukayesesi noktasında bir test fonksiyonu olarak kabul görmüştür.

Literatürdeki ilgili problemin çözüldüğü bazı çalışmalar şunlardır: Lee ve Geem (2005) tarafından Harmoni Arama Algoritması ile kaynaklı kiriş problemi optimize edilmiştir. Coello (2000) tarafından geliştirilen “kendi kendine uyarlanabilir ceza yaklaşımı” kaynaklı kiriş problemi üzerinde test edilmiştir. Coello (2008) tarafından kaynaklı kiriş problemini Parçacık Sürü Optimizasyonu ile çözülmüştür. David vd. (2016) tarafından yapılan bir çalışmada kaynaklı kiriş problemi pek çok farklı algoritma ile çözümler sonucularını mukayese edilmiştir.

Alberdi vd. (2015) tarafından elik erevelerin optimize edildiėi alıřmada, birleřimler iin drt farklı topoloji tanımlanıp ereve optimizasyonu ile birlikte birleřim topoloji optimizasyonu da gerekleřtirmiřtir. alıřmada Harmoni Arama, Karınca Kolonisi, Genetik Algoritma, Tabu Arama algoritmaları kullanılmıřtır.

Jin vd. (2017) tarafından yapılan alıřmada ereve ile birlikte kaynaklı birleřimlerin de optimizasyonunu saėlamak amacıyla bir optimizasyon prosedr nerilmiřtir. Maliyet hesabına dayanan bu yntem literatrdeki diėer maliyet hesapları ile kıyaslandığında olduka basit kalmaktadır. alıřmada optimizasyonu saėlamak amacıyla Genetik Algoritma kullanılmıřtır. Algoritmanın kontrol kısıtları Japon deprem ynetmeliklerinden oluřturulmuřtur.

3. MATERYAL ve METOT

3.1 Çalışmada Kullanılan Metasezgisel Algoritmalar

Bu bölümde çalışmada kullanılmış olan metasezgisel optimizasyon algoritmaları, örnekleri ile birlikte açıklanmıştır. Metasezgisel optimizasyon algoritmalarının genellikle belirli esin kaynakları bulunmaktadır. Bu bölümde açıklanacak Teaching and Learning Based Optimization (Öğrenme ve Öğretme Tabanlı Optimizasyon) (TLBO) algoritması insan davranışlarından esinlenirken, Sinüs Kosinüs Algoritması (SCA) matematikten esinlenmektedir. Jaya algoritmasının ise herhangi bir esin kaynağı bulunmamaktadır.

3.1.1 Jaya Algoritması

Rao (2016a) tarafından geliştirilmiştir. Jaya kelime olarak Sanskritçe’de zafer anlamına gelmektedir. Diğer pek çok metasezgisel algorithmadan farklı olarak herhangi bir doğa olayı, sosyal hayat, hayvan davranışları gibi bir esin kaynağı bulunmamaktadır. Algoritma, başarıya yaklaşımaya (en iyi çözüme ulaşmaya) ve başarısızlıktan kaçmaya (yani en kötü çözümden uzaklaşmaya) dayanan bir matematiksel felsefe ile çalışmaktadır. Algorithmaya özgü bir parametre içermemektedir. Bu durum algoritmanın aynı optimizasyon problemi özelinde tutarlı sonuçlar verebilmesi noktasında kararlılığını arttırmaktadır. Daha iyi bir performans ve sonuç için doğru parametreleri arama ihtiyacını ortadan kaldırdığından, parametrelili algorithmalara göre daha kullanışlı olduğu düşünülmektedir. Algoritma tek fazlı yapıdadır. Yani her iterasyon için popülasyondaki elemanlar bir defa güncellenmektedir. Algoritmanın adımları aşağıdaki gibidir:

Adım-1: Popülasyon eleman sayısı n , tasarım değişkeni sayısı d ve iterasyon sayısı i belirlenir. Popülasyon (X) her bir değişken için tanım kümeleri içerisinde rastlantısal olarak seçilir. Neticede n aday çözüm için n satırlı, d tasarım değişkeni sayısı için d sütunlu bir popülasyon elde edilir. Bu matrisin her bir satırı bir aday çözümü ifade ederken her bir sütun bir değişkendir.

Adım-2: Popülasyon amaç fonksiyonundan geçirilerek uygunluk değerleri olan “ f ” elde edilir. Her satır için bir “ f ” değeri bulunacağından bir sütun vektörü oluşacaktır.

Adım-3: Amaca yönelik en iyi çözüm (*best*), en kötü çözüm (*worst*) ve bu çözümleri veren elemanların koordinatları ($X_{j,best,i}$, ve $X_{j,worst,i}$) belirlenir.

Adım-4: Popülasyon (X) denklem (3.1) ile güncellenir.

$$X'_{j,k,i} = X_{j,k,i} + r_{1,j,i}(X_{j,best,i} - |X_{j,k,i}|) - r_{2,j,i}(X_{j,worst,i} - |X_{j,k,i}|) \quad (3.1)$$

Yukarıdaki denklemde,

$X_{j,k,i}$: i . iterasyonda k . çözümün j . değişkeni. (Güncellenecek değişken.)

$X'_{j,k,i}$: $X_{j,k,i}$ 'in denklem (3.1) ile güncellenmiş hali.

$r_{1,j,i}$: i . iterasyondaki j . Değişken için $[0,1]$ aralığındaki rastsal sayı.

$r_{2,j,i}$: i . iterasyondaki j . Değişken için $[0,1]$ aralığındaki rastsal sayı.

$X_{j,best,i}$: En iyi çözümü veren çözüm kümesinin j . değişkeni.

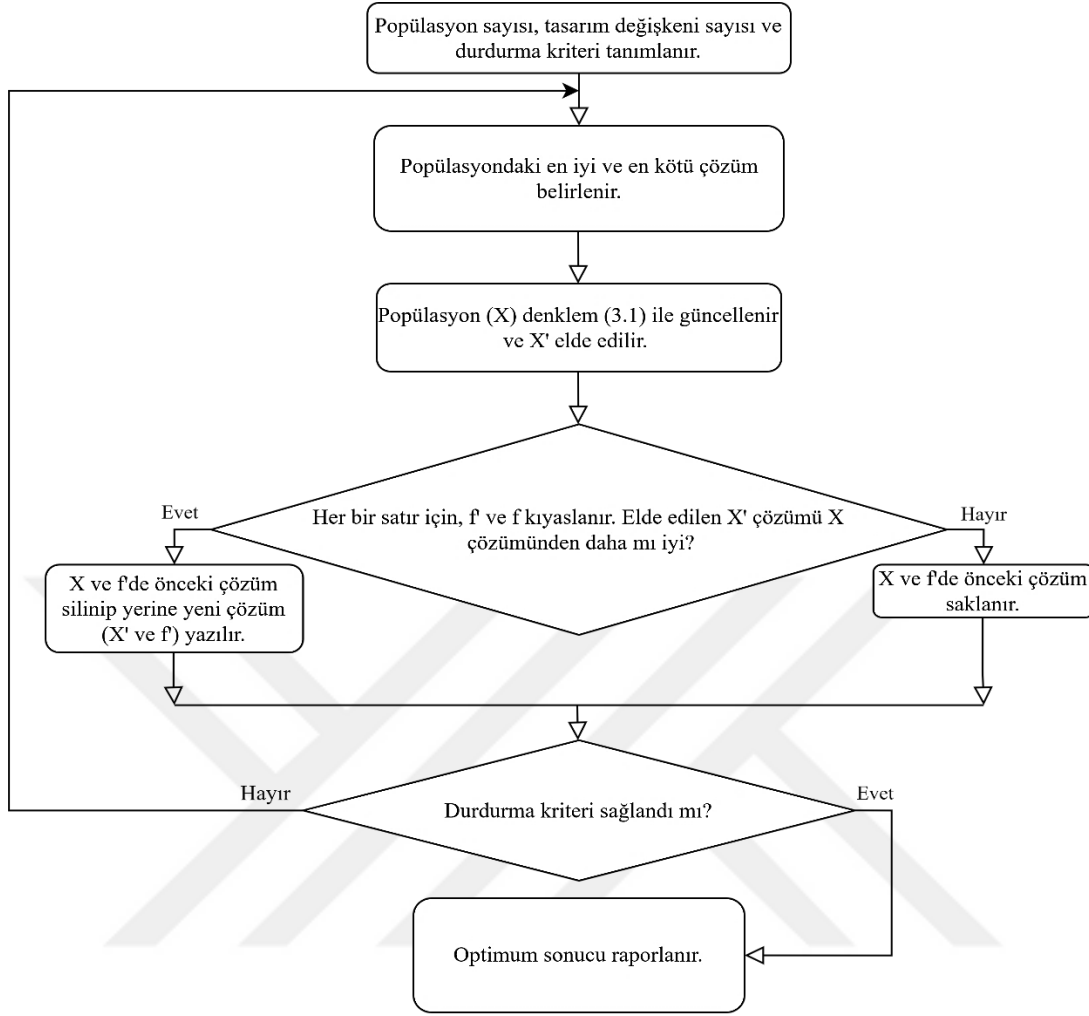
$X_{j,worst,i}$: En kötü çözümü veren çözüm kümesinin j . değişkeni.

Denklemdeki “ $r_{1,j,i}(X_{j,best,i} - |X_{j,k,i}|)$ ” kısmı algoritmanın en iyiyi örnek alarak ona doğru yaklaşma işlevini sağlarken “ $r_{2,j,i}(X_{j,worst,i} - |X_{j,k,i}|)$ ” kısmı ise kötüden uzaklaşma işlevini sağlamaktadır. Jaya algoritması sadece denklem (3.1) ile bütün işlevlerini gerçekleştirerek ilerlemektedir.

Adım-5: Güncellenen yeni aday çözümler matrisi (X'), amaç fonksiyonundan geçirilerek f' bulunur.

Adım-6: Her bir eleman için f ile f' kıyaslanarak eski çözüm ile yeni çözüm arasında aç gözlü seçim yapılır. Daha iyi olan sonuç ve sonucu veren çözüm kümesi saklanırken kötü çözüm ve çözüm kümesi silinir. f ve X bu şekilde güncellenir.

Adım-7: Tekrar Adım-3'e dönülerek durdurma kriterleri sağlanana kadar döngü devam ettirilir. Jaya algoritmasının akış diyagramı Şekil 3.1'de gösterilmiştir.



Şekil 3.1 Jaya algoritması için akış diyagramı.

Java algoritması için sayısal örnek:

$$\min f(x_i) \sum_{i=1}^d x_i^2 \quad (3.2)$$

Denklem (3.2)'de verilmiş olan minimizasyon problemi $-100 \leq x_i \leq 100$ tanım kümesi esas alınarak ve problemin boyutu (tasarım değişkeni sayısı) olan $d=2$ alınarak Jaya algoritması ile çözülmek istenirse:

Adım-1: Popülasyon her bir değişken için belirlenen sınırlara göre rastgele oluşturulmuştur. ($n=5$ seçilmiştir. d soruda 2 olarak verilmiştir. X matrisi 5×2 olacaktır.)

Adım-2: Popülasyon amaç fonksiyonundan geçirilerek $f(x)$ değerleri bulunmuştur.

Adım-3: En iyi (*best*) ve en kötü (*worst*) sonuç belirlenmiştir. İlk üç adımdaki işlemler Çizelge 3.1'de gösterilmiştir.

Çizelge 3.1 Jaya algoritması için ilk üç adımının sonuçları.

Aday	x_1	x_2	$f(x)$	Durum
1	-7	78	6133	
2	23	32	1553	
3	41	-67	6170	worst
4	-67	21	4930	
5	10	-3	109	best

Adım-4: Problemin boyutu kadar r_1 ve r_2 rastsal sayıları $[0,1]$ aralığında belirlenerek Çizelge 3.2’de gösterilmiştir. Popülasyon denklem (3.1)’den geçirilerek Çizelge 3.3’de verilen geçici popülasyon elde edilmiştir.

Çizelge 3.2 Denklem (3.2)’de kullanılacak r_1 ve r_2 rastsal sayıları.

	r_1	r_2
x_1 için	0,43	0,75
x_2 için	0,87	0,32

Adım-5: Güncellenen yeni aday çözümler matrisi (X'), amaç fonksiyonundan geçirilerek f' elde edilmiş ve Çizelge 3.3’de gösterilmiştir.

Çizelge 3.3 X' popülasyonu ve f' değerleri.

Aday	x'_1	x'_2	$f'(x)$
1	-31,21	53,93	3882,51
2	3,91	33,23	1119,52
3	27,67	-85,02	7994,03
4	-72,01	28,28	5985,20
5	-13,25	14,18	376,63

Adım-6: Her bir aday çözüm için f ile f' kıyaslanmıştır. Eski çözüm ile yeni çözüm arasında aç gözlü seçim yapılarak elde edilen yeni popülasyon Çizelge 3.4’de gösterilmiştir.

Çizelge 3.4 Güncellenmiş X popülasyonu ve f değerleri.

Aday	x_1	x_2	$f(x)$
1	-31,21	53,93	3882,509
2	3,91	33,23	1119,521
3	41	-67	6170
4	-67	21	4930
5	10	-3	109

Adım-7: Durdurma kriteri sağlanıyorsa optimum sonuç rapor edilecektir. Sağlanmıyorsa tekrar Adım-3'e dönülerek durdurma kriterleri sağlanana kadar döngü devam ettirilir.

3.1.2 Teaching and Learning Based Optimization (TLBO)

Teaching And Learning Based Optimization (Öğrenme-Öğretme Tabanlı Optimizasyon) (TLBO), bir sınıf içerisindeki öğrencilerin öğrenme ve öğretme sürecini taklit eden popülasyon tabanlı bir algoritmadır. Rao vd. (2011) tarafından geliştirilmiştir. Öğrenci grubu popülasyon olarak kabul edilmektedir. Öğrencilere sunulan farklı dersler optimizasyon probleminin farklı tasarım değişkenlerini ifade etmektedir. Bir öğrencinin derslerinden aldığı puanlar optimizasyon probleminin uygunluk değerine yani amaç fonksiyonundan geçirilmiş değerine benzetilmektedir. TLBO algoritmasının pek çok geliştirilmiş ya da elitleştirilmiş çeşidi bulunmaktadır. Temel TLBO algoritmasında popülasyonun en iyi çözüm değerine sahip elemanı öğretmen olarak kabul edilir. TLBO, öğretmen aşaması ve öğrenci aşaması olmak üzere iki ayrı faza sahiptir. İki fazlı yapısından dolayı bir iterasyon için her eleman iki defa güncellenmektedir. Bu durum daha az iterasyonla çözüme ulaşmayı sağlarken her bir iterasyon için geçen süreyi uzatmaktadır. Algoritmanın kendine özgü parametreleri bulunmamaktadır. Bu durum algoritmayı daha kolay kullanılabilir kılmaktadır.

Algoritmanın ilk aşaması olan öğretmen aşaması, sınıf içerisindeki öğrencilerin derslerini öğretmen vasıtasıyla öğrenme sürecini taklit etmektedir. Bu aşamada öğretmen, öğrettiği konudaki sınıfın ortalama sonucunu kendi kapasitesine bağlı olarak arttırmaya çalışır. Standart TLBO algoritmasında, sınıfın en iyi elemanı öğretmen olarak kabul edildiği için aslında popülasyonun en başarılı bireyinin diğer bireyleri kendisine yaklaştırması söz konusudur. TLBO algoritmasının öğretmen aşamasının adımları aşağıda açıklanmıştır:

Adım-1: Popülasyon eleman sayısı n , tasarım değişkeni sayısı d ve iterasyon sayısı i belirlenir. Popülasyon (X) her bir değişken için tanım kümeleri içerisinde rastlantısal olarak oluşturulur. Neticede n öğrenci için n satırlı, d ders sayısı için d sütunlu bir popülasyon elde edilir. Bu matrisin her bir satırı yani her bir öğrenci bir aday çözümü ifade ederken her bir sütun yani her bir ders bir değişkendir.

Adım-2: Tanımlanan popülasyon amaç fonksiyonundan geçirilerek “ f ” uygunluk değerleri elde edilir. Her bir öğrenci için puan hesabı yapılır. Her bir satır için yani her bir öğrenci için bir f değeri oluşacağından bir sütun vektörü elde edilecektir.

Adım-3: Sınıfın bütün derslerdeki ortalaması alınır.

Adım-4: En iyi uygunluk değerine sahip olan birey öğretmen olarak kabul edilir.

Adım-5: Öğretmen aşamasında popülasyonun güncellenmesi denklem (3.3) ile gerçekleşmektedir.

$$X'_{j,k,i} = X_{j,k,i} + Difference_Mean_{j,k,i} \quad (3.3)$$

Bu denklemde,

i : iterasyon

m : öğretilen konu sayısı (Tasarım değişkeni sayısı, problemin boyutu.)

n : Öğrenci sayısı (popülasyon büyüklüğü. $k=1,2,\dots,n$)

$X_{j,k,i}$: i . iterasyonda k . çözümün j . değişkeni. TLBO algoritması için değişken k . öğrencinin aldığı j . derse karşılık gelmektedir. ($j=1,2,\dots,m$)

$X'_{j,k,i}$: Popülasyondaki her bir $X_{j,k,i}$ değerinin denklem (3.3) ile güncellenmiş hali.

$Difference_Mean_{j,k,i}$: Her ders için mevcut ortalama sonuç ile her ders için öğretmenin sonucu arasındaki farkı ifade eder. Denklem (3.4) ile hesaplanmaktadır.

$$Difference_Mean_{j,k,i} = r_i * (X_{j,kbest,i} - T_F * M_{j,i}) \quad (3.4)$$

Bu denklemde,

r_i : $[0,1]$ aralığında rastgele sayı. (Her sütun yani konu için bir r_i üretilir.)

$X_{j,kbest,i}$: Uygunluk değeri en iyi olan öğrencinin yani öğretmenin j dersindeki sonucu.

$M_{j,i}$: Belirli bir derste (tasarım değişkeninde) öğrencilerin sınıf ortalaması.

T_F : Öğretim faktörü. Bu değerin 1 ya da 2 olarak alınması tavsiye edilmektedir. Birçok kıyaslama işlevi üzerinde yapılan denemeler neticesinde T_F 'in 1 ya da 2 seçilmesi durumunda algoritma performansının optimum seviyeye ulaştığı tespit edilmiştir (Rao 2016b). T_F , TLBO algoritması için bir parametre değildir. Denklem (3.5) vasıtasıyla eşit olasılıkla 1 ya da 2 olarak algoritma tarafından seçilmektedir.

$$T_F = \text{round}[1 + \text{rand}(0,1)\{2 - 1\}] \quad (3.5)$$

Denklem (3.3) neticesinde elde edilen geçici popülasyon (X') amaç fonksiyonundan geçirilerek uygunluk değerleri (f') tespit edilir.

Adım-6: Geçici popülasyonun uygunluk değerleri (f') ile popülasyonun uygunluk değerleri (f) kıyaslanarak aç gözlü seçim mekanizması işletilir. Bunun neticesinde her bir öğrenci için (her bir satır için) daha uygun olan çözüm kalırken uygunluğu düşük olan çözüm silinir. Öğretmen aşaması bu adımda tamamlanarak güncellenmiş popülasyon (X) ve uygunluk değerleri (f) öğrenci aşamasında kullanılmak üzere kaydedilir.

Algoritmanın ikinci aşaması, öğrencilerin kendi aralarındaki etkileşim sayesinde birbirlerinden öğrendikleri süreci taklit eden öğrenci aşamasıdır. Bu aşamada bir öğrenci, bilgisini arttırmak için diğer öğrencilerle rastgele etkileşime girmektedir. Etkileşime girdiği öğrenci kendisinden daha bilgiliyse yeni bilgiler öğrenir. TLBO algoritmasının öğrenci aşamasının adımları aşağıda açıklanmıştır:

Adım-1: Öğrencilerin birbirine öğretme mekanizması denklem (3.6), (3.7), (3.8) ve (3.9) ile matematiksel olarak modellenmiştir. Aşağıdaki denklemler vasıtasıyla öğretmen aşamasının sonucunda elde edilen popülasyon tekrar güncellenir.

$$X''_{j,P,i} = X'_{j,P,i} + r_i(X'_{j,P,i} - X'_{j,Q,i}) \quad ; \quad X'_{total-P,i} < X'_{total-Q,i} \quad (3.6)$$

$$X''_{j,P,i} = X'_{j,P,i} + r_i(X'_{j,Q,i} - X'_{j,P,i}) \quad ; \quad X'_{total-Q,i} < X'_{total-P,i} \quad (3.7)$$

$$X''_{j,P,i} = X'_{j,P,i} + r_i(X'_{j,P,i} - X'_{j,Q,i}) \quad ; \quad X'_{total-Q,i} < X'_{total-P,i} \quad (3.8)$$

$$X''_{j,P,i} = X'_{j,P,i} + r_i(X'_{j,Q,i} - X'_{j,P,i}) \quad ; \quad X'_{total-P,i} < X'_{total-Q,i} \quad (3.9)$$

Yukarıdaki denklemlerde, P ve Q rastgele seçilmiş iki öğrencidir. Burada $X'_{total-P,i}$ ve $X'_{total-Q,i}$ değerleri $X_{total-P,i}$ ve $X_{total-Q,i}$ 'in yani öğrenci matrisindeki popülasyonun öğretmen aşamasında güncellenmiş versiyonunun fonksiyondan geçmiş halidir.

$X'_{total-P,i} \neq X'_{total-Q,i}$ şartı sağlanmalıdır. Yani öğrenci kendisiyle ya da kendisine eşit bir öğrenciyle eşleşmemelidir. Denklem (3.6) ve (3.7) minimizasyon için, (3.8) ve (3.9) maksimizasyon için kullanılır.

Adım-2: Öğrencilerin birbirine öğretme mekanizması neticesinde elde edilen yeni değerlerin uygunluk fonksiyonu bulunarak öğretmen aşamasının sonundaki değerler ile kıyaslanır. Aç gözlü seçim yapılarak daha iyi olan çözüm saklanır. Bu sayede TLBO algoritmasının öğrenci aşaması da tamamlanır. Sonuç olarak bir iterasyon tamamlanmış olur. Bir iterasyonun iki fazında popülasyon iki defa güncellenmiştir. Sonraki iterasyon için tekrar öğretmen aşamasına dönülerek işlem sürdürülür. TLBO algoritması için akış şeması Şekil 3.2’de gösterilmiştir.

TLBO algoritması için sayısal örnek:

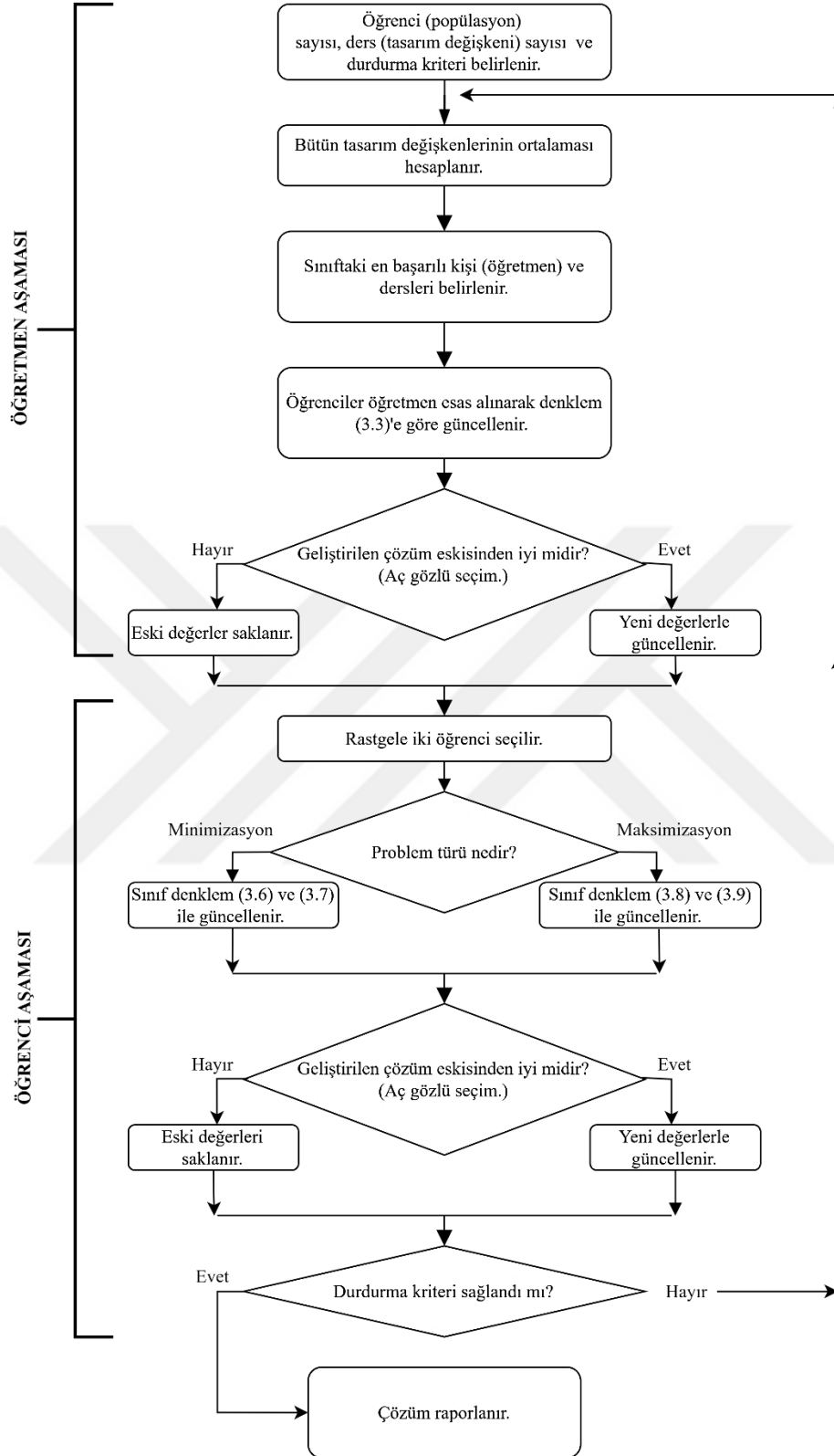
Denklem (3.2)’de verilmiş olan minimizasyon problemi $-100 \leq x_i \leq 100$ tanım kümesi esas alınarak ve tasarım değişkeni sayısı $d=2$ için TLBO algoritması ile bir iterasyon için çözülmek istenirse:

Öğretmen aşaması için:

Adım-1: Popülasyon boyutu $n=5$, Tasarım değişkeni sayısı $d=2$, iterasyon sayısı (durdurma kriteri) $t=1$ olarak belirlenmiştir. Sınıfı oluşturan matris $n \times d$ yani 5×2 boyutunda tanım kümesi içerisinde rastlantısal olarak oluşturulmuştur.

Adım-2: Tanımlanan popülasyon (X) amaç fonksiyonundan geçirilerek uygunluk değerleri (f) elde edilmiştir.

Adım-3: Oluşturulan sınıfta (popülasyonda) her bir dersin (tasarım değişkeni) sınıf ortalamaları belirlenmiştir.



Şekil 3.2 TLBO algoritması için akış şeması.

Adım-4: En iyi uygunluk değerine sahip olan birey öğretmen olarak kabul edilmiştir. Bu adıma kadar yapılan işlemler Çizelge 3.5’de gösterilmiştir.

Çizelge 3.5 TLBO algoritması için ilk dört adım.

Öğrenci	x_1	x_2	$f(x)$	
1	21	-4	457	
2	41	13	1850	
3	-62	3	3853	
4	91	-72	13465	
5	12	-4	160	Öğretmen
Ortalama (M)	20,6	-12,8		

Adım-5: Sınıf, denklem (3.3) vasıtası ile güncellenip geçici popülasyon elde edilmiştir. Bunun için öncelikle *Difference_Mean* değeri denklem (3.4) ile T_F değeri denklem (3.5) ile bulunmuştur. Bu adımdaki işlemler Çizelge 3.6 ve 3.7’de gösterilmiştir.

Çizelge 3.6 T_F ve *Difference_Mean* değerleri.

	r_1	r_2	T_F	<i>Difference_Mean</i>
x_1	0,35	-	1	-3,01
x_2	-	0,73		6,42

Çizelge 3.7 Denklem (3.3) ile elde edilen geçici popülasyon.

Öğrenci	x'_1	x'_2	$f'(x)$
1	17,99	2,42	329,50
2	37,99	19,42	1820,38
3	-65,01	9,42	4315,04
4	87,99	-65,58	12042,98
5	8,99	2,42	86,68

Adım-6: Çizelge 3.5’de verilen sınıfın ilk hali ile denklem (3.3) ile güncellenerek elde edilmiş geçici hali kıyaslanarak aç gözlü seçim yapılmıştır. Bu adımın sonunda öğretmen aşaması tamamlanmıştır. Aç gözlü seçim neticesinde elde edilen sınıf Çizelge 3.8’de gösterilmiştir. Çizelge 3.8’de gösterilen değerler öğrenci aşamasında girdi olarak kullanılacaktır.

Çizelge 3.8 Öğretmen aşamasının çıktısı olarak elde edilmiş nihai popülasyon.

Öğrenci	x_1	x_2	$f(x)$
1	17,99	2,42	329,4965
2	37,99	19,42	1820,377
3	-62	3	3853
4	87,99	-65,58	12042,98
5	8,99	2,42	86,6765

Öğrenci aşaması için:

Adım-1: Problem minimizasyon problemi olduğundan dolayı denklem (3.6) ve (3.7) kullanılarak geçici popülasyon elde edilmiştir. Her öğrenci etkileşime katılmak zorundadır. Etkileşilecek öğrenci rastlantısal olarak belirlenir. Bu örnek için 1 ile 2, 2 ile 3, 3 ile 5, 4 ile 1 ve 5 ile 4 etkileşime sokulmuştur. Etkileşim hesabı için gerekli olan x_1 için $r_1=0,85$, x_2 için $r_2=0,94$ ilgili aralıkta rastsal olarak belirlenmiştir. Adım-1 için sonuçlar Çizelge 3.9’da verilmiştir.

Çizelge 3.9 Öğrenci aşaması için güncellenmiş geçici popülasyon.

Öğrenci	Etkileşim	Bilgi Aktarımı	x''_1	x''_2	$f'(x)$
1	1-2	1→2	0,99	-13,56	184,85
2	2-3	2→3	122,98	34,85	16339,31
3	3-5	5→3	-1,66	2,45	8,78
4	4-1	1→4	28,49	-1,66	814,44
5	5-4	5→4	-58,16	66,34	7783,58
			$r_1=0,85$	$r_2=0,94$	

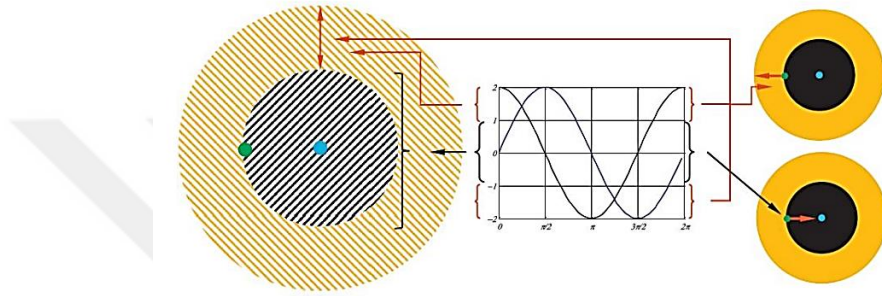
Adım-2: Çizelge 3.9’da verilen öğrenci aşaması için güncellenmiş geçici popülasyon ile önceki popülasyon olan Çizelge 3.8’de verilen popülasyon arasında aç gözlü seçim yapılır. Her bir eleman için iyi olan değer tutulur, kötü olan değer atılır. Aç gözlü seçim neticesinde elde edilmiş popülasyon Çizelge 3.10’da gösterilmiştir. Bu işlem ile TLBO algoritması için bir iterasyon tamamlanmış olur. Birinci iterasyonun sonucunda elde edilen en iyi fonksiyon değeri $f(x)=8,78$ ’dir. Durdurma kriteri sağlanıyorsa elde edilen optimum sonuç raporlanır, kriter sağlanmıyorsa yeniden öğretmen aşamasına gidilerek döngü sürdürülür.

Çizelge 3.10 Birinci iterasyonun sonucunda elde edilen değerler.

Öğrenci	x_1	x_2	$f(x)$
1	0,99	-13,56	184,85
2	37,99	19,42	1820,38
3	-1,66	2,45	8,78
4	28,49	-1,66	814,44
5	8,99	2,42	86,68

3.1.3 Sinüs-Kosinüs Algoritması (SCA)

Sinüs – kosinüs algoritması Mirjalili (2015) tarafından geliştirilen, matematik tabanlı bir metasezgisel algoritmadır. SCA’da diğer sürü tabanlı algoritmalar gibi keşif (global arama) ve sömürü (keşfedilmiş global optimumun etrafında derinlemesine lokal arama) olmak üzere iki ana ilke etrafında çalışmaktadır. Algoritmanın arama mekanizması Şekil 3.3’de gösterilmiştir.



Şekil 3.3 SCA için arama modeli (Mirjalili 2015).

Bu çalışmada kullanılan diğer algoritmaların aksine SCA, çalışmak için kendine özgü bazı parametrelere ihtiyaç duymaktadır. Bu durum doğru parametrelerin seçilebilmesi için bir dizi deneme ve inceleme ihtiyacını ortaya çıkarmaktadır. Ayrıca algoritmanın optimizasyon problemlerinin çözümü için, tek başına kullanılabilmesini zorlaştırdığı düşünülmektedir. SCA bu dezavantajına karşın oldukça kısa süren işlem adımları ile pratik ve hızlı bir algoritma olarak tercih sebebi olabilmektedir. Algoritmanın adımları aşağıdaki gibidir:

Adım-1: Popülasyon eleman sayısı (n), tasarım değişkeni sayısı (d) ve durdurma kriteri olarak iterasyon sayısı (t) belirlenir. Her bir eleman için tasarım değişkenleri tanım kümeleri içerisinde rastgele seçilerek popülasyon oluşturulur. Neticede ortaya n satırlı ve d sütunlu bir popülasyon matrisi çıkar.

Adım-2: Her bir eleman amaç fonksiyonundan geçirilerek uygunluk değerleri elde edilir.

Adım-3: Denklem (3.10) ile popülasyon güncellenerek geçici popülasyon elde edilir.

$$X_i^{t+1} = \begin{cases} x_i^t + r_1 * \sin(r_2) * |r_3 * P_i^t - X_i^t| & , \quad r_4 < 0,5 \\ x_i^t + r_1 * \cos(r_2) * |r_3 * P_i^t - X_i^t| & , \quad r_4 \geq 0,5 \end{cases} \quad (3.10)$$

Bu denklemde,

P_i^t : Elde edilen en iyi çözümün ilgili tasarım değişkeni.

r_2 : İçe ya da dışa doğru hareket miktarını belirler. $([0, 2\pi]$ aralığında radyan cinsinden rastgele açı ölçüsü.)

r_3 : Stokastik ağırlık katsayısı. $r_3 > 1$ ise stokastik önemli $r_3 < 1$ ise stokastik daha etkisizdir. $r_3 = b * [0,1]$ olarak tanımlandığında b değeri SCA için bir çalışma parametresi olarak ortaya çıkmaktadır. Rastlantısallığa eşit oranda şans vermek için r_3 parametresi $[0,2]$ aralığında rastgele sayı olarak seçilebilir.

r_4 : Denklemdeki sin-cos arası geçişi sağlayan parametredir. Sinüs ve kosinüs kullanımına eşit şans vermek için $[0,1]$ aralığında rastgele sayı olarak seçilebilir.

r_1 : Bir sonraki pozisyon bölgesi veya hareket yönünü belirler. Keşif ve sömürü arasındaki dengeyi sağlamak için kullanılmaktadır yani lokal ekstremum noktalarında takılmayı önler. Denklem (3.11) kullanılarak sinüs ve kosinüs aralığı adaptif olarak değişime uğratılır.

$$r_1 = a - t \frac{a}{T} \quad (3.11)$$

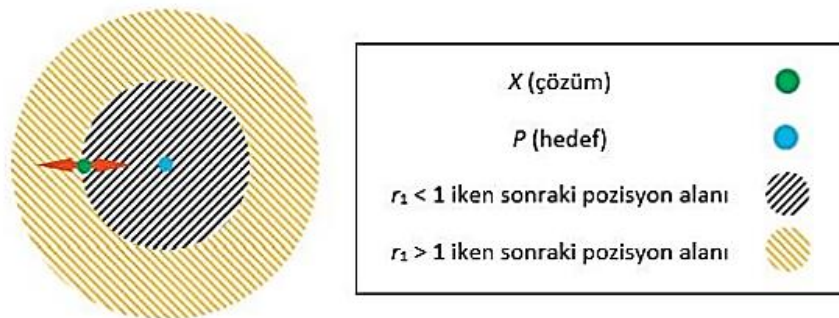
Denklem (3.11)'de,

t : içinde bulunulan iterasyon numarası

T : Toplam iterasyon sayısı

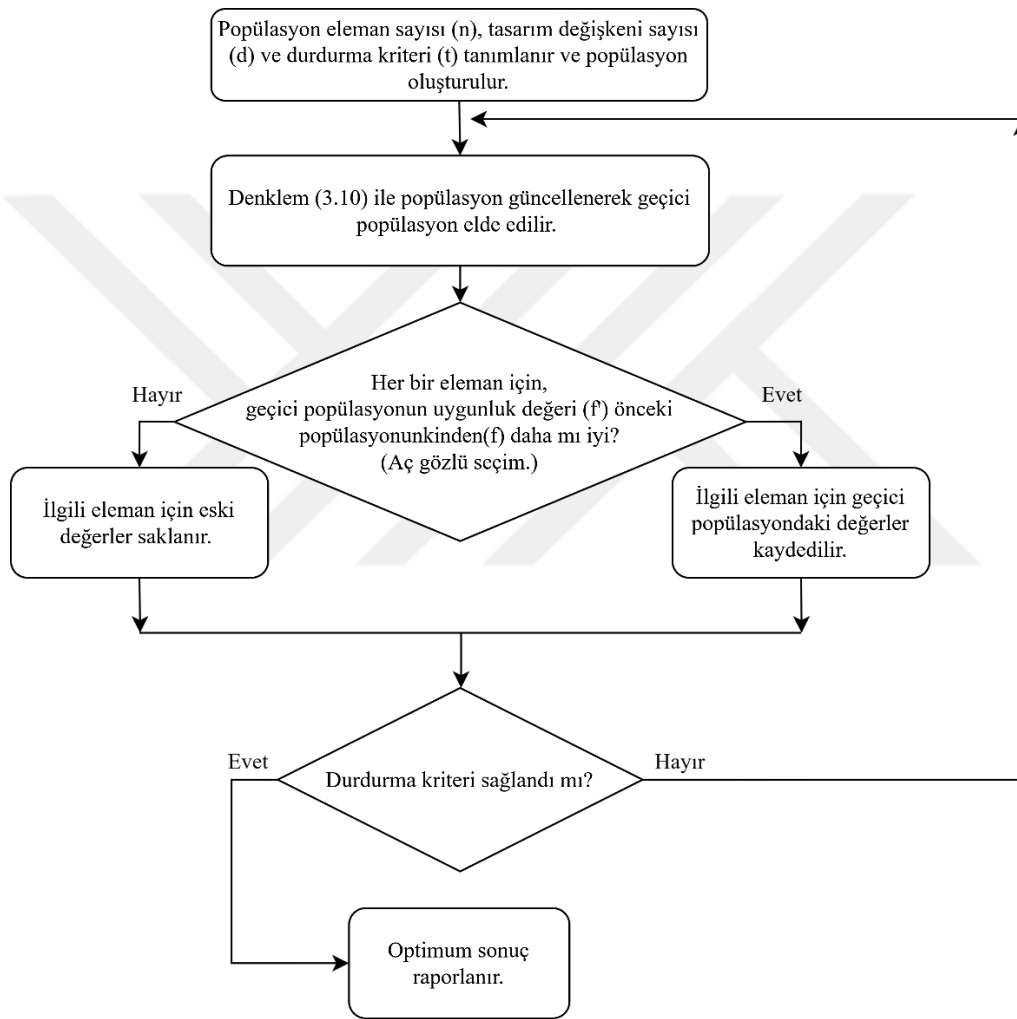
a : Algoritma parametresidir. Sinüs ve kosinüsün arama genliğindeki değişimle ilişkilidir.

r_1 parametresi her iterasyon için güncellenirken r_2, r_3, r_4 , parametreleri hem her iterasyon için hem de her bir tasarım değişkeni için güncellenmektedir. r_1 'in çözüm üretmeye olan etkisi Şekil 3.4'de gösterilmiştir.



Şekil 3.4 r_1 'in çözüm üretmedeki etkisi (Mirjalili 2015).

Adım-4: Denklem (3.10) ile elde edilen geçici popülasyon ile önceki popülasyon arasında uygunluk değerleri kıyaslanarak aç gözlü seçim yapılır. Her bir eleman için daha iyi olan değerler tutulurken daha kötü değerler silinir. Bu adımla birlikte SCA için bir iterasyon tamamlanmış olur. Durdurma kriteri sağlanıyorsa optimum sonuç raporlanır. Kriter sağlanmıyorsa Adım-3'e gidilerek iterasyon sürdürülür. SCA için akış diyagramı Şekil 3.5'de verilmiştir.



Şekil 3.5 SCA için akış diyagramı.

SCA için sayısal örnek:

Denklem (3.2)'de verilmiş olan minimizasyon problemi $-100 \leq x_i \leq 100$ tanım kümesi esas alınarak ve tasarım değişkeni sayısı $d=2$ için SCA ile çözülmek istenirse:

Adım-1: Popülasyon eleman sayısı $n=5$, tasarım değişkeni sayısı $d=2$, iterasyon sayısı $t=2$ olarak seçilmiştir. Her bir eleman için tasarım değişkenleri tanım kümeleri içerisinde rastgele seçilerek popülasyon oluşturulmuştur. Bunun neticesinde 5×2 boyutunda popülasyon matrisi ortaya çıkmıştır.

Adım-2: Her bir eleman amaç fonksiyonundan geçirilerek uygunluk değerleri elde edilmiştir. En iyi çözüm sonraki adımda kullanılmak üzere belirlenmiştir. İlk iki adım Çizelge 3.11’de gösterilmiştir.

Çizelge 3.11 SCA’nın ilk iki adımının sonuçları.

Aday	x_1	x_2	$f(x)$	
1	-18	81	6885	
2	19	38	1805	
3	-14	53	3005	
4	10	-71	5141	
5	19	-23	890	P

Adım-3: Denklem (3.10) kullanılarak popülasyon güncellenerek geçici popülasyon elde edilmiştir. Elde edilen çözüm Çizelge 3.12’de gösterilmiştir.

Çizelge 3.12 Denklem (3.10) ile elde edilen geçici popülasyon.

Aday	a	t	T	r_1	r_2 ($\pi \text{ rad}$)	r_3	r_4	x'_1	x'_2	$f'(x)$
1					0,43	1,78	0,33	-12,60	93,71	8939,99
2					1,43	1,12	0,78	19,08	40,24	1983,05
3	0,5	1	2	0,25	0,74	0,33	0,46	-10,58	63,21	4107,98
4					1,76	0,47	0,16	10,26	-56,22	3266,13
5					0,12	1,97	0,97	23,57	-17,46	860,69

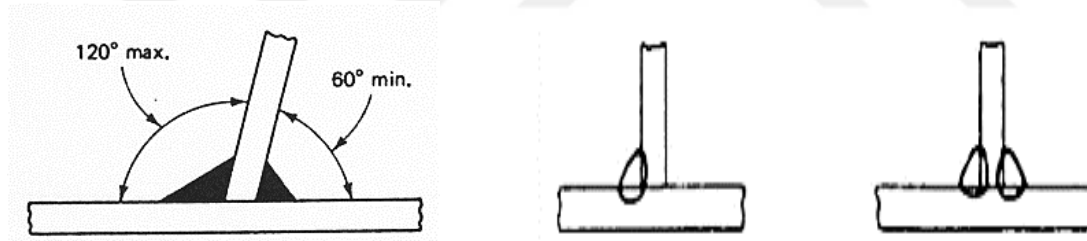
Adım-4: Çizelge 3.11’de verilen popülasyonun uygunluk değerleri ile Çizelge 3.12’de verilen geçici popülasyonun uygunluk değerleri kıyaslanarak aç gözlü seçim yapılır. Daha iyi olan sonuç saklanırken daha kötü sonuç silinir. Aç gözlü seçim neticesinde elde edilen yeni popülasyon Çizelge 3.13’de gösterilmiştir. Elde edilen değerler aynı zamanda birinci iterasyonun sonucudur. Durdurma kriteri sağlanıyorsa optimum sonuç raporlanır. Kriter sağlanmıyorsa Çizelge 3.13’de gösterilen değerler sonraki iterasyon için girdi olarak kullanılırlar.

Çizelge 3.13 SCA için birinci iterasyonun sonuçları.

Aday	x_1	x_2	$f(x)$	
1	-18,00	81,00	6885,00	
2	19,00	38,00	1805,00	
3	-14,00	53,00	3005,00	
4	10,26	-56,22	3266,13	
5	23,57	-17,46	860,69	P

3.2 ÇYTHYE 2018'e Göre Köşe Kaynaklı Birleşimlerin Tasarımı

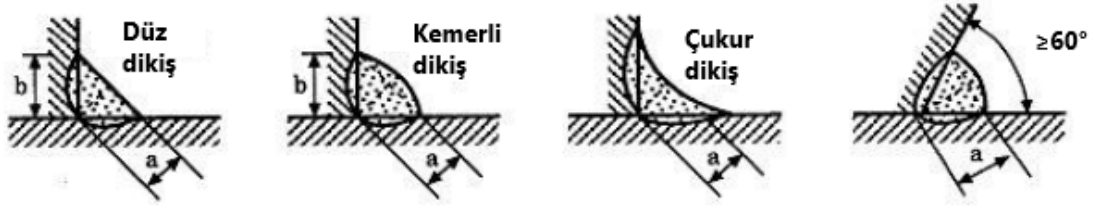
Çelik Yapıların Tasarım, Hesap ve Yapım Esaslarına Dair Yönetmelik (2018) (ÇYTHYE 2018)'e göre birleşen çelik elemanlar arasındaki açı 60°'den büyük 120°'den küçük olan kaynaklar köşe kaynak olarak değerlendirilerek tasarlanacaktır. Tipik köşe kaynak uygulamaları Şekil 3.6'da gösterilmiştir. Birleşim açısının 60°'den küçük olması durumunda ilgili kaynak kısmi penetrasyonlu küt kaynak olarak düşünülecektir. Köşe kaynak uzunluğu hesaplanırken krater kayıpları dikkate alınmalıdır. Krater kaybı olarak her bir kaynak ucu için bir kaynak kalınlığı kadar kaynak boyu azaltılmalıdır (ÇYTHYE 2018).



Şekil 3.6 Tipik köşe kaynak uygulamaları (Yorgun vd. 2017).

3.2.1 Etkin Alan

Köşe kaynakların etkin alanları kaynak etkin uzunluğu ile kaynak etkin kalınlığının çarpılması ile elde edilmektedir. Kaynak etkin kalınlığı (a), kaynak en kesiti içerisine çizilen hayali üçgenin yüksekliğine yani kaynak kökünden kaynak yüzeyine alınabilecek en kısa mesafeye eşittir. Köşe kaynak tipine bağlı olarak kaynak kalınlıkları Şekil 3.7'de gösterilmiştir (ÇYTHYE 2018).



Şekil 3.7 Köşe kaynaklarının etkin kalınlıkları (Gaylord vd. 1992).

3.2.2 Sınırlamalar

Köşe kaynaklarda belirlenecek minimum etkin kalınlık, hesaplanmış olan kuvvetin emniyetli şekilde aktarılmasına yetecek minimum kalınlıktan ve Çizelge 3.14’de verilen minimum kalınlıklardan daha az seçilemez. Kısmi veya tam penetrasyonlu küt kaynakları desteklemek amacıyla yapılan köşe kaynaklar bu sınırlamalara tabi değildir (ÇYTHYE 2018).

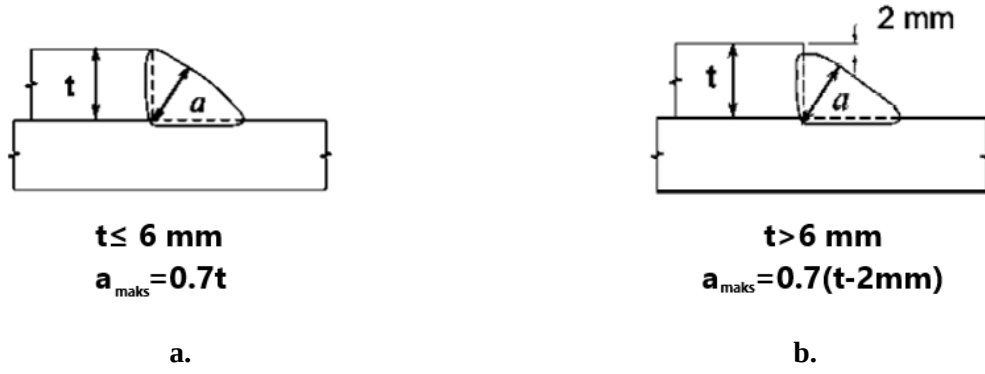
Çizelge 3.14 Köşe kaynakların minimum kalınlıkları (ÇYTHYE 2018).

Bileşen İnce Elemanın Kalınlığı, t (mm)	Minimum Köşe Kaynak Kalınlığı, ^a (mm)
$t \leq 6$	3,0
$6 < t \leq 19$	3,5
$13 < t \leq 19$	4,0
$19 < t$	5,5

^a:Tek geçişli kaynaklar kullanılmalıdır.

Maksimum kaynak kalınlığı için ise, kaynak ile birleştirilen elemanın kenar kalınlığı, t olmak üzere, aşağıdaki şartlara uyacaktır:

- Kaynaklanan ince elemanın kenar kalınlığı olan t , 6 mm’den küçük ise kaynak kalınlığı a , $0,7t$ ’den daha büyük seçilemez (Şekil 3.8a).
- Kaynaklanan ince elemanın kenar kalınlığı olan t , 6 mm’den büyük ise kaynak kalınlığı a , $0,7(t - 2 \text{ mm})$ ’den daha büyük seçilemez (Şekil 3.8b) (ÇYTHYE 2018).



Şekil 3.8 Maksimum kaynak kalınlıkları **a.** $t < 6 \text{ mm}$ **b.** $t > 6 \text{ mm}$ (Yorgun vd. 2017).

Köşe kaynakları için minimum etkin uzunluk, kaynak kalınlığı olan a 'nın 6 katından veya 40 mm'den daha küçük seçilmemelidir. Bu koşula uyulmayan durumlarda kaynak etkin kalınlığı, kaynak uzunluğunun 1/6'sı olarak kabul edilecektir. Lamadan yapılmış çelik çekme elemanı için uç birleşim de sadece boyuna doğrultuya köşe kaynaklar kullanılırsa, bu kaynakların uzunluğu kaynaklar arasındaki dik uzunluktan daha az olmamalıdır (Şekil 3.9).

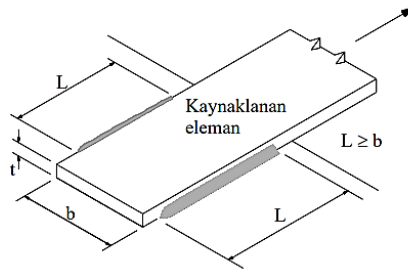
Çelik elemanların kaynaklı uç birleşimlerindeki etkin kaynak uzunluğu aşağıdaki şartlara göz önüne alınarak denklem (3.12a), (3.12b) ve (3.12c) ile hesaplanmaktadır.

$$L \leq 150a \text{ için} \quad L_e = L \quad (3.12a)$$

$$150a < L \leq \text{ için} \quad L_e = \beta L \quad (3.12b)$$

$$\beta = 1.2 - 0,0014(L/a) \leq 1.0$$

$$400a < L \text{ için} \quad L_e = 250a \quad (3.12c)$$



Şekil 3.9 Yalnızca boyuna kaynaklar ile yapılmış uç birleşim (ÇYTHYE 2018).

Bu denklemlerde,

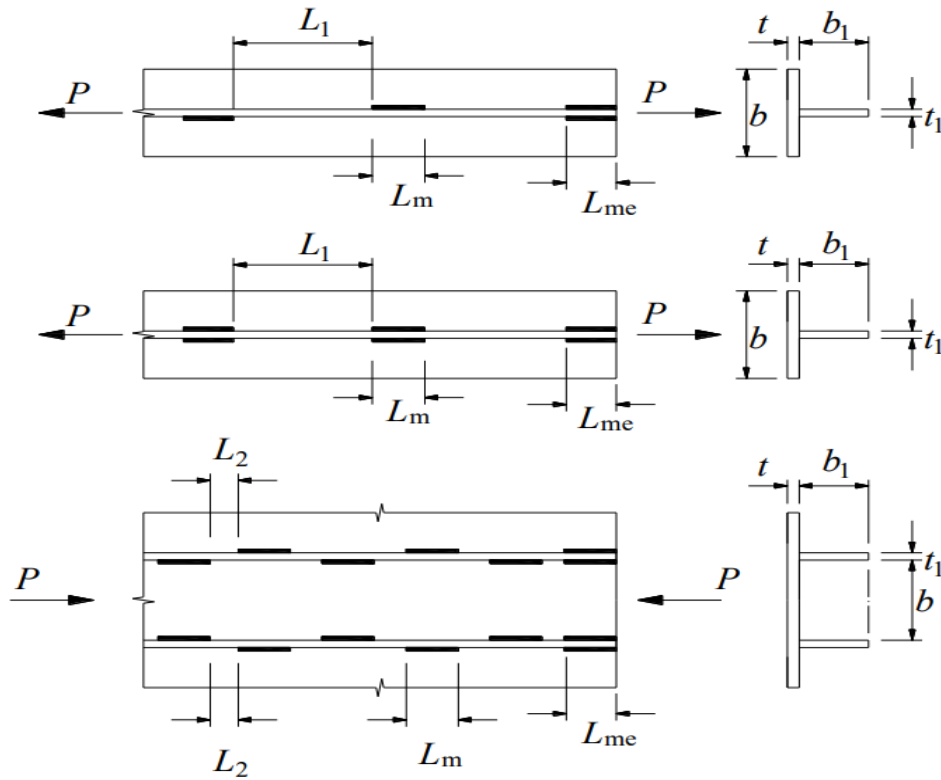
L : Kaynağın uzunluğu.

L_e : Etkin kaynak uzunluğu.

a : Kaynak etkin kalınlığı (Kaynak enkesiti içerisinde çizilen hayali üçgenin yüksekliği).

β : Azaltma katsayısıdır.

Düğüm noktalarında birleşen yüzeyler boyunca ve levhalar kullanılarak oluşturulan yapma elemanların enkesit parçalarının birleşimlerinde, belirli aralıklarla düzenlenen süreksiz köşe kaynakların (metot kaynağı) kullanılmasına izin verilebilir, (Şekil 3.10). Bu durumda her bir köşe kaynak parçasının uzunluğu, L_m , kaynak kalınlığının 6 katından ve 40mm'den az olamaz.



$$L_{me} \geq \text{Maksimum } (b ; b_1)$$

Çekme kuvveti etkisindeki yapma enkesitler için

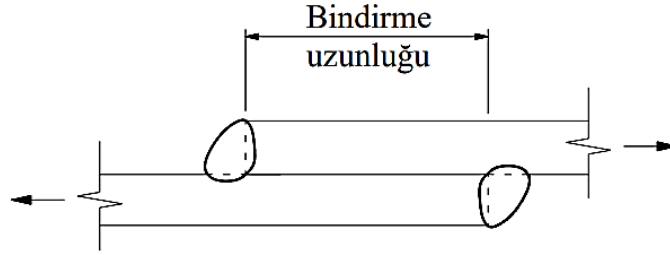
$$L_1 \leq \text{Minimum } (14t ; 14t_1 ; 200\text{mm})$$

Basınç kuvveti veya kesme kuvveti etkisindeki yapma enkesitler için

$$L_2 \leq \text{Minimum } (12t ; 12t_1 ; 0.25b ; 200\text{mm})$$

Şekil 3.10 Belirli aralıklarla düzenlenen süreksiz köşe kaynakların uygulama sınırları (ÇYTHYE 2018).

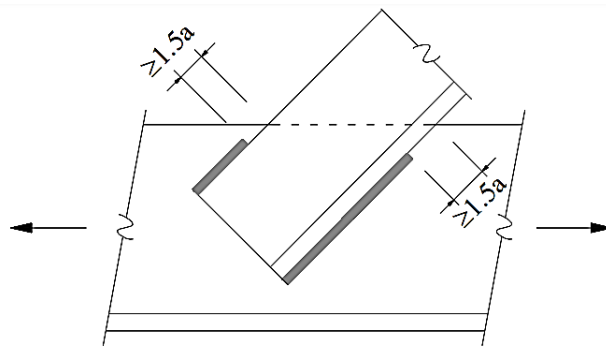
Bindirmeli birleşimlerde, bindirme uzunluğu, birleşen elemanların ince olanının kalınlığının 5 katından ve 25 mm'den küçük olamaz. Sadece enine doğrultuda (kuvvet eksenine dik) köşe kaynakların kullanılacağı aksel çekme kuvveti etkisindeki levhaların ve lamaların bindirmeli birleşimlerinde her iki bindirme ucu da enine doğrultuda köşe kaynakla bağlanacaktır, (Şekil 3.11).



Şekil 3.11 Minimum bindirme uzunluğu (ÇYTHYE 2018).

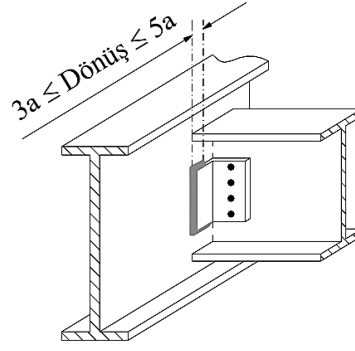
Köşe kaynağın, kaynaklanan kenarın uç noktasından önce sonlandırılmasına, diğer kenar boyunca uzatılmasına veya çepeçevre uygulanmasına, aşağıda tanımlanan hususlar göz önünde tutularak izin verilebilir.

- Çekme kuvveti etkisindeki bir eleman üzerine bağlanan başka bir elemanın uç birleşimi için kullanılacak köşe kaynaklar, çekme etkisindeki eleman kenarından kaynak kalınlığının en az 1,5 katı kadar geride sonlandırılacaktır, (Şekil 3.12).



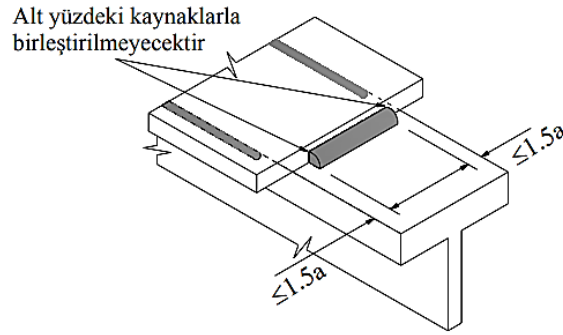
Şekil 3.12 Çekme etkisindeki bir elemana köşe kaynaklı bağlantı (ÇYTHYE 2018).

- Mafsallı bağlantıların teşkilinde kullanılan elemanların, uç dönüşleri yapılarak uygulanan köşe kaynaklarının dönüş uzunlukları, kaynak kalınlığının 5 katını ve kenar uzunluğun yarısını aşamaz, (Şekil 3.13).



Şekil 3.13 Kiriş ucu ile birleşim elemanı arasında dönme uyumunun sağlanabilmesi için kaynak uygulaması (ÇYTHYE 2018).

- c. Rijitlik levhalarının kiriş başlıklarına bağlandığı durumlar hariç olmak üzere, enine rijitlik levhalarını 20mm ve daha ince kalınlıklı gövde levhasına bağlayan köşe kaynakların sonlandırıldığı noktanın, gövde – başlık kaynağına uzaklığı, gövde kalınlığının 4 katından az, 6 katından fazla olamaz.
- d. Aynı düzlemin farklı yüzeylerine uygulanan köşe kaynaklar, eleman köşe bölgelerine yakın noktalarda birbirleriyle birleştirilmeyecek şekilde sonlandırılmalıdır, (Şekil 3.14).



Şekil 3.14 Bir düzlemin farklı yüzeylerindeki köşe kaynaklar için uygulama detayı (ÇYTHYE 2018).

Kesme kuvvetini aktarmak, bindirmeli bağlantılarda birleşen yüzeye dik etkilere karşı dayanım sağlamak, burkulma veya bindirme bölgesinden ayrılmaları önlemek amacıyla ve yapma elemanların enkesit parçalarının bağlantıları için oluşturulan dairesel veya oval deliklerde köşe kaynakların kullanılmasına izin verilir. Bu tür köşe kaynaklar, dairesel ve oval dolgu kaynakları kapsamında değerlendirilemez (ÇYTHYE 2018).

3.2.3 Kaynaklı Birleşimlerin Dayanımı

Kaynaklı birleşim noktalarının tasarım dayanımları (ϕR_n) veya güvenli dayanımları (R_n/Ω) esas metalin çekme ve kayna etkisindeki kırılma sınır durumları ve kaynak metalinin kırılma sınır durumları hesaplanarak kritik olan yani en küçük kapasiteye sahip değer esas alınarak yapılacaktır.

Esas metalin karakteristik dayanımı olan R_{nBM} ve kaynak metalinin karakteristik dayanımı olan R_{nw} sırasıyla denklem (3.13) ve (3.14) ile hesaplanacaktır.

$$R_{nBM} = F_{nBM} A_{BM} \quad (3.13)$$

$$R_{nw} = F_{nw} A_{we} \quad (3.14)$$

Bu denklemlerde,

F_{nBM} : Esas metal için karakteristik gerilme

F_{nw} : Kaynak metali için karakteristik gerilme

A_{BM} : Kaynak boyunca esas metalin yüzey alanı

A_{we} : Etkin kaynak alanı

ϕ , Ω , F_{nBM} ve F_{nw} için alınacak değerler ve kullanılacak yöntemler Çizelge 3.15’de verilmiştir.

Çizelge 3.15 Köşe kaynakların ϕ , Ω , F_{nBM} ve F_{nw} değerleri ve ÇYTHYE yönetmeliği içerisinde esas alınacak bölümler (ÇYTHYE 2018).

KÖŞE KAYNAKLAR					
Yük Tipi ve Kaynak Eksenine Göre Doğrultusu	İlgili Metal	ϕ ve Ω	Karakteristik Gerilme (F_{nBM} veya F_{nw})	Etkin Alan (A_{BM} veya A_{wt})	Kaynak Metali Gerekli Dayanım Düzeyi
Kesme	Esas Metal	Bölüm 13.4			
	Kaynak Metali	ϕ =0,75 Ω =2,00	0,60 F_E	Bölüm 13.2.2.1	Bölüm 13.2.6
Kaynak eksenine paralel çekme ve basınç	Kaynak eksenine paralel çekme ve basınç etkilerinin kaynak tasarımında göz önüne alınmasına gerek yoktur.				

Köşe kaynakların karakteristik gerilmesi olan F_{nw} kaynağın boyuna ekseninin kuvvetin doğrultusuyla arasındaki açı dikkate alınmadan Çizelge 3.15’den kullanılabilir. İlgili açı dikkate alındığında ise kaynakların dayanımları için (tasarım dayanımları ϕR_n veya güvenli dayanımları R_n/Ω) $\phi = 0,75$ (YDKT) veya $\Omega = 2,00$ (GKT) alınarak aşağıdaki gibi hesaplanabilir:

- a) Ağırlık merkezinden geçen eksenel yüke maruz, birbirine paralel konumda ya da aynı eksenin üzerinde üniform kalınlıklı köşe kaynak grubu için karakteristik dayanım denklem (3.15) ve denklem (3.16) vasıtasıyla hesaplanmaktadır.

$$R_{nw} = F_{nw}A_{we} \quad (3.15)$$

$$F_{nw} = 0,60F_E(1,0 + 0,50\sin^{1,5}\theta) \quad (3.16)$$

- b) Ağırlık merkezinden geçen eksenel yüke maruz, eksenine paralel ve eksenine dik yük etkisindeki köşe kaynaklardan oluşan üniform kalınlıklı kaynak grubunun karakteristik dayanımı denklem (3.17) ve denklem (3.18) ile hesaplanan değerlerin büyüğüne eşit kabul edilecektir.

$$R_{nw} = R_{nw1}R_{nwt} \quad (3.17)$$

$$R_{nw} = 0,85R_{nw1} + 1,5R_{nwt} \quad (3.18)$$

Bu denklemlerde,

F_E : Kaynak metali için karakteristik çekme dayanımı.

θ : Derece cinsinden yük doğrultuş ve kaynak boyuna eksenini arasındaki açı.

R_{nw1} : Eksenine paralel yüke maruz köşe kaynakların toplam karakteristik dayanımı.

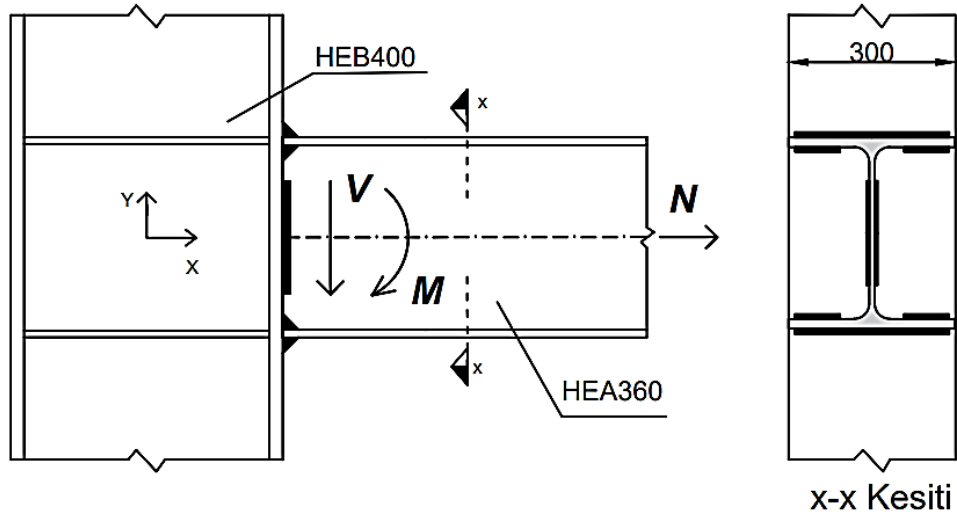
R_{nwt} : Eksenine dik yüke maruz köşe kaynakların toplam karakteristik dayanımı.

R_{nw1} ve R_{nwt} köşe kaynakların boyuna eksenleri ile kuvvet doğrultusundaki açı dikkate alınmadan hesaplanan toplam karakteristik dayanımı ifade eder (ÇYTHYE 2018)

3.3 Problemlerin Tanıtımı ve Optimizasyon Modellerinin Oluşturulması

3.3.1 Giriş

Bu tez çalışmasında temel birleşim modeli olarak HEA 360 profilden teşkil edilmiş kirişin HEB400 profilden teşkil edilmiş kolona köşe kaynak ile bağlantısı seçilmiştir. İlgili modelde, HEA 360 profil; başlık dış, başlık iç ve gövde kaynak grupları ile kolona köşe kaynaklar vasıtasıyla bağlanmaktadır. Her bir kaynak grubu için kaynak kalınlıkları (a_x) ve kaynak uzunlukları (L_x) eşit kabul edilmiştir. Bunun sebebi; kaynak gruplarındaki asimetrik ölçülerin, kaynak düzleminde eksantrisite oluşturmaları neticesinde arama uzayında optimumdan uzaklaşacak bölgeler yaratmasıdır. Bu durumun optimizasyon algoritmalarının performansını olumsuz etkilememesi için kaynak grupları simetrik olarak boyutlandırılmıştır. Birleşim; moment (M), kesme kuvveti (V) ve normal kuvvet (N) etkisi altındadır. Seçilen birleşim modelinin genel hali Şekil 3.15’de gösterildiği gibidir.



Şekil 3.15 Birleşim modelinin temel görünümü.

Çelik malzeme sınıfı olarak bütün profiller için S275 tercih edilmiştir. Kaynak metalinin esas metalden daha mukavim olma şartını sağlamak üzere kaynak metali için $F_E=480$ N/mm² seçilmiştir. Sayısal modellerde kullanılacak bütün yükler sabit yük olarak kabul edilmiştir.

3.3.2 Sayısal Modeller

Giriş kısmında temel hali tanıtılan birleşim modelinden, farklı yükleme durumları için üç adet sayısal model (SM) oluşturulmuştur. Birleşim modeli, yük ve dayanım katsayıları ile tasarım (YDKT) ve güvenlik katsayıları ile tasarım (GKT) prensipleri için; moment etkin, normal kuvvet etkin ve kesme kuvveti etkin yükleme durumları altında çözülecektir. Sayısal modeller için isimlendirme Çizelge 3.16’da gösterilmiştir.

Çizelge 3.16 Tasarım esasları ve yükleme durumlarına göre sayısal modeller.

Tasarım Prensibi	Yükleme Durumları		
	Moment Etkin	Normal Kuvvet Etkin	Kesme Kuvveti Etkin
GKT	SM-1A	SM-2A	SM-3A
YDKT	SM-1B	SM-2B	SM-3B

Sayısal modeller için farklı yükleme durumlarını temsil eden yük dağılımları Çizelge 3.17’de gösterilmiştir. Bu yükleme durumları, Şekil 3.15’de verilen profiller ile alakalıdır. Yük ve momentler birleşimin kapasitesini mümkün olduğunca yüksek oranda kullanacak şekilde seçilmiştir.

Çizelge 3.17 Yükleme durumları için kuvvet dağılımları.

Yükleme Durumu	M (kNm)	N (kN)	V (kN)
Moment Etkin	80	10	10
Normal Kuvvet Etkin	10	220	10
Kesme Kuvveti Etkin	10	10	100

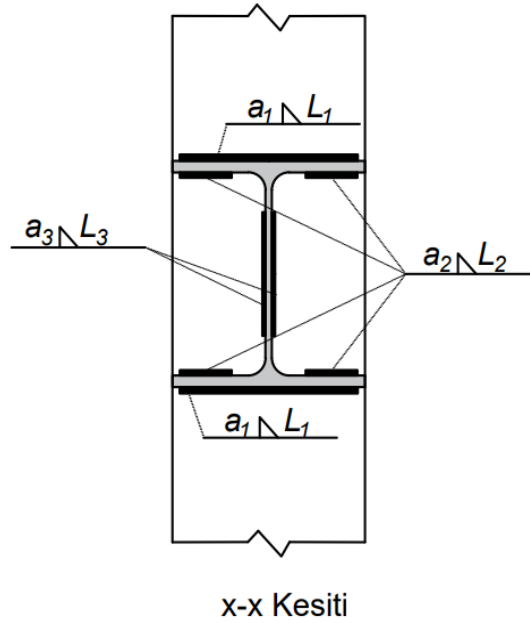
3.3.3 Tasarım Değişkenleri

Tüm sayısal modellerde üç adet kaynak grubu bulunmaktadır. Bunlar; başlık dış kaynak grubu, başlık iç kaynak grubu ve gövde kaynak grubudur. Optimizasyon algoritmalarının verimsiz bölgelerde arama yapmasının önüne geçilmesi amacıyla her bir kaynak grubu için eşit kaynak kalınlığı ve eşit kaynak uzunluğu seçilmiştir. Yani her şartta kaynak alanları x ve y eksenine göre simetriktr. Bu sayede kaynak düzlemindeki eksantrisitenin önüne geçilmiştir. Bunun sonucu olarak her bir kaynak grubu için bir kaynak kalınlığı ve bir kaynak uzunluğu olmak üzere toplamda altı adet tasarım değişkeni ortaya çıkmıştır. İlgili tasarım değişkenleri Çizelge 3.18’de açıklanmıştır.

Çizelge 3.18 Tasarım değişkenleri.

Tasarım Değişkeni	Açıklama
a_1	Başlık dış kaynak grubu için kaynak kalınlığı.
a_2	Başlık iç kaynak grubu için kaynak kalınlığı.
a_3	Gövde kaynak grubu için kaynak kalınlığı.
L_1	Başlık dış kaynak grubu için kaynak uzunluğu.
L_2	Başlık iç kaynak grubu için kaynak uzunluğu.
L_3	Gövde kaynak grubu için kaynak uzunluğu.

Çizelge 3.18’de açıklanan tasarım değişkenleri şematik olarak Şekil 3.16’da gösterilmiştir.



Şekil 3.16 Tasarım değişkenleri.

3.3.4 Popülasyonun Yapısı

İncelenecek optimizasyon problemleri 6 adet tasarım değişkenine sahip olduğundan ($d=6$ boyutlu) oluşturulacak popülasyon matrisi 6 sütuna sahiptir. Matrisin satır sayısını algoritma parametresi olarak girilen popülasyon eleman sayısı (n) belirleyecektir. Örneğin, popülasyon matrisinin $n=3$ için genel görünümü denklem (3.19)’da gösterildiği gibidir.

$$X = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & L_{1,1} & L_{1,2} & L_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} & L_{2,1} & L_{2,2} & L_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} & L_{3,1} & L_{3,2} & L_{3,3} \end{bmatrix} \quad (3.19)$$

3.3.5 Amaç Fonksiyonu

Bu tez çalışmasında yapılan optimizasyonun nihai amacı minimum kaynak alanları ile maksimum faydanın sağlanmasıdır. Bu maksatla, yönetmelikteki kısıt ve gerekliliklere uyarak kaynaklardaki kullanım kapasitesinin %100'e yaklaştırılması amaçlanmıştır. İlgili birleşim modeli için kaynak alanının minimizasyonu, kaynak kapasite kullanımının maksimizasyonu anlamına gelmektedir. Buradan hareketle, sayısal modellerin hepsinde bütün kısıt ve gerekliliklere uyan minimum kaynak alanı aranmaktadır. Dolayısıyla amaç fonksiyonu olarak toplam kaynak alanı seçilmiştir. Amaç fonksiyonu kısıtsız durum için denklem (3.20)'de gösterildiği gibidir.

$$\min F_{(n)} = 2a_1L_1 + 4a_2L_2 + 2a_3L_3 \quad (3.20)$$

Amaç fonksiyonu denklem (3.20)'de gösterildiği gibi seçilmiş olsa da kısıtsız optimizasyon yöntemi yönetmelik ve tasarım esaslarından doğan gerekliliklerin sağlatılması noktasında yetersizdir. İlgili şartların sağlayan sonuçların elde edilebilmesi için kısıtlı optimizasyon yöntemi kullanılmıştır. Kısıtlı optimizasyon yöntemi için amaç fonksiyonu, kısıtsız optimizasyon için verilmiş olan amaç fonksiyonuna ceza fonksiyonu olan $p_{(n)}$ 'nin eklenmesi ile elde edilmiştir. $p_{(n)}$ vasıtasıyla kısıt ya da kısıtlara uymayan aday çözümler cezalandırılarak uygunluk değerleri cazip olmaktan uzaklaştırılmaktadır. İlgili amaç fonksiyon denklem (3.21)'de gösterilmiştir.

$$\min F_{(n)} = (2a_1L_1 + 4a_2L_2 + 2a_3L_3) + p_{(n)} \quad (3.21)$$

3.3.6 Kısıtlar

Kısıtlar, yönetmeliklerce ortaya konmuş kuralların optimizasyon algoritmasına tanıtılması ve uygulatılması amacıyla kontrol elemanı olarak kullanılmıştır. Kısıt kontrolleri oluşturulurken temel bilgisayar mantığından gelen “1” ve “0” kodlama

yönteminden yararlanılmıştır. Her bir aday çözüm ve her bir kısıt için, ilgili kısıta uyuluyorsa “0” uyulmuyorsa “1” değeri kısıt kontrol vektöründe saklanmaktadır. İlgili yöntem denklem (3.22)’de gösterilmiştir. Her bir kısıtın kontrolü için bir sütun vektörü oluşturulmuştur. Bu kısıt kontrol vektörleri ceza fonksiyonu olan $p_{(n)}$ ’in oluşturulması için kullanılacaktır. Kısıtları değerlendirmek ve ceza vermek için pek çok farklı yöntem ortaya atılmıştır. Bu çalışmada özgün bir yöntem olan “adil ceza yöntemi” uygulanacaktır.

Her bir kısıt için:

$$g_{k(n)} = 0 ; \text{ ilgili kısıta uyuluyorsa} \quad (3.22a)$$

$$g_{k(n)} = 1 ; \text{ ilgili kısıta uyulmuyorsa} \quad (3.22b)$$

Bu denklemde,

$g_{k(n)}$: Kısıt kontrol vektörü (kısıtlar için kısıta uymama sayısı yani hata sayacı).

k : kısıt numarası.

n : Popülasyon eleman numarasıdır.

Optimize edilecek sayısal modeller GKT ve YDKT tasarım prensipleri için ayrı iki süreç olarak kodlanmıştır. İki tasarım esasları farklı esaslara dayansa da hesaplarındaki benzerlik nedeniyle bu bölümde tek başlık altında açıklanmıştır. Güvenlik katsayıları ile tasarım ile yük ve dayanım katsayılarına göre tasarım prensiplerinde kullanılan kısıtlara ait semboller sırasıyla Çizelge 3.19 ve 3.20’de açıklanmıştır. Kaynak kalınlığı kontrolü kısıtı ile etkin kaynak uzunluğu kontrolü kısıtı kısıtları her iki tasarım prensibinde ortak kullanılmaktadır. Her bir tasarım prensibi için 6’şar adet kısıtla kontrol yapılmıştır.

Çizelge 3.19 GKT tasarım prensibi için kısıtlar.

Sembol	Kısıt Açıklaması
g_1	Kaynak kalınlığı kontrolü
g_2	Etkin kaynak uzunluğu kontrolü
g_3	GKT için başlık dış kaynak grubuna gelen gerilmelerin kontrolü
g_4	GKT için gövde üst ucuna gelen gerilmelerin kontrolü
g_5	GKT için gövde kaynak grubuna gelen kesme gerilmelerinin kontrolü
g_6	GKT için gövde kaynak grubuna gelen bileşke gerilmelerin kontrolü

Çizelge 3.20 YDKT tasarım prensibi için kısıtlar.

Sembol	Kısıt Açıklaması
g_1	Kaynak kalınlığı kontrolü
g_2	Etkin kaynak uzunluğu kontrolü
g_7	YDKT için başlık dış kaynak grubuna gelen gerilmelerin kontrolü
g_8	YDKT için gövde üst ucuna gelen gerilmelerin kontrolü
g_9	YDKT için gövde kaynak grubuna gelen kesme gerilmelerinin kontrolü
g_{10}	YDKT için gövde kaynak grubuna gelen bileşke gerilmelerin kontrolü

3.3.6.1 Kaynak Kalınlığı Kısıt Kontrolü

Kaynak kalınlığının kontrol edilebilmesi için yönetmelikte belirlenen minimum ve maksimum kaynak kalınlıkları belirlenmiştir. Minimum kaynak kalınlığı için Çizelge 3.14’de verilen sınırlar esas alınmıştır. Maksimum kaynak kalınlıkları Bölüm 3.2.2’de verilmiş olan ifadeler ile hesaplanmıştır. Kaynak kalınlığı kontrolünde esas alınan “ t_{min} ” birleşen elemanlardan ince olanın kalınlığını ifade etmektedir. Buradan hareketle başlık ve gövde için farklı kalınlıklar söz konusu olduğundan başlık için ayrı, gövde için ayrı kaynak kalınlığı kontrolü yapılmıştır. Yapılan kontroller denklem (3.23) ve (3.24)’te gösterilmiştir.

$$a_{n,kgn} > a_{max} \parallel a_{n,kgn} < a_{min} \Rightarrow g_{1(n,kgn)}^{dm_başlık} = 1 \quad (3.23a)$$

$$a_{min} < a_{n,kgn} < a_{max} \Rightarrow g_{1(n,kgn)}^{dm_başlık} = 0 \quad (3.23b)$$

$$a_{n,kgn} > a_{max} \parallel a_{n,kgn} < a_{min} \Rightarrow g_{1(n,kgn)}^{dm_gövde} = 1 \quad (3.24a)$$

$$a_{min} < a_{n,kgn} < a_{max} \Rightarrow g_{1(n,kgn)}^{dm_gövde} = 0 \quad (3.24b)$$

Gövde ve başlık için yapılan kaynak kalınlığı kontrolleri ilgili matrisler toplanarak birleştirilmiş ve g^{dm}_1 kısıt kontrol matrisi elde edilmiştir (Denklem 3.25). Bu matrisin her bir satırı toplanarak g_1 kısıt kontrol vektörüne ulaşılmıştır. İlgili işlem denklem (3.26)’da gösterildiği gibidir. Bu sayede kaynak kalınlığı kısıtı tek bir sütun vektörü ile kayıt altına alınmıştır.

$$g_{1(n,kgn)}^{dm} = g_{1(n,kgn)}^{dm_{başlık}} + g_{1(n,kgn)}^{dm_{gövde}} \quad (3.25)$$

$$g_{1(n)} = \sum_{x=1}^{kgn_{max}} (g_{1(n,x)}^{dm}) \quad (3.26)$$

Bu denklemlerde,

n : Popülasyon eleman numarası.

kgn : Kaynak grup numarası. (Başlık dış, başlık iç, gövde için sırasıyla $kgn=1$, $kgn=2$, $kgn=3$)

a : Kaynak kalınlığı.

$g_{1(n)}^{dm}$: Her bir kaynak için kaynak kalınlığı kısıt değerlendirilmelerini saklayan geçici matris.

g_1 : Popülasyonun her bir elemanı için kaynak kalınlığı kısıtına uyup uymama durumunu kaydeden kalıcı kısıt kontrol vektörü.

3.3.6.2 Etkin Kaynak Uzunluğu Kısıt Kontrolü

Etkin kaynak uzunluklarının hesaplanmasında Bölüm 3.2.2’de verilmiş olan esaslar kullanılmıştır. Minimum ve maksimum etkin kaynak uzunlukları belirlenerek hesaplanan etkin kaynak uzunluklarının bu arada olup olmadığı değerlendirilmiştir. İşlem her bir kaynak grubu (başlık dış, başlık iç, gövde) için yapılarak $(n \times 3)$ boyutundaki g_2^d matrisi elde edilmiştir. Bu kontrol denklem (3.27) ve denklem (3.28)’de gösterilmiştir.

$$Le_{n,kgn} > Le_{max} \parallel Le_{n,kgn} < Le_{min} \Rightarrow g_{2(n,kgn)}^{dm} = 1 \quad (3.27)$$

$$Le_{min} < Le_{n,kgn} < Le_{max} \Rightarrow g_{2(n,kgn)}^{dm} = 0 \quad (3.28)$$

g_2^d matrisinin her bir satırındaki elemanları toplanarak g_2 kısıt kontrol vektörüne ulaşılmıştır. Bu işlem denklem (3.29)’de gösterilmiştir.

$$g_2(n) = \sum_{x=1}^{kgn_{max}} (g_2^{dm}(n,x)) \quad (3.29)$$

Bu denklemlerde,

n : Popülasyon eleman numarası.

kgn : Kaynak grup numarası. (Başlık dış, başlık iç, gövde için sırasıyla $kgn=1$, $kgn=2$, $kgn=3$)

Le : Etkin kaynak uzunluğu.

g_2^{dm} : Her bir kaynak için etkin kaynak uzunluğu kısıt değerlendirmelerini saklayan geçici matris.

g_2 : Popülasyonun her bir elemanı için etkin kaynak uzunluğu kısıtına uyup uymama durumunu kaydeden kalıcı kısıt kontrol vektörü.

3.3.6.3 Başlık Dış Kaynak Grubunun Dış Ucuna Gelen Gerilmelerin Kısıt Kontrolleri (g_3 ve g_7)

Kaynak dayanımlarının belirlenmesinde Bölüm 3.2.3’de verilen esasalar uygulanmıştır. Köşe kaynakların güvenli dayanım gerilmesi denklem (3.30) ile tasarım dayanım gerilmesi denklem (3.31) ile hesaplanmıştır.

$$F_{emw_GKT} = 0,6F_e/\Omega \quad (3.30)$$

$$F_{emw_YDKT} = \phi 0,6F_e \quad (3.31)$$

Başlık dış kaynak grubuna gelen gerilme f_{x1} denklem (3.32) ile hesaplanmıştır.

$$f_{x1} = \frac{M}{I_w} * \frac{d}{2} + \frac{N}{A_{we}} \quad (3.32)$$

Bu denklemlerde,

F_{emw_GKT} : Köşe kaynakların güvenli dayanım gerilmesi.

F_{emw_YDKT} : Köşe kaynakların tasarım dayanım gerilmesi.

F_e : Kaynak metalinin karakteristik çekme dayanımı.

Ω : Güvenlik katsayısı.

ϕ : Dayanım katsayısı.

I_w : Toplam etkin kaynak dikişi alanının ve kaynak dikişlerinin kirişin kuvvetli eksenine göre toplam atalet momenti.

d : Kirişin yüksekliği

M : Moment.

N : Normal kuvvet.

A_{we} : Etkin kaynak alanı.

GKT için ilgili kaynak kapasitesinin kullanım oranı (GS_{g_3}) denklem (3.33) ile YDKT için kaynak kapasitesinin kullanım oranı (GS_{g_7}) denklem (3.34) ile hesaplanmıştır.

$$GS_{g_3} = f_{x1} / F_{emw_GKT} \quad (3.33)$$

$$GS_{g_7} = f_{x1} / F_{emw_YDKT} \quad (3.34)$$

g_3 ve g_7 için kısıta uygunluk kontrolü denklem (3.35) ve (3.36)'de gösterildiği gibi yapılmıştır.

$$GS_{g_3} < 1 \Rightarrow g_3(n) = 0 \quad (3.35)$$

$$GS_{g_3} \geq 1 \Rightarrow g_3(n) = 1$$

$$GS_{g_7} < 1 \Rightarrow g_7(n) = 0 \quad (3.36)$$

$$GS_{g_7} \geq 1 \Rightarrow g_7(n) = 1$$

3.3.6.4 Gövde Kaynak Grubunun Üst Ucuna Gelen Gerilmelerin Kısıt Kontrolleri (g_4 ve g_8)

Gövde kaynak dikişlerinin üst uçlarına gelen gerilmeler denklem (3.37)'de gösterildiği gibi bulunmuştur.

$$f_{x2} = \frac{M}{I_w} * \frac{Le_3}{2} + \frac{N}{A_{we}} \quad (3.37)$$

Bulunan gerilmenin kaynak kapasitesini kullanım oranı GKT için denklem (3.38) ve YDKT için (3.39)'de gösterildiği gibi bulunmuştur.

$$GS_{g_4} = f_{x2} / F_{emw_GKT} \quad (3.38)$$

$$GS_{g_8} = f_{x2} / F_{emw_YDKT} \quad (3.39)$$

g_4 ve g_8 için kısıta uygunluk kontrolü denklem (3.40) ve (3.41)'de gösterildiği gibi yapılmıştır.

$$GS_{g_4} < 1 \Rightarrow g_4(n) = 0 \quad (3.40)$$

$$GS_{g_4} \geq 1 \Rightarrow g_4(n) = 1$$

$$GS_{g_8} < 1 \Rightarrow g_8(n) = 0 \quad (3.41)$$

$$GS_{g_8} \geq 1 \Rightarrow g_8(n) = 1$$

3.3.6.5 Gövde Kaynak Grubunun Kesme Kısıt Kontrolleri (g_5 ve g_9)

Gövde kaynaklarının üst ucuna gelen kesmeden dolayı oluşan gerilmeler f_{y2} denklem (3.42) ile hesaplanmıştır.

$$f_{y2} = \frac{V}{A_{we_3}} \quad (3.42)$$

Denklem 3.42 ile bulunan gerilmelerin kaynak kapasitesini kullanım oranı GKT için (GS_{g_5}) denklem (3.43) ve YDKT için (GS_{g_9}) denklem (3.44)'deki gibi hesaplanmıştır.

$$GS_{g_5} = f_{y2} / F_{emw_GKT} \quad (3.43)$$

$$GS_{g_9} = f_{y2} / F_{emw_YDKT} \quad (3.44)$$

g_5 ve g_9 için kısıta uygunluk kontrolü denklem (3.45) ve (3.46)'de gösterildiği gibi yapılmıştır.

$$GS_{g_5} < 1 \Rightarrow g_5(n) = 0 \quad (3.45)$$

$$GS_{g_5} \geq 1 \Rightarrow g_5(n) = 1$$

$$GS_{g_9} < 1 \Rightarrow g_9(n) = 0 \quad (3.46)$$

$$GS_{g_9} \geq 1 \Rightarrow g_9(n) = 1$$

3.3.6.6 Gövdeye Gelen Bileşke Gerilmelerin Kısıt Kontrolleri (g_6 ve g_{10})

Gövdeye gelen bileşke gerilmeler (f_r) denklem (3.47) ile hesaplanmıştır.

$$f_r = \sqrt{f_x^2 + f_y^2} \quad (3.47)$$

Gövdeye gelen bileşke gerilmelerin kaynak kapasitesini kullanım oranları GKT için (GS_{g_6}) denklem (3.48) ve YDKT için ($GS_{g_{10}}$) denklem (3.49)'de gösterildiği gibi hesaplanmıştır.

$$GS_{g_6} = f_{y2} / F_{emw_GKT} \quad (3.48)$$

$$GS_{g_{10}} = f_{y2} / F_{emw_YDKT} \quad (3.49)$$

g_6 ve g_{10} için kısıta uygunluk kontrolü denklem (3.50) ve (3.51)'de gösterildiği gibi yapılmıştır.

$$GS_{g_6} < 1 \Rightarrow g_{6(n)} = 0 \quad (3.50)$$

$$GS_{g_6} \geq 1 \Rightarrow g_{6(n)} = 1$$

$$GS_{g_{10}} < 1 \Rightarrow g_{10(n)} = 0 \quad (3.51)$$

$$GS_{g_{10}} \geq 1 \Rightarrow g_{10(n)} = 1$$

3.3.7 Adil Ceza Yöntemi ve Ceza Fonksiyonu

Kısıtlı optimizasyon problemlerinin çözülebilmesi için problemin yapısına göre seçilebilecek pek çok farklı cezalandırma metodu bulunmaktadır. Bu tez çalışmasında “adil ceza yöntemi” olarak isimlendirilen özgün bir sistem önerilmiştir. Adil ceza yöntemi, her bir popülasyon elemanının ihlal ettiği kısıt sayısı nispetinde cezalandırması fikrini temel almaktadır. Adaletli cezalar ile, popülasyonun daha kısa sürede daha iyi yerlere gidebilmesi için imkân sağlanarak algoritmaların performansının artırılması hedeflenmektedir. Bölüm 3.3.6’da popülasyon elemanlarının kısıt kontrolleri yapılırken her bir elemanın, her bir kısıt için kaç kere kısıtı ihlal ettiği yani her bir satırın her bir kısıt için hata sayacıları $g_{k(n)}$ sütun vektörlerine kayıt edilmiştir.

Bölüm 3.3.2’de tanıtılan sayısal modeller ele alındığında, algoritmalar cezalandırma aşamasına gelince her bir tasarım prensibi için 6 adet kısıt hata sayacı elde edilecektir. Bunlar GKT için $g_1, g_2, g_3, g_4, g_5, g_6$ iken YDKT için $g_1, g_2, g_7, g_8, g_9, g_{10}$ ’dur. Bu kısıt hata sayacıları adil ceza yöntemi için kullanılacaktır. Cezalandırma fonksiyonunda kullanılacak g_{top} parametresi denklem (3.52)’de gösterildiği gibi hesaplanmaktadır.

$$g_{top(n)} = \sum_{x=1}^{k_{max}} (g_x(n)) \quad (3.52)$$

Adil ceza yönteminin açıklanabilmesi amacıyla; $n=3$ elemanlı bir popülasyonun kısıt kontrolleri rastlantısal olarak verilerek, YDKT için Çizelge 3.21 oluşturulmuştur. İlgili hata sayacıları yan yana getirilerek satır bazında toplanarak $g_{top(n)}$ elde edilmiştir.

Çizelge 3.21 YDKT için rastlantısal olarak oluşturulmuş örnek kısıt hata sayaçları.

n	g_1	g_2	g_7	g_8	g_9	g_{10}	g_{top}
1	2	0	0	1	0	0	3
2	0	1	0	0	0	0	1
3	0	3	0	0	1	0	4

Elde edilen $g_{top(n)}$ değeri, bir satırın bütün kısıtları kaç kere ihlal ettiğini göstermektedir. Bu durumda hangi popülasyon elemanının kısıtlara daha çok uyduğu, hangisinin kısıtları daha çok ihlal ettiğini matematiksel olarak gösteren bir parametre elde edilmiş olur. Adil ceza metodunun gereği olarak her bir satır, kısıtı ihlal ettiği ölçüde yani $g_{top(n)}$ nispetinde cezalandırılacaktır. Adil ceza yöntemi için kullanılan ceza fonksiyonu denklem (3.53)'de verilmiştir.

$$p_{(n)} = kc_k(F_{(n)} + 1000) * g_{top(n)} \quad (3.53)$$

Bu denklemlerde,

$p_{(n)}$: Her bir eleman için verilecek ceza miktarı.

k : Kısıt no.

kc_k : Her bir kısıt için kısıt ceza çarpanı. Gerekli durumlarda kısıtlardan herhangi birisi için verilecek cezanın artırılıp azaltılmasına imkân tanımaktadır.

$F_{(n)}$: Her bir eleman için ceza öncesi hesaplanmış amaç fonksiyonu değeri.

$g_{top(n)}$: Her bir eleman için kısıtların toplam ihlal edilme sayısı.

Cezalandırma öncesindeki amaç fonksiyonunun çarpım durumundaki tasarım değişkenlerinden oluşmaktadır. Bu durum minimizasyon problemleri için ceza fonksiyonuna bir sabit eklenme ihtiyacını ortaya çıkarmaktadır. Sabitin olmadığı durumda algoritma bütün tasarım değişkenlerini sıfıra götürdüğünde ceza verilemeyecektir. Bu sorunun önüne geçilebilmesi maksadıyla ceza fonksiyonuna toplam durumunda 1000 sabiti eklenmiştir. Adil Ceza Yöntemi bu haliyle, minimizasyon ya da maksimizasyon için kısıtlı optimizasyon problemlerinin çözümünde kullanılabilecek şekilde tasarlanmıştır.

Bu çalışmada kullanılan ceza fonksiyonlarında, $k=1,2,4,5,6,8,9,10$ için $kc_k=15$ seçilmiştir. Başlık dış kaynak grubuna gelen gerilmelerin kontrolü kısıtları cezalandırılırken ($k=3$ ve $k=7$ için) kısıt ceza çarpanı beş katına çıkarılarak $kc_k=75$ olarak seçilmiştir. Bunun nedeni g_3 ve g_7 kısıtlarını sağlamak için daha büyük cezalara ihtiyaç duyulmasıdır.

3.3.8 Optimizasyon Algoritmaları İçin Belirlenen Parametreler

Optimizasyon algoritmalarının çalıştırılabilmesi için bazı parametrelere ihtiyaç duyulmaktadır. Bu parametreleri algoritmaya özel olmayan çalıştırma parametreleri ve algoritmaya özel parametreler olarak ikiye ayırmak mümkündür. Jaya ve TLBO algoritmaları algoritmaya özel herhangi bir parametreye ihtiyaç duymazken SCA'nın algoritmaya özel parametrelere gereksinimi vardır.

Algoritmaya özel olmayan çalıştırma parametreleri her algoritma için ortak seçilmiştir. Bu parametreler; popülasyon eleman sayısı (n), tasarım değişkeni sayısı (d) iterasyon sayısı (T), arama uzayının sınırlarını belirleyen her bir tasarım değişkeni için alt sınırlar (p_{alt}) ve her bir tasarım değişkeni için üst sınırlar ($p_{üst}$)'dür.

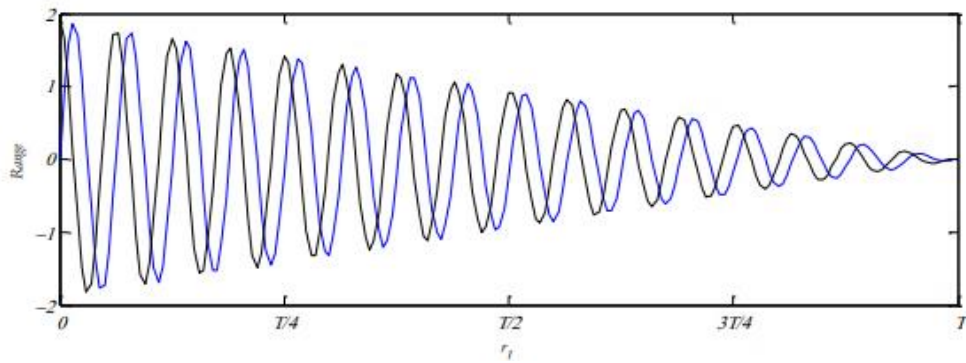
Seçilen ortak parametreler Çizelge 3.22'de verilmiştir. Çizelge 3.22'de görüldüğü gibi; çeşitli kıyaslama durumları elde edebilmek amacıyla üç farklı popülasyon eleman sayısı (n) ile analiz yapılmıştır. 3 farklı sayısal model, 2 farklı tasarım prensibi, 3 farklı popülasyon eleman sayısı ve 3 farklı optimizasyon algoritması ile toplamda birbirinden farklı 54 adet analiz gerçekleştirilmiştir.

Bütün algoritmaların global optimuma ulaşabilmesi amacıyla iterasyon sayısı (T) mümkün olduğunca büyük seçilmiştir. Arama uzayının sınırlarını belirleyen parametreler olan p_{alt} ve $p_{üst}$, algoritmaları gereksiz aramaların yapılacağı bölgelerden mümkün olduğunca uzak tutacak kadar dar, çözümü içerebilecek bölgelere mümkün olduğunca kapsayacak kadar geniş bir aralıktan seçilmeye çalışılmıştır.

Çizelge 3.22 Algoritmalar için ortak çalışma parametreleri.

Sembol	Parametre Değeri
n	250 500 1000
d	6
T	1000
p_{alt}	a_1, a_2 ve a_3 için 3 mm L_1, L_2 ve L_3 için 20 mm
$p_{üst}$	a_1, a_2 ve a_3 için 15 mm L_1 için 300 mm L_2 için 118 mm L_3 için 261 mm

SCA için yukarıda açıklanan ortak çalışma parametrelerine ek olarak algoritmaya özel bazı parametrelere de ihtiyaç duyulmaktadır. İlgili parametreler Bölüm 3.1.3’de açıklanmıştır. a parametresinin seçimi için literatür incelendiğinde, Mirjalili (2015) tarafından SCA’nın tanıtıldığı çalışmada a parametresinin nasıl seçileceğine ilişkin bir öneri bulunmamaktadır. Çalışmada sinüs ve kosinüs arama fonksiyonları için fonksiyon genliğini ayarlayan r_1 ’in açıklandığı bir grafikte $a=3$ seçildiği görülmektedir (Şekil 3.17). Buradan hareketle arama uzayının taranmasında a ’nın oldukça etkin bir parametre olduğu anlaşılmaktadır.



Şekil 3.17 Sinüs ve kosinüs aralığı için azalan model ($a=3$) (Mirjalili, 2015).

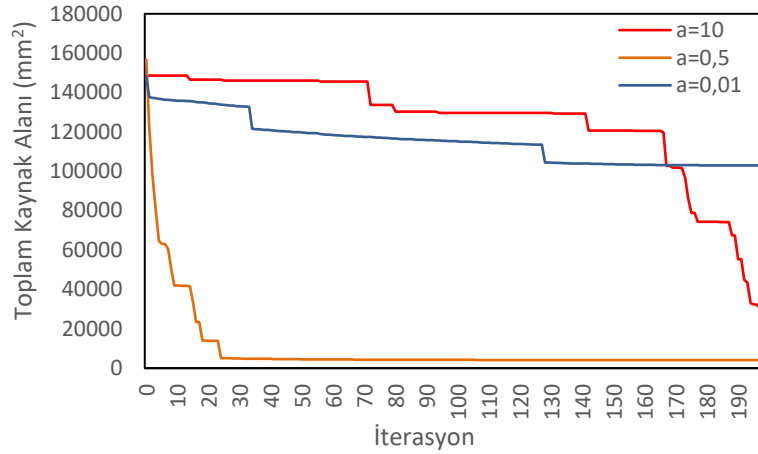
Demiral (2019) tarafından yapılan çalışmada gezgin satıcı problemi için SCA’nın parametre analizleri gerçekleştirilmiştir. Çalışmada esas alınan parametreler a , b ve T ’dir. Özellikle “ a ” parametresinin algoritmanın performansı üzerinde çok etkin olduğu vurgulanmıştır. Çalışmanın sonucunda gezgin satıcı problemi için en iyi sonucu veren parametreler $a=0,1$, $b=2$ ve $T=1000$ bulunmuştur.

Bu çalışmada, b parametresi rastlantısallığa eşit şans vermek amacıyla 2 olarak seçilmiştir. a parametresinin seçilebilmesi için bir dizi analiz yapılması ihtiyacı ortaya çıkmıştır. Algoritmaları en çok zorlayan sayısal modellerden birisi olan SM-1A için $a=3,00$ 'dan $a=0,10$ 'a kadar bir dizi denemeler gerçekleştirilmiştir. Denemelerin kısa sürede tamamlanabilmesi maksadıyla $n=100$, $T=200$ seçilerek denenen her bir a parametresi için 10 adet tekrarlı çalıştırma yapılmıştır. Çalıştırmalardan elde edilen en iyi sonuçlar, ortalama sonuçlar, standart sapmalar ve toplam kısıt ihlal sayıları Çizelge 3.23'de verildiği gibidir.

Çizelge 3.23 a parametresi için 10 çalıştırmanın sonuçları.

a	En iyi (mm^2)	Ortalama (mm^2)	St. Sapma	$\sum g_{top}$
3,00	4100,21	4395,56	339,03	0
2,00	4093,80	4184,17	115,53	0
1,00	4035,65	4076,69	29,86	0
0,75	4012,05	4090,80	67,58	0
0,50	3980,46	4044,77	36,05	0
0,25	4005,95	4139,99	72,57	0
0,10	4183,07	37701,47	42546,06	4

Çizelge 3.23'e göre, ilgili model için a parametresi, 3'ten 0,50'ye yaklaştıkça kıyaslama parametrelerinde olumlu değişimler gözlenmiştir. En iyi sonuçlar $a=0,50$ iken elde edilmiştir. $a < 0,50$ durumda bütün karşılaştırma değerlerinde olumsuz değişimler, hatta $a=0,10$ için kısıtları sağlayamayacak kadar kötü sonuçlara rastlanmıştır. Bütün bunların neticesinde $a=0,50$ olarak seçilmiştir. Optimumdan çok büyük ($a=10$), optimumdan çok küçük ($a=0,01$) ve optimum bölgede ($a=0,5$) seçilme durumları için yukarıdaki çalıştırma şartları ile analizler yapıp Şekil 3.18 elde edilmiştir. a optimum değerinden çok büyük seçildiğinde ($a=10$) çok büyük arama adımları ile arama gerçekleştirildiği görülmüştür. Arama uzayında çok büyük adımlar ile yapılan taramalarda lokalde takılmaların ardından daha iyi sonucu bulunca çok sert düşüşler olduğu görülmüştür. İterasyon-toplam kaynak alanı grafiğinin oluşturduğu desen merdiven kesitini andırmaktadır. a optimum değerinden çok aşağıda seçildiğinde ($a=0,01$) ise arama uzayı çok küçük arama adımları ile tarandığından optimum değere ulaşmak zorlaşarak lokalde takılmaların olduğu ve grafikte düz çizgi oluşturma eğilimi görülmektedir. Optimuma bölgede ($a=0,5$) tipik metasezgisel algoritma davranışı ile karşılaşılmıştır. Daha hassas analizler yapılarak daha başarılı parametrelerin tespit edilme ihtimali her zaman için mümkündür. SCA için seçilen algoritmaya özel çalışma parametreleri Çizelge 3.24'de verilmiştir.



Şekil 3.18 Geniş aralıkta seçilen a parametrelerinin sonuçları.

Çizelge 3.24 SCA için özel çalışma parametreleri.

Sembol	Seçilen Parametre Değeri
a	0,5
b	2

3.3.9 Algoritma Performanslarının Karşılaştırılması

Her bir algoritma için, tüm sayısal modellerden popülasyon eleman sayıları $n=250$, $n=500$ ve $n=1000$ için bazı karşılaştırma parametreleri toplanmıştır. Bu parametreler, her biri 1000 iterasyonluk 10 çalıştırma için, ulaşılan en iyi değer (minimum kaynak alanı), ortalama değer (ortalama kaynak alanı), ulaşılan en kötü değer (maksimum kaynak alanı), ulaşılan toplam kaynak alanlarının standart sapmaları ve her biri 1000 iterasyonluk 10 çalıştırmanın tamamlanma süresidir. İşlem süreleri her analiz için aynı bilgisayardan özdeş koşullarda alınmıştır. Analizlerde kullanılan bilgisayar Intel® Core™ i3-3120M CPU @ 2,50 GHz işlemciye ve 8GB RAM belleğine sahip olup Windows 10 Home 64 bit işletim sistemi ile çalışmaktadır. Bütün analizler Matlab yazılım dilinde oluşturulmuş kodlar vasıtasıyla MATLAB (2016) programı kullanılarak yapılmıştır. Ayrıca karşılaştırmalar için her bir iterasyondaki amaç fonksiyonu değerlerinin ortalamaları elde edilerek iterasyon-toplam kaynak alanı grafikleri oluşturulmuştur.

4. BULGULAR

Oluşturulmuş olan sayısal modeller, optimizasyon algoritmaları ile çözümlenmiş ve sonuçlar bu bölümde verilmiştir. Elde edilen sonuçlar, Bölüm 3.3.6’da detayları verilen ve ÇYTHYE (2018)’na bağlı kalınarak tanımlanan tüm kısıtları sağlamaktadır.

4.1 Sayısal Model – 1

Sayısal model – 1 momentin etkin olduğu durumu ifade etmektedir. SM-1A için (moment etkin durumdaki GKT tasarım prensibi ile optimizasyon) elde edilen en iyi sonuçlar Çizelge 4.1’de verilmiştir. İlgili veriler her bir satırda açıklanan çalıştırma koşulları için her biri 1000 iterasyonluk 10 çalıştırmanın neticesinde elde edilmiştir.

Çizelge 4.1 SM-1A için en iyi sonuç değerleri.

	<i>n</i>	<i>a</i> ₁ (mm)	<i>a</i> ₂ (mm)	<i>a</i> ₃ (mm)	<i>L</i> ₁ (mm)	<i>L</i> ₂ (mm)	<i>L</i> ₃ (mm)	<i>A_w</i> (mm ²)	GS- 1	GS- 2	GS- 3	GS- 4
<i>Jaya</i>	250	4,67	4,00	4,00	300,00	48,00	48,00	3955,28	1,00	0,13	0,22	0,25
	500	4,67	4,00	4,00	300,00	48,00	48,00	3955,26	1,00	0,13	0,22	0,25
	1000	4,67	4,00	4,00	300,00	48,01	48,00	3955,36	1,00	0,13	0,22	0,25
<i>SCA</i>	250	4,66	4,00	4,00	300,00	48,71	48,38	3960,86	1,00	0,13	0,21	0,25
	500	4,67	4,03	4,00	296,36	50,28	48,07	3965,59	1,00	0,13	0,22	0,25
	1000	4,65	4,01	4,00	298,89	49,69	48,06	3961,83	1,00	0,13	0,22	0,25
<i>TLBO</i>	250	4,67	4,00	4,00	300,00	48,00	48,00	3955,25	1,00	0,13	0,22	0,25
	500	4,67	4,00	4,00	300,00	48,00	48,00	3955,25	1,00	0,13	0,22	0,25
	1000	4,67	4,00	4,00	300,00	48,00	48,00	3955,25	1,00	0,13	0,22	0,25

Çizelge 4.1’de görüldüğü gibi, toplam kaynak alanı (*A_w*) için elde edilen en iyi sonuç 3955,25 mm²’dir. Bu sonuca TLBO algoritmasının *n*=250, *n*=500 ve *n*=1000 popülasyon elemanı için yapılan analizlerin tamamında ulaşılmıştır. En iyi değer için başlık dış, başlık iç ve gövde kaynak gruplarının kalınlıkları sırasıyla *a*₁=4,67, *a*₂=4,00 ve *a*₃=4,00mm; kaynak uzunlukları ise sırasıyla *L*₁=300,00, *L*₂=48,00 ve *L*₃=48,00 mm olarak bulunmuştur. Jaya algoritması için elde edilen sonuçlar da en iyi sonuca oldukça yakındır. Jaya algoritması ile elde edilen en iyi sonuçta; popülasyon eleman sayısı *n*=500 için toplam kaynak alanı 3955,26 mm² bulunmuştur. Kaynak uzunlukları ve kaynak kalınlıkları için TLBO algoritması ile farklar mikrometreler mertebesinde. SCA ile elde edilen en iyi sonuç popülasyon eleman sayısı *n*=250 için 3960,86 mm²’dir. Bu sonuç için başlık dış, başlık iç ve gövde kaynak gruplarının kalınlıkları sırasıyla *a*₁=4,66,

$a_2=4,00$ ve $a_3=4,00$ mm; kaynak uzunlukları ise sırasıyla $L_1=300,00$, $L_2=48,71$ ve $L_3=48,38$ mm olarak bulunmuştur. Bütün algoritmaların bulduğu en iyi sonuçlar için başlık dış kaynak grubunun üst ucuna gelen gerilmelerin kapasite kullanımı (GS-1) %100'e oldukça yakın, gövde kaynağının üst ucunda oluşan gerilmelerin kapasite kullanımı (GS-2) %13, gövdeye gelen kesme gerilmelerinin kapasite kullanımı (GS-3) %22, gövde bileşke kapasite kullanımları (GS-4) ise yaklaşık %25'dir. Bu kapasite kullanımlarının neticesinde momentin etkin olduğu durumun en çok başlığın dış ucunu zorladığı görülmüştür. Ayrıca, kapasite kullanımı arttıkça bulunan kaynak alanı azalmaktadır. Buradan hareketle algoritmalar tarafından yapılan aramanın maksimum kapasite kullanımı ile minimum kaynak alanı olduğu doğrulanmıştır. SM-1A için her biri 1000 iterasyonluk 10 tekrarlı çalıştırmada toplam kaynak alanları esas alınarak oluşturulmuş algoritma karşılaştırma parametreleri Çizelge 4.2'de verilmiştir.

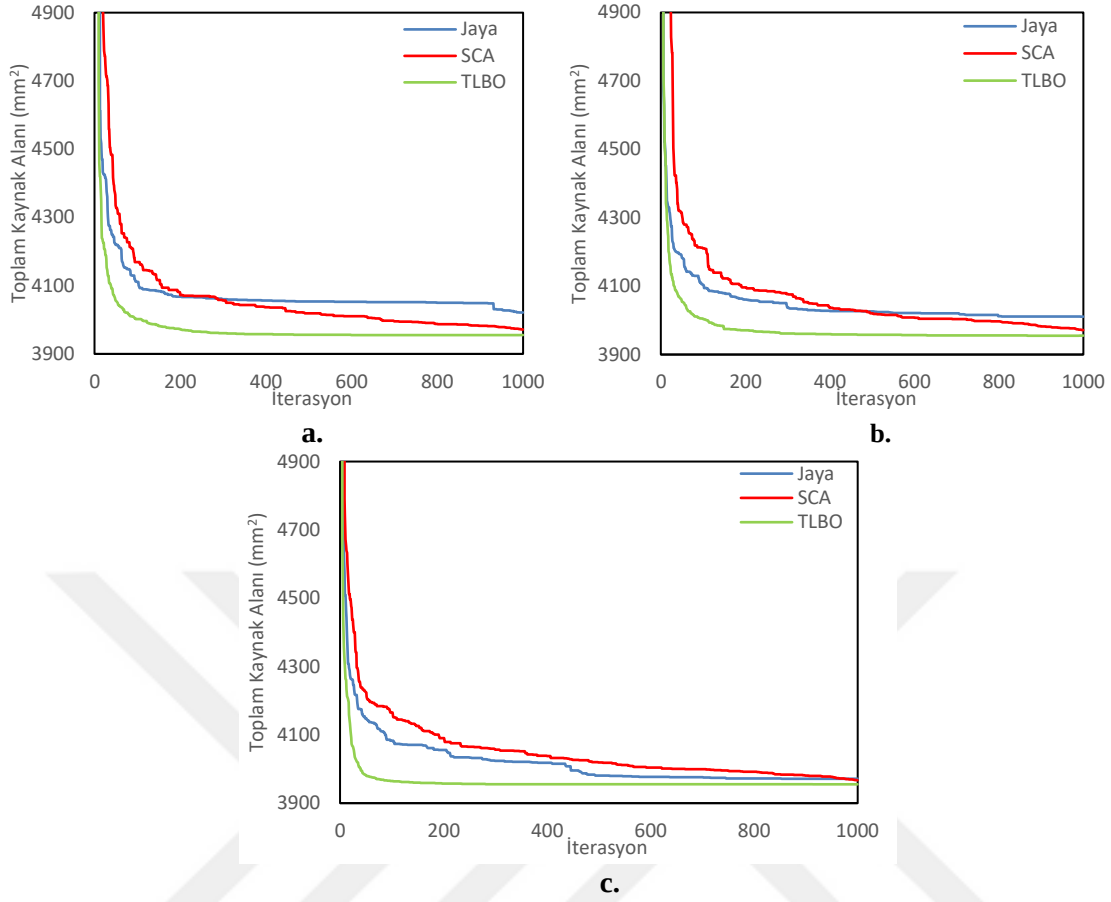
Çizelge 4.2 SM-1A için algoritma karşılaştırma parametreleri.

<i>n</i>	Durum	Jaya	SCA	TLBO
250	En İyi (mm ²)	3955,28	3960,86	3955,25
	Ortalama (mm ²)	4021,31	3971,43	3955,26
	En Kötü (mm ²)	4231,91	3979,33	3955,29
	Standart Sapma	101,92	6,29	0,01
	İşlem Süresi (sn)	63,68	61,40	139,24
500	En İyi (mm ²)	3955,26	3965,59	3955,25
	Ortalama (mm ²)	4010,78	3971,33	3955,34
	En Kötü (mm ²)	4184,49	3978,12	3955,82
	Standart Sapma	78,99	4,97	0,20
	İşlem Süresi (sn)	124,61	116,48	272,18
1000	En İyi (mm ²)	3955,36	3961,83	3955,25
	Ortalama (mm ²)	3971,51	3967,09	3955,25
	En Kötü (mm ²)	4022,45	3976,53	3955,25
	Standart Sapma	25,50	5,09	0,00
	İşlem Süresi (sn)	153,59	184,18	486,79

Çizelge 4.2'den hareketle, tüm popülasyon eleman sayıları (*n*) için en iyi sonuçlar karşılaştırıldığında TLBO algoritmasının en başarılı algoritma olduğu görülmektedir. Jaya algoritması ile bulunan en iyi sonuçlar TLBO algoritmasına çok yakındır. SCA ile elde edilen sonuçlar kullanılabilir olmakla beraber en başarısız sonuçları içermektedir. Ortama sonuçlar dikkate alındığında, TLBO algoritması en iyi ortalama değerlere sahipken ardından SCA ve Jaya algoritması gelmektedir. En kötü değerler karşılaştırıldığında, Jaya algoritması en başarısız konumdadır. En büyük kaynak alanları Jaya algoritması ile bulunmuştur. Jaya algoritmasının ardından en büyük kaynak

alanlarını SCA vermiştir. Sonuçlar arasındaki en büyük standart sapmalar Jaya algoritmasına aittir. Bu durum Jaya algoritmasının 10 çalıştırma için görece geniş bir aralıkta sonuçlar bulduğunu göstermektedir. Jaya algoritmasının en iyi sonuçlarda en başarılı algoritmaya çok yakinken en kötü sonuçlarda en başarısız algoritma olmasının sebebi bu şekilde açıklanabilir. TLBO algoritması en küçük standart sapma değerlerine sahiptir. Bu sonuç TLBO algoritmasının her çalıştırma için birbirine çok yakın ve kararlı sonuçlar verdiğini göstermektedir. TLBO algoritmasının $n=1000$ için standart sapması neredeyse sıfır bulunmuştur yani algoritma her çalıştırmada aynı sonuca ulaşabilmiştir. İşlem süreleri kıyaslandığında TLBO algoritması çözüm için en çok zaman harcayan algoritma olmuştur öyle ki TLBO'nun çözümü bulması için geçen süre diğer algoritmaların iki katından daha fazladır. Jaya algoritması ve SCA birbirine yakın sürelerde çözümü tamamlamıştır. Çözüm süreleri yakın olan sonuçlar karşılaştırılmak istenirse, TLBO algoritması için $n=250$ elemanla bulunan sonuçlar (139,24 sn) Jaya algoritması için $n=1000$ eleman ile bulunan sonuçlardan (153,59 sn) ve SCA için $n=1000$ eleman ile bulunan sonuçlardan (184,18 sn) her karşılaştırma parametresinde daha başarılıdır. Yani TLBO algoritması eşit elemanla iki katından daha uzun sürelerde çözüm bulsa dahi, daha az elemanla ve daha az süre ile diğer algoritmaların daha çok elemanla ve daha çok süre ile bulduğu sonuçlardan daha başarılı sonuçlar bulmuştur.

Çizelge 4.2'de verilen çalıştırma koşulları için iterasyon – toplam kaynak alanı grafikleri aşağıda verilmiştir. Grafiklerde her bir çalıştırma koşulu için her biri 1000 iterasyonluk 10 çalıştırmanın ortalama değerleri gösterilmektedir. Şekil 4.1.a $n=250$ eleman için, Şekil 4.1.b $n=500$ eleman için, Şekil 4.1.c ise $n=1000$ eleman için çizilmiştir. Grafikler incelendiğinde, popülasyon eleman sayısı n arttıkça algoritmaların her bir iterasyon için buldukları toplam kaynak alanı değerlerinin birbirine daha erken yaklaştığı görülmektedir. Buradan hareketle eleman sayısı ile global optimuma ulaşma süresi arasında ters orantı olduğu görülmektedir. Çalıştırma koşulu farketmeksizin TLBO algoritması global optimuma ilk ulaşan algoritma olmuştur. Grafikler incelendiğinde SCA ile Jaya algoritmasının performansları birbirine yakın bulunmuştur. Bütün popülasyon eleman sayıları için iterasyonun erken aşamalarında Jaya algoritması daha küçük kaynak alanları bulurken iterasyon ilerledikçe SCA'nın öne geçtiği görülmektedir.



Şekil 4.1 SM-1A için 10 çalıştırmanın ortalama değerleri. **a.** $n=250$, **b.** $n=500$, **c.** $n=1000$.

SM-1B (moment etkin durumdaki YDKT tasarım prensibi ile optimizasyon) için yapılan analizlerin sonuçları Çizelge 4.3’de verilmiştir. İlgili veriler her bir satır için 1000 iterasyonla yapılan 10 tekrarlı çalıştırmanın en iyi sonuçlarını göstermektedir.

Çizelge 4.3 SM-1B için en iyi sonuç değerleri.

	n	a_1 (mm)	a_2 (mm)	a_3 (mm)	L_1 (mm)	L_2 (mm)	L_3 (mm)	A_w (mm ²)	GS- 1	GS- 2	GS- 3	GS- 4
Jaya	250	4,27	4,00	4,00	300,00	48,00	48,00	3716,75	1,00	0,13	0,17	0,22
	500	4,27	4,00	4,00	300,00	48,01	48,00	3716,76	1,00	0,13	0,17	0,22
	1000	4,27	4,00	4,00	300,00	48,02	48,01	3717,05	1,00	0,13	0,17	0,22
SCA	250	4,24	4,00	4,01	298,98	50,02	48,06	3724,01	1,00	0,13	0,17	0,22
	500	4,25	4,01	4,03	299,84	49,16	48,21	3725,12	1,00	0,13	0,17	0,21
	1000	4,28	4,01	4,00	298,86	48,32	48,13	3720,53	1,00	0,13	0,17	0,22
TLBO	250	4,27	4,00	4,00	300,00	48,00	48,00	3716,73	1,00	0,13	0,17	0,22
	500	4,27	4,00	4,00	300,00	48,00	48,00	3716,73	1,00	0,13	0,17	0,22
	1000	4,27	4,00	4,00	300,00	48,00	48,00	3716,73	1,00	0,13	0,17	0,22

Çizelge 4.3’ e göre, toplam kaynak alanı (A_w) için elde edilen en iyi sonuç 3716,73 mm²’dir. Bu sonuca TLBO algoritmasının $n=250$, $n=500$ ve $n=1000$ popülasyon elemanı

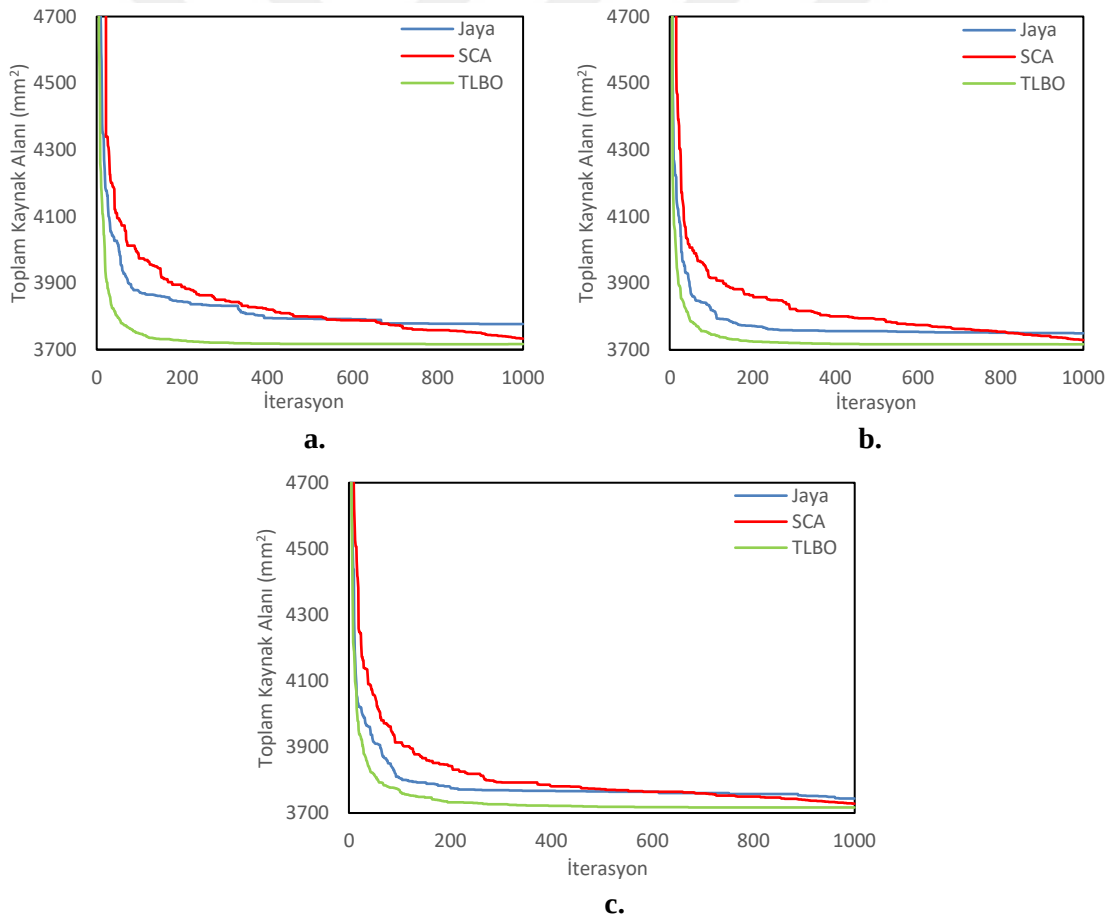
için yapılan analizlerin tamamında ulaşılmıştır. En iyi değer için başlık dış, başlık iç ve gövde kaynak gruplarının kalınlıkları sırasıyla $a_1=4,27$, $a_2=4,00$ ve $a_3=4,00$ mm; kaynak uzunlukları ise sırasıyla $L_1=300,00$, $L_2=48,00$ ve $L_3=48,00$ mm olarak bulunmuştur. Jaya algoritması için elde edilen sonuçlar da en iyi sonuca oldukça yakındır. Kaynak uzunluk ve kalınlıkları için bulunan farklar mikrometre seviyesindedir. Jaya algoritması ile elde edilen en iyi sonuçta; popülasyon eleman sayısı $n=250$ için toplam kaynak alanı $3716,75 \text{ mm}^2$ olarak bulunmuştur. SCA ile elde edilen en iyi sonuç popülasyon eleman sayısı $n=1000$ için $3720,53 \text{ mm}^2$ 'dir. Bu sonuç için başlık dış, başlık iç ve gövde kaynak gruplarının kalınlıkları sırasıyla $a_1=4,28$, $a_2=4,01$ ve $a_3=4,00$ mm; kaynak uzunlukları ise sırasıyla $L_1=298,86$, $L_2=48,32$ ve $L_3=48,13$ mm bulunmuştur. Algoritmalarca elde edilen sonuçların tamamında başlık kaynak grubunun kapasite kullanımı (GS-1) %100'e oldukça yakındır. Algoritmalar YDKT için moment baskın modeli optimize ederken kritik olan başlık dış kaynak grubunun kapasite kullanımını maksimize etme eğilimindedir. Gövde kaynak grubu için kapasite kullanımları oldukça düşük kalmıştır.

Çizelge 4.4 SM-1B için algoritma karşılaştırma parametreleri.

n	Durum	Jaya	SCA	TLBO
250	En İyi (mm^2)	3716,75	3724,01	3716,73
	Ortalama (mm^2)	3776,97	3733,04	3716,73
	En Kötü (mm^2)	4105,20	3744,39	3716,74
	Standart Sapma	131,62	7,36	0,00
	İşlem Süresi (sn)	63,85	61,04	139,02
500	En İyi (mm^2)	3716,76	3725,12	3716,73
	Ortalama (mm^2)	3749,95	3729,64	3716,73
	En Kötü (mm^2)	3902,78	3740,36	3716,73
	Standart Sapma	56,53	4,74	0,00
	İşlem Süresi (sn)	98,82	94,96	249,01
1000	En İyi (mm^2)	3717,05	3720,53	3716,73
	Ortalama (mm^2)	3743,90	3728,15	3716,75
	En Kötü (mm^2)	3852,48	3736,46	3716,81
	Standart Sapma	44,24	4,85	0,03
	İşlem Süresi (sn)	182,14	187,90	481,23

SM-1B için her biri 1000 iterasyonluk 10 tekrarlı çalıştırmada toplam kaynak alanları esas alınarak oluşturulmuş algoritma karşılaştırma parametreleri Çizelge 4.4'de verilmiştir. Elde edilen karşılaştırma sonuçları SM-1A ile benzerlik göstermektedir. En iyi sonuçlar için başarı sıralaması TLBO, Jaya ve SCA şeklinde olmuştur. Ortalama değerler kıyaslanınca en başarılı algoritma TLBO iken bunu SCA ve Jaya algoritması izlemektedir. En kötü sonuçlara bakıldığında en başarısız sonuçlar Jaya algoritması ile

elde edilmektedir. Jaya algoritması en iyi sonuçlarda en başarılı algoritmaya çok yakinken en kötü sonuçlarda en başarısız algoritma konumundadır. Bunun sebebinin standart sapmasının diğer algoritmalarla göre çok büyük olmasından kaynaklandığı düşünülmektedir. Standart sapması en küçük algoritma TLBO olmuştur. Süreler bakımından TLBO algoritması diğer algoritmaların iki katından daha fazla zamanda çözüme ulaşabilmiştir. Buna rağmen yakın süreli analizler incelendiğinde TLBO algoritmasının daha az süreyle daha başarılı sonuçlar bulabildiği görülmektedir. SCA ve Jaya birbirine yakın sürelerde analizleri tamamlamıştır. Çizelge 4.4’de verilen çalıştırma koşulları için iterasyon – toplam kaynak alanı grafikleri aşağıda verilmiştir. Grafiklerde her bir çalıştırma koşulu için her biri 1000 iterasyonluk 10 çalıştırmanın ortalama değerleri gösterilmektedir.



Şekil 4.2 SM-1B için 10 çalıştırmanın ortalama değerleri. **a.** $n=250$, **b.** $n=500$, **c.** $n=1000$.

Şekil 4.2.a $n=250$ eleman için, Şekil 4.2.b $n=500$ eleman için, Şekil 4.2.c ise $n=1000$ eleman için çizilmiştir. Ortalama sonuçları esas alan grafikler incelendiğinde, her bir

çalıştırma durumu için TLBO algoritmasının global optimuma ilk ulaşan algoritma olduğu görülmektedir. Jaya algoritması iterasyonun erken aşamalarında SCA'ya göre daha küçük kaynak alanları bulmuştur. İterasyon ilerledikçe SCA, Jaya algoritmasının önüne geçerek daha başarılı sonuçlar elde etmiştir.

4.2 Sayısal Model – 2

Sayısal model-2 (SM-2) normal kuvvetin etkin olduğu durumu ifade etmektedir. Çizelge 4.5’de SM-2A (normal kuvvet etkin durumdaki GKT tasarım prensibi ile optimizasyon) için, $n=250$, $n=500$ ve $n=1000$ popülasyon elemanı ile 1000 iterasyonluk 10 çalıştırmada elde edilen en iyi sonuçlar verilmiştir.

Çizelge 4.5 SM-2A için en iyi sonuç değerleri.

	n	a_1 (mm)	a_2 (mm)	a_3 (mm)	L_1 (mm)	L_2 (mm)	L_3 (mm)	A_w (mm ²)	GS- 1	GS- 2	GS- 3	GS- 4
Jaya	250	4,00	4,00	4,00	142,07	48,03	48,03	2290,66	1,00	0,78	0,22	0,81
	500	4,00	4,00	4,00	141,87	48,01	48,31	2291,42	1,00	0,78	0,22	0,81
	1000	4,00	4,00	4,00	141,18	48,55	48,01	2291,05	1,00	0,78	0,22	0,81
SCA	250	4,06	4,01	4,03	137,93	48,71	49,37	2298,23	1,00	0,78	0,21	0,81
	500	4,01	4,01	4,00	133,75	51,73	49,45	2298,95	1,00	0,78	0,21	0,81
	1000	4,06	4,02	4,02	134,55	50,67	48,92	2299,11	1,00	0,78	0,21	0,81
TLBO	250	4,00	4,00	4,00	142,27	48,00	48,00	2290,19	1,00	0,78	0,22	0,81
	500	4,00	4,00	4,00	142,27	48,00	48,00	2290,19	1,00	0,78	0,22	0,81
	1000	4,00	4,00	4,00	142,27	48,00	48,00	2290,19	1,00	0,78	0,22	0,81

Çizelge 4.5’ e göre toplam kaynak alanı (A_w) için elde edilen en iyi sonuç 2290,19 mm²’dir. Bu sonuca TLBO algoritmasının $n=250$, $n=500$ ve $n=1000$ popülasyon elemanı için yapılan analizlerin tamamında ulaşılmıştır. En iyi değer için başlık dış, başlık iç ve gövde kaynak gruplarının kalınlıkları sırasıyla $a_1 = a_2 = a_3 = 4,00$ mm; kaynak uzunlukları ise sırasıyla $L_1 = 142,27$, $L_2 = L_3 = 48,00$ mm olarak bulunmuştur. Jaya algoritması için elde edilen sonuçlar da TLBO algoritması ile bulunan sonuçlara oldukça yakındır. $n=250$ için toplam kaynak alanı 2290,66 mm² olarak tespit edilmiştir. Başlık dış, başlık iç ve gövde kaynak gruplarının kalınlıkları sırasıyla $a_1 = a_2 = a_3 = 4,00$ mm; kaynak uzunlukları ise sırasıyla $L_1 = 142,07$, $L_2 = L_3 = 48,03$ mm bulunmuştur. SCA ile elde edilen en iyi sonuç popülasyon eleman sayısı $n=250$ için 2298,23 mm²’dir. Bu sonuç için başlık dış, başlık iç ve gövde kaynak gruplarının kalınlıkları sırasıyla $a_1 = 4,06$, $a_2 = 4,01$ ve $a_3 = 4,03$ mm; kaynak uzunlukları ise sırasıyla $L_1 = 137,93$, $L_2 = 48,71$ ve $L_3 = 49,37$ mm olarak

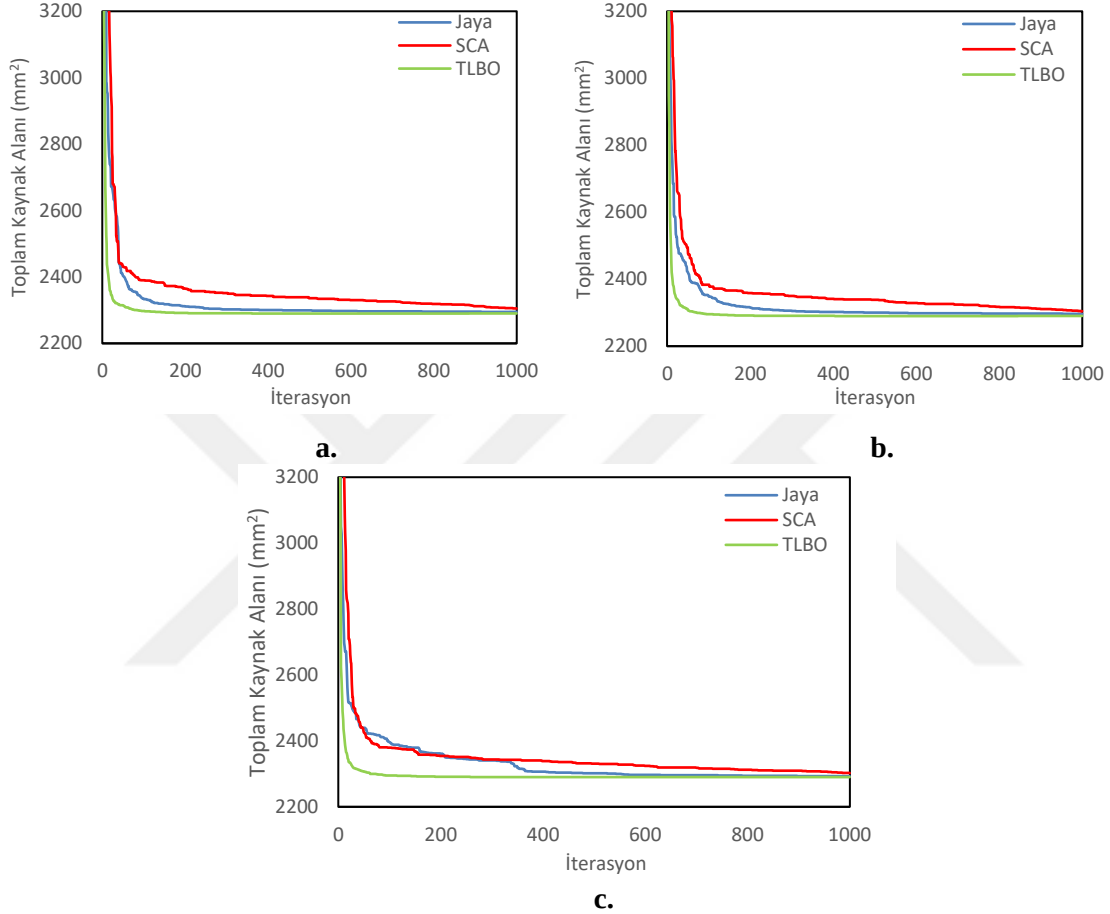
bulunmuştur. Bütün algoritmaların bulduğu en iyi sonuçlar için başlık kaynak grubunun kapasite kullanımları (GS-1) yaklaşık %100 gövde bileşke kapasite kullanımları (GS-4) ise yaklaşık %81'dir. Gövde bileşke gerilmesini arttıran gövdeye gelen kesme gerilmesinden ziyade gövde kaynağının üst ucunda oluşan gerilmedir. İlgili gerilme için kapasite kullanım oranı (GS-2) yaklaşık %78'dir. SM-2A için her biri 1000 iterasyonluk 10 tekrarlı çalıştırmada toplam kaynak alanları esas alınarak oluşturulmuş algoritma karşılaştırma parametreleri Çizelge 4.6'da verilmiştir.

Çizelge 4.6 SM-2A için algoritma karşılaştırma parametreleri.

<i>n</i>	Durum	Jaya	SCA	TLBO
250	En İyi (mm ²)	2290,66	2298,23	2290,19
	Ortalama (mm ²)	2294,84	2305,16	2290,19
	En Kötü (mm ²)	2322,69	2311,21	2290,20
	Standart Sapma	9,86	3,59	0,00
	İşlem Süresi (sn)	58,52	57,56	137,41
500	En İyi (mm ²)	2291,42	2298,95	2290,19
	Ortalama (mm ²)	2296,70	2304,51	2290,19
	En Kötü (mm ²)	2337,66	2309,44	2309,44
	Standart Sapma	14,40	2,99	0,00
	İşlem Süresi (sn)	91,94	97,67	250,86
1000	En İyi (mm ²)	2291,05	2299,11	2290,19
	Ortalama (mm ²)	2292,79	2302,36	2290,19
	En Kötü (mm ²)	2295,91	2308,87	2290,20
	Standart Sapma	1,30	3,06	0,00
	İşlem Süresi (sn)	178,97	181,51	479,73

Karşılaştırma parametreleri incelendiğinde, TLBO algoritması her popülasyon eleman sayısı için sıfır standart sapma ile en başarılı sonuçları bulduğu görülmektedir. Bu durum TLBO algoritmasını açık ara öne çıkarmaktadır. Jaya algoritması ve SCA kıyaslandığında $n=250$ ve $n=500$ için Jaya algoritması daha yüksek standart sapma değerlerine sahip olsa dahi en iyi değerler ve ortalama değerlerde SCA'dan daha başarılı sonuçlar vermiştir. Buna karşın en kötü değerlerde SCA'dan daha başarısız sonuçlar bulmuştur. Jaya algoritması $n=1000$ için daha küçük bir standart sapmaya ulaşarak bütün değerlerde SCA'dan daha başarılı sonuçlar elde etmiştir. İşlem süreleri bakımında Jaya ve SCA birbirine yakınken özdeş çalıştırma parametreleri için TLBO diğer algoritmaların iki katından daha fazla zamana ihtiyaç duymuştur. Buna karşın yakın sürelerde tamamlanan analizlerin sonuçları incelenecek olursa ($n=250$ için TLBO, $n=1000$ için Jaya ve $n=1000$ için SCA), TLBO ile daha az zamanda elde edilen sonuçların diğer algoritmalarından daha başarılı olduğu görülmektedir. Çizelge 4.6'da verilen çalıştırma koşulları için iterasyon –

toplam kaynak alanı grafikleri aşağıda verilmiştir. Grafiklerde her bir çalıştırma koşulu için her biri 1000 iterasyonluk 10 çalıştırmanın ortalama değerleri gösterilmektedir. Şekil 4.3.a $n=250$ eleman için, Şekil 4.3.b $n=500$ eleman için, Şekil 4.3.c ise $n=1000$ eleman için çizilmiştir.



Şekil 4.3 SM-2A için 10 çalıştırmanın ortalama değerleri. **a.** $n=250$, **b.** $n=500$, **c.** $n=1000$.

İlgili grafikler özelinde SM-2A modeli için algoritma performansları incelendiğinde, her bir durum için TLBO algoritmasının global optimuma ilk ulaşan algoritma olduğu görülmektedir. Jaya algoritması iterasyon ilerledikçe TLBO ile elde edilen sonuçlara yaklaşmaktadır. SCA ile elde edilen sonuçlar da oldukça kabul edilebilir durumdadır ancak TLBO ve Jaya algoritmasının bulduğu sonuçları tam olarak bulamamıştır. $n=1000$ için iterasyonun erken aşmalarında SCA ile Jaya algoritmasının performansları benzer olsa da ilerleyen iterasyonlarda Jaya algoritması süreçten pozitif ayrılmış ve TLBO algoritmasına yaklaşarak daha optimum kaynak alanları bulmuştur. Artan popülasyon elemanı sayısı ile algoritmaların bulduğu ortalama değerlerin birbirine daha erken

yaklaştıkları gözlemlenmiştir. SCA'nın diğer algoritmalarla elde edilen sonucu tam olarak bulamamasının, parametrelili bir algoritma olması ve hassas parametre ayarlamasına ihtiyaç duyması gibi sebeplerden kaynaklandığı düşünülmektedir.

SM-2B (normal kuvvet etkin durumdaki YDKT tasarım prensibi ile optimizasyon) için elde edilen sonuçlar Çizelge 4.7'de verilmiştir. İlgili veriler her bir satırda açıklanan çalıştırma koşulları için her biri 1000 iterasyonluk 10 çalıştırmanın neticesinde elde edilmiştir.

Çizelge 4.7 SM-2B için en iyi sonuç değerleri.

	<i>n</i>	<i>a</i> ₁ (mm)	<i>a</i> ₂ (mm)	<i>a</i> ₃ (mm)	<i>L</i> ₁ (mm)	<i>L</i> ₂ (mm)	<i>L</i> ₃ (mm)	<i>A</i> _w (mm ²)	GS- 1	GS- 2	GS- 3	GS- 4
<i>Jaya</i>	250	4,00	4,00	4,00	102,88	48,04	48,01	1976,18	1,00	0,74	0,17	0,76
	500	4,00	4,00	4,00	102,70	48,09	48,05	1976,44	1,00	0,74	0,17	0,76
	1000	4,01	4,00	4,00	102,38	48,24	48,02	1977,21	1,00	0,74	0,17	0,76
<i>SCA</i>	250	4,03	4,02	4,02	99,77	49,03	48,43	1981,60	1,00	0,74	0,17	0,76
	500	4,07	4,00	4,00	96,46	50,38	49,08	1984,37	1,00	0,74	0,17	0,76
	1000	4,01	4,02	4,00	99,15	49,47	48,79	1981,86	1,00	0,74	0,17	0,76
<i>TLBO</i>	250	4,00	4,00	4,00	103,00	48,00	48,00	1975,99	1,00	0,74	0,17	0,76
	500	4,00	4,00	4,00	103,00	48,00	48,00	1975,99	1,00	0,74	0,17	0,76
	1000	4,00	4,00	4,00	103,00	48,00	48,00	1975,99	1,00	0,74	0,17	0,76

Çizelge 4.7'de görüldüğü gibi, toplam kaynak alanı (*A_w*) için elde edilen en iyi sonuç 1975,99 mm²'dir. Bu sonuca TLBO algoritmasının *n*=250, *n*=500 ve *n*=1000 popülasyon elemanı için yapılan analizlerin tamamında ulaşılmıştır. En iyi değer için başlık dış, başlık iç ve gövde kaynak gruplarının kalınlıkları sırasıyla *a*₁= *a*₂= *a*₃=4,00 mm; kaynak uzunlukları ise sırasıyla *L*₁=103,00, *L*₂= *L*₃=48,03 mm olarak bulunmuştur. Jaya algoritması ile elde edilen minimum kaynak alanı *n*=250 için 1976,18 mm² olarak bulunmuştur. Bu sonuç için başlık dış, başlık iç ve gövde kaynak gruplarının kalınlıkları sırasıyla *a*₁= *a*₂= *a*₃=4,00 mm; kaynak uzunlukları ise sırasıyla *L*₁=102,88, *L*₂=48,04, *L*₃=48,01 mm olarak bulunmuştur. SCA ile elde edilen en iyi sonuç *n*=250 için 1981,60 mm²'dir. Bu sonuç için başlık dış, başlık iç ve gövde kaynak gruplarının kalınlıkları sırasıyla *a*₁= 4.03 *a*₂= *a*₃=4.02 mm; kaynak uzunlukları ise sırasıyla *L*₁=99,77, *L*₂=49,03, *L*₃=48,43 mm olarak bulunmuştur. Bütün çalıştırma durumları için başlık dış kaynak grubunun dış ucuna gelen gerilmeler kapasiteyi neredeyse %100 oranında kullanmaktadır (GS-1). Gövde bileşke gerilme kapasitesinin (GS-4) ise %76'sı kullanılmaktadır. Bu

kapasite kullanımı gövde kesmeden değil gövde kaynak grubunun üst ucuna gelen gerilmelerden (GS-2) kaynaklanmaktadır.

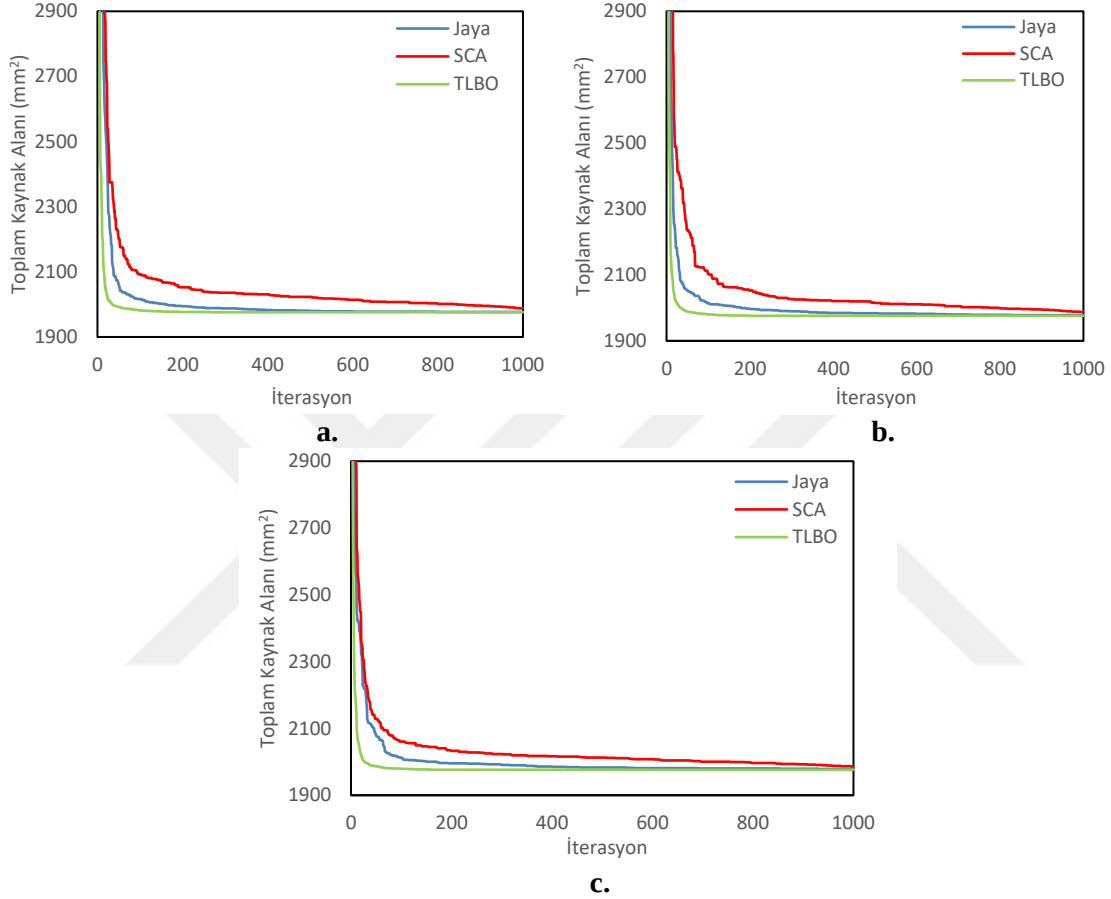
Çizelge 4.8 SM-2B için algoritma karşılaştırma parametreleri.

<i>n</i>	Durum	Jaya	SCA	TLBO
250	En İyi (mm ²)	1976,18	1981,60	1975,99
	Ortalama (mm ²)	1976,77	1987,80	1975,99
	En Kötü (mm ²)	1978,04	1996,47	1975,99
	Standart Sapma	0,48	4,89	0,00
	İşlem Süresi (sn)	57,51	56,89	135,71
500	En İyi (mm ²)	1976,44	1984,37	1975,99
	Ortalama (mm ²)	1977,42	1987,35	1975,99
	En Kötü (mm ²)	1979,62	1991,01	1975,99
	Standart Sapma	1,20	2,38	0,00
	İşlem Süresi (sn)	98,36	94,82	278,97
1000	En İyi (mm ²)	1977,21	1981,86	1975,99
	Ortalama (mm ²)	1978,37	1985,94	1975,99
	En Kötü (mm ²)	1982,47	1990,98	1975,99
	Standart Sapma	1,60	2,90	0,00
	İşlem Süresi (sn)	181,89	182,18	487,40

SM-2B için her biri 1000 iterasyonluk 10 tekrarlı çalıştırmada toplam kaynak alanları esas alınarak oluşturulmuş algoritma karşılaştırma parametreleri Çizelge 4.8’de verilmiştir. Parametreler incelendiğinde, SM-2A’ya benzer şekilde TLBO algoritmasının her popülasyon eleman sayısı için sıfır standart sapma ile en başarılı sonuçları bulduğu görülmektedir. Bu durum TLBO algoritmasını açık ara öne çıkarmaktadır. Jaya algoritması ve SCA kıyaslandığında SM-2A’nın aksine tüm eleman sayıları için karşılaştırma parametrelerinin tamamında Jaya algoritması daha başarılı bulunmuştur. Bundaki en büyük etkenin Jaya algoritmasının SCA’ya göre daha düşük standart sapma değerleri bulması olduğu düşünülmektedir. Bu sayısal model için Jaya algoritması SCA’dan daha kararlı çalışmıştır.

İşlem süreleri bakımında Jaya ve SCA birbirine yakınken, özdeş çalıştırma parametreleri için TLBO diğer algoritmaların iki katından daha fazla zamana ihtiyaç duymuştur. Yakın sürelerde tamamlanan analizlerin sonuçları incelenecek olursa ($n=250$ için TLBO, $n=1000$ için Jaya ve $n=1000$ için SCA), TLBO ile daha az zamanda daha az eleman ile elde edilen sonuçların, diğer algoritmalarla daha çok zamanda daha fazla elemanla elde edilen sonuçlardan daha başarılı olduğu görülmektedir.

Çizelge 4.8’de verilen çalıştırma koşulları için iterasyon – toplam kaynak alanı grafikleri aşağıda verilmiştir. Grafiklerde her bir çalıştırma koşulu için her biri 1000 iterasyonluk 10 çalıştırmanın ortalama değerleri gösterilmektedir. Şekil 4.4.a $n=250$ eleman için, Şekil 4.4.b $n=500$ eleman için, Şekil 4.4.c ise $n=1000$ eleman için çizilmiştir.



Şekil 4.4 SM-2B için 10 çalıştırmanın ortalama değerleri. **a.** $n=250$, **b.** $n=500$, **c.** $n=1000$.

SM-2B için grafikler incelendiğinde, her bir popülasyon eleman sayısı için TLBO algoritmasının optimum sonuca ilk ulaşan algoritma olduğu görülmektedir. Jaya algoritmasının bulduğu ortalama sonuçlar toplam iterasyonun ancak yarısı tamamlandığında TLBO algoritmasına yaklaşabilmiştir. SCA ise tüm popülasyon eleman sayıları için neredeyse iterasyonlar tamamlanmak üzereyken diğer algoritmalarla yaklaşabilmiştir. Sonuca daha az iterasyonla ulaşma bakımından başarı sıralaması TLBO, Jaya ve SCA şeklinde olmuştur.

4.3 Sayısal Model – 3

Sayısal model – 3 kesme kuvvetinin etkin olduğu durumu ifade etmektedir. SM-3A (kesme kuvvetinin etkin olduğu GKT tasarım prensibi ile optimizasyon) için farklı popülasyon eleman sayıları (n) ile elde edilen en iyi sonuçlar Çizelge 4.9’da verilmiştir.

Çizelge 4.9 SM-3A için en iyi sonuç değerleri.

	n	a_1 (mm)	a_2 (mm)	a_3 (mm)	L_1 (mm)	L_2 (mm)	L_3 (mm)	A_w (mm ²)	GS- 1	GS- 2	GS- 3	GS- 4
Jaya	250	4,00	4,00	4,00	48,00	48,00	95,92	1919,36	0,51	0,16	0,99	1,00
	500	4,00	4,00	4,00	48,00	48,00	95,91	1919,36	0,51	0,16	0,99	1,00
	1000	4,02	4,00	4,01	48,43	48,05	95,98	1927,64	0,50	0,16	0,99	1,00
SCA	250	4,01	4,00	4,01	48,06	48,10	95,84	1924,37	0,51	0,16	0,99	1,00
	500	4,00	4,00	4,14	48,10	48,06	93,27	1926,50	0,51	0,15	0,99	1,00
	1000	4,01	4,00	4,01	48,15	48,05	95,85	1923,27	0,51	0,16	0,99	1,00
TLBO	250	4,00	4,00	4,00	48,00	48,00	95,92	1919,34	0,51	0,16	0,99	1,00
	500	4,00	4,00	4,00	48,00	48,00	95,92	1919,34	0,51	0,16	0,99	1,00
	1000	4,00	4,00	4,00	48,00	48,00	95,92	1919,34	0,51	0,16	0,99	1,00

Çizelge 4.9’de görüldüğü gibi, toplam kaynak alanı (A_w) için elde edilen en iyi sonuç 1919,34 mm²’dir. Bu sonuca TLBO algoritmasının $n=250$, $n=500$ ve $n=1000$ popülasyon elemanı için yapılan analizlerin tamamında ulaşılmıştır. En iyi değer için başlık dış, başlık iç ve gövde kaynak gruplarının kalınlıkları sırasıyla $a_1= a_2= a_3=4,00$ mm; kaynak uzunlukları ise sırasıyla $L_1= L_2=48,00$, $L_3=95,92$ mm olarak bulunmuştur. Jaya algoritması ile elde edilen minimum kaynak alanı $n=250$ ve $n=500$ için 1919,36 mm² olarak bulunmuştur. Bu sonuç için başlık dış, başlık iç ve gövde kaynak gruplarının kalınlıkları sırasıyla $a_1= a_2= a_3=4,00$ mm; kaynak uzunlukları ise sırasıyla $L_1= L_2=48,00$, $L_3=95,92$ mm olarak bulunmuştur. SCA ile elde edilen en iyi sonuç $n=1000$ için 1923,27 mm²’dir. Bu sonuç için başlık dış, başlık iç ve gövde kaynak gruplarının kalınlıkları sırasıyla $a_1= a_3=4,01$ ve $a_2= 4,00$ mm; kaynak uzunlukları ise sırasıyla $L_1=48,15$, $L_2=48,05$, $L_3=95,85$ mm olarak bulunmuştur. Bütün çalıştırma durumları için başlık kaynak grubunun dış ucuna gelen gerilmeler kapasiteyi yaklaşık %51 oranında kullanmaktadır (GS-1). Gövde bileşke gerilme kapasitesinin (GS-4) ise neredeyse %100’ü kullanılmaktadır. Bu kapasite kullanımı gövde kaynak grubunun üst ucuna gelen gerilmelerden değil gövde kaynak grubunda oluşan kesme gerilmesinden kaynaklanmaktadır. Gövde kesme kapasitesinin yaklaşık %99’u kullanılmaktadır.

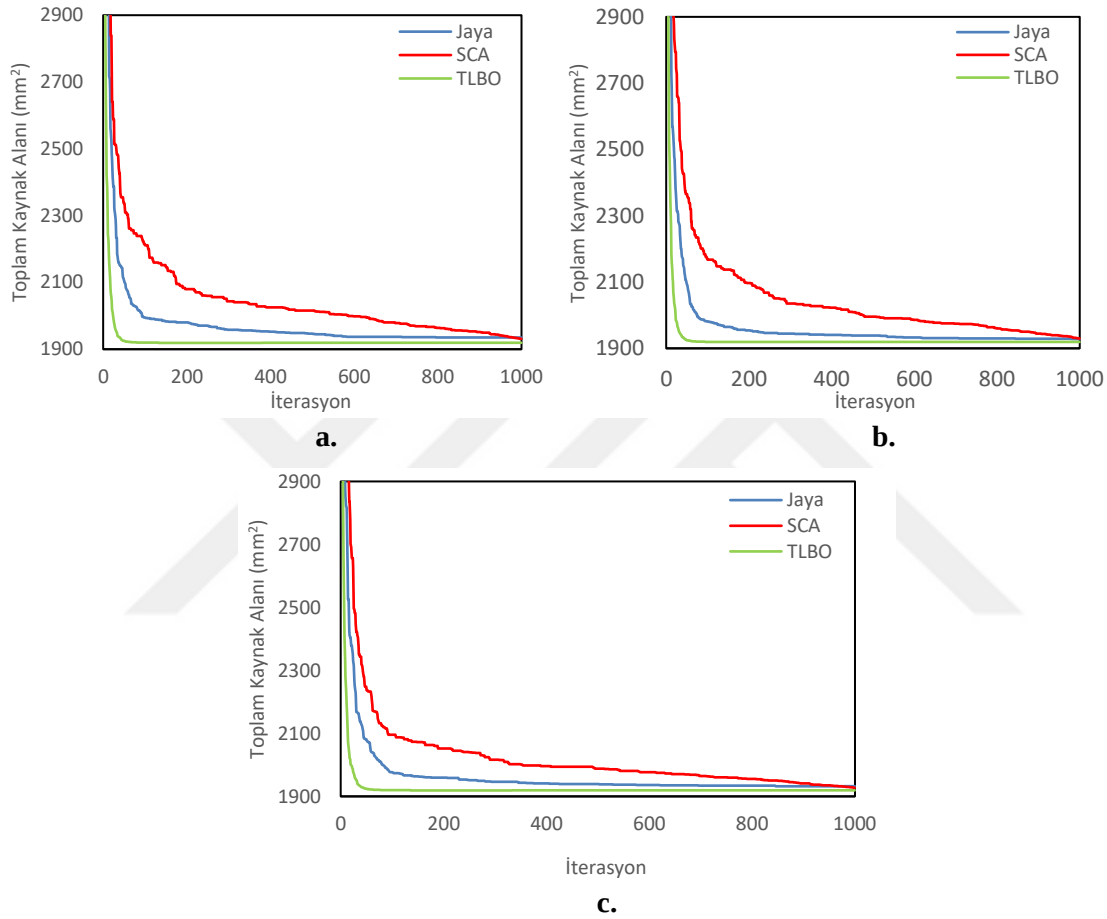
SM-3A için her biri 1000 iterasyonluk 10 tekrarlı çalıştırmada toplam kaynak alanları esas alınarak oluşturulmuş algoritma karşılaştırma parametreleri Çizelge 4.10'da verilmiştir. Buna göre, TLBO algoritmasının her popülasyon eleman sayısı için sıfır standart sapma ile en başarılı sonuçları bulduğu görülmektedir. Bu durum TLBO algoritmasını açık ara öne çıkarmaktadır.

Çizelge 4.10 SM-3A için algoritma karşılaştırma parametreleri.

<i>n</i>	Durum	Jaya	SCA	TLBO
250	En İyi (mm ²)	1919,36	1924,37	1919,34
	Ortalama (mm ²)	1935,08	1931,13	1919,34
	En Kötü (mm ²)	1961,93	1938,78	1919,34
	Standart Sapma	15,10	3,93	0,00
	İşlem Süresi (sn)	55,91	56,32	138,22
500	En İyi (mm ²)	1919,36	1926,50	1919,34
	Ortalama (mm ²)	1928,09	1929,47	1919,34
	En Kötü (mm ²)	1940,01	1934,81	1919,34
	Standart Sapma	7,60	2,71	0,00
	İşlem Süresi (sn)	107,67	118,71	330,81
1000	En İyi (mm ²)	1927,64	1923,27	1919,34
	Ortalama (mm ²)	1932,35	1927,15	1919,34
	En Kötü (mm ²)	1941,08	1932,00	1919,34
	Standart Sapma	4,51	2,70	0,00
	İşlem Süresi (sn)	175,20	183,24	484,65

Jaya algoritması ve SCA kıyaslandığında her eleman sayısı için Jaya algoritmasının standart sapmaları daha büyüktür. Bunun anlamı, her bir çalıştırma için Jaya algoritması ile elde edilen sonuçların görece geniş bir aralığa yayılmasıdır. Buradan hareketle SCA bu model için Jaya algoritmasından daha kararlı çalışmıştır çıkarımı yapılabilir. Bu durum $n=250$ ve $n=500$ için en iyi sonuçlarda Jaya algoritması önde olsa da ortalama sonuçlarda SCA'nın gerisinde kalmasına sebep olmuştur. $n=1000$ eleman için ise SCA tüm parametreler için Jaya algoritmasından daha başarılı sonuçlar vermiştir. İşlem süreleri bakımından Jaya ve SCA yakınken TLBO ise bunların iki katından daha fazla sürede analizi tamamlamıştır. TLBO algoritmasının analiz süresinin diğer algoritmaların analiz süresinin iki katından fazla olmasının sebebi daha sofistike bir yapıya sahip olmasıyla açıklanabilir zira SCA ve Jaya algoritması her bir iterasyon için popülasyona tek geliştirmede bulunuyorken TLBO bir iterasyonda iki defa geliştirmede bulunmaktadır. Yakın sürelerde tamamlanan analizlerin sonuçları incelenecek olursa ($n=250$ için TLBO, $n=1000$ için Jaya ve $n=1000$ için SCA), TLBO ile daha az zamanda daha az eleman ile elde edilen sonuçların, diğer algoritmalarla daha çok zamanda daha

fazla elemanla elde edilen sonuçlardan daha başarılı olduğu görülmektedir. Çizelge 4.10’da verilen çalıştırma koşulları için iterasyon – toplam kaynak alanı grafikleri aşağıda verilmiştir. Grafiklerde her bir çalıştırma koşulu için her biri 1000 iterasyonluk 10 çalıştırmanın ortalama değerleri gösterilmektedir. Şekil 4.5.a $n=250$ eleman için, Şekil 4.5.b $n=500$ eleman için, Şekil 4.5.c ise $n=1000$ eleman için çizilmiştir.



Şekil 4.5 SM-3A için 10 çalıştırmanın ortalama değerleri. **a.** $n=250$, **b.** $n=500$, **c.** $n=1000$.

İlgili grafiklerde, her popülasyon eleman sayısı için TLBO algoritmasının global optimuma ilk ulaşan algoritma olduğu görülmektedir. İterasyon ilerledikçe önce Jaya algoritması daha sonra ise SCA, TLBO ile bulunan ortalama değerlere yaklaşabilmiştir. SCA ile bulunan ortalama değerler iterasyonun her safhası için diğer algoritmalar ile bulunan değerden daha yukarıda yani daha kötü bir değerde kalmıştır. Bu durumun SCA’nın hassas parametre ayarlaması ihtiyacından kaynaklandığı düşünülmektedir. Daha hassas parametre ayarlamaları ile daha optimum sonuçlar elde edilme ihtimali her zaman bulunmaktadır. TLBO algoritması ise çok daha az iterasyon sayısı ile çok daha

başarılı sonuçlara ulaşmıştır. Bunun sebebi olarak ise TLBO'nun sofistike yapısı ve diğer algoritmalar bir iterasyon için popülasyonu bir defa geliştiriyorken TLBO algoritmasının iki defa gelişime uğratmasının etkili olduğu düşünülmektedir.

SM-3B (kesme kuvveti etkin durumda YDKT tasarım prensibi ile yapılan optimizasyon) için $n=250$, $n=500$ ve $n=1000$ popülasyon elemanı ile bulunan en iyi sonuç değerleri Çizelge 4.11'de verildiği gibidir.

Çizelge 4.11 SM-3B için en iyi sonuç değerleri.

	n	a_1 (mm)	a_2 (mm)	a_3 (mm)	L_1 (mm)	L_2 (mm)	L_3 (mm)	A_w (mm ²)	GS- 1	GS- 2	GS- 3	GS- 4
Jaya	250	4,00	4,00	4,00	48,00	48,00	77,98	1775,90	0,47	0,12	0,99	1,00
	500	4,00	4,00	4,00	48,00	48,00	77,99	1776,10	0,47	0,12	0,99	1,00
	1000	4,00	4,00	4,00	48,00	48,00	77,99	1776,03	0,47	0,12	0,99	1,00
SCA	250	4,01	4,01	4,03	48,09	48,08	77,50	1782,00	0,47	0,12	0,99	1,00
	500	4,02	4,00	4,01	48,17	48,05	77,82	1780,94	0,47	0,12	0,99	1,00
	1000	4,00	4,00	4,03	48,25	48,01	77,69	1780,83	0,47	0,12	0,99	1,00
TLBO	250	4,00	4,00	4,00	48,00	48,00	77,99	1775,88	0,47	0,12	0,99	1,00
	500	4,00	4,00	4,00	48,00	48,00	77,99	1775,88	0,47	0,12	0,99	1,00
	1000	4,00	4,00	4,00	48,00	48,00	77,99	1775,88	0,47	0,12	0,99	1,00

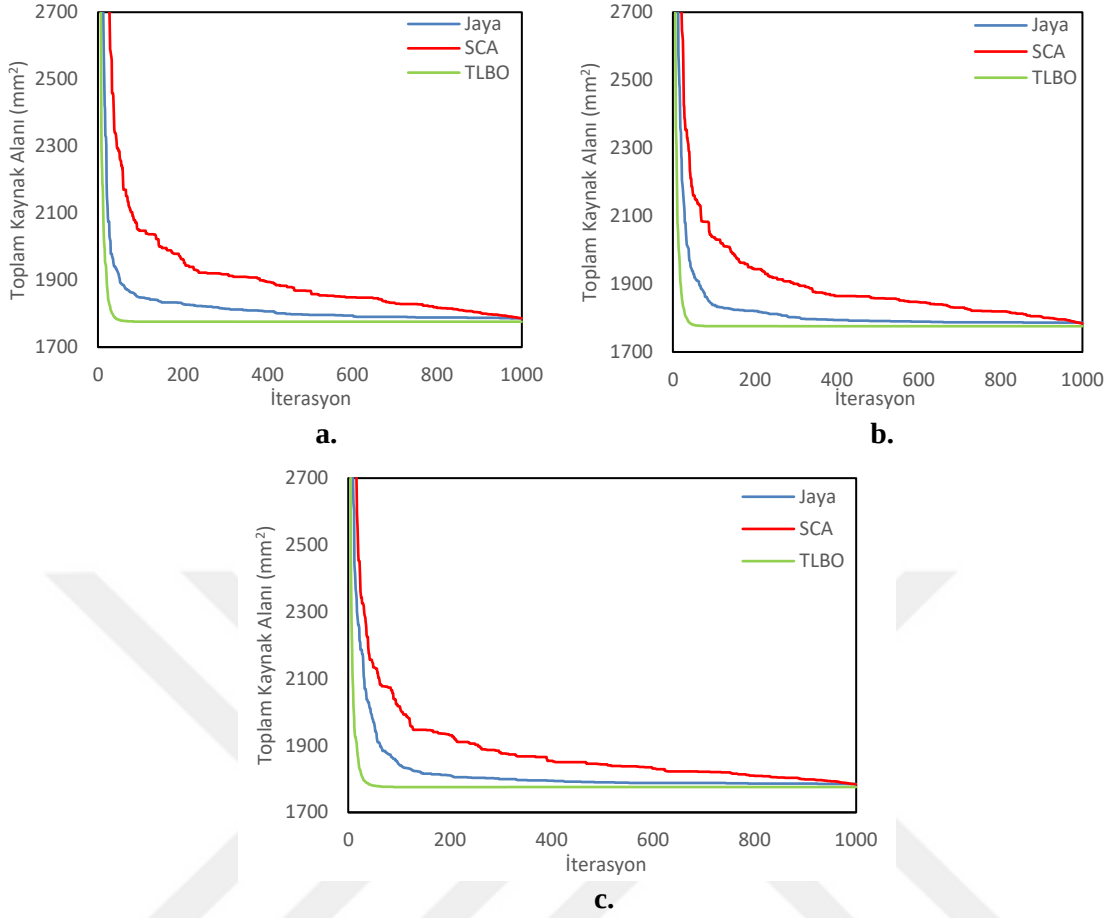
Çizelge 4.11'de görüldüğü gibi, toplam kaynak alanı (A_w) için elde edilen en iyi sonuç 1775,88 mm²'dir. Bu sonuca TLBO algoritmasının $n=250$, $n=500$ ve $n=1000$ popülasyon elemanı için yapılan analizlerin tamamında ulaşılmıştır. En iyi değer için başlık dış, başlık iç ve gövde kaynak gruplarının kalınlıkları sırasıyla $a_1 = a_2 = a_3 = 4,00$ mm; kaynak uzunlukları ise sırasıyla $L_1 = L_2 = 48,00$, $L_3 = 77,99$ mm olarak bulunmuştur. Jaya algoritması ile elde edilen minimum kaynak alanı $n=250$ için 1775,90 mm² olarak bulunmuştur. Bu sonuç için kaynak kalınlıkları ve kaynak uzunlukları TLBO ile bulunan değerlerden mikrometreler mertebesinde farklılıklar içermektedir. SCA ile elde edilen en iyi sonuç $n=1000$ için 1980,83 mm²'dir. Bu sonuç için başlık dış, başlık iç ve gövde kaynak gruplarının kalınlıkları sırasıyla $a_1 = a_2 = 4,00$ ve $a_3 = 4,03$ mm; kaynak uzunlukları ise sırasıyla $L_1 = 48,25$, $L_2 = 48,01$, $L_3 = 77,69$ mm olarak bulunmuştur. Bütün çalıştırma durumları için başlık dış kaynak grubunun dış ucuna gelen gerilmeler kapasiteyi yaklaşık %47 oranında kullanmaktadır (GS-1). Gövde bileşke gerilme kapasitesinin (GS-4) ise neredeyse %100'ü kullanılmaktadır. Bu kapasite kullanımı gövde kaynak grubunun üst ucuna gelen gerilmelerden değil gövde kaynak grubunda oluşan kesme gerilmesinden

kaynaklanmaktadır. Gövde kesme kapasitesinin yaklaşık %99'u kullanılmaktadır. SM-3B için her biri 1000 iterasyonluk 10 tekrarlı çalıştırmada toplam kaynak alanları esas alınarak oluşturulmuş algoritma karşılaştırma parametreleri Çizelge 4.12'de verilmiştir. Buna göre, TLBO algoritmasının her durumda yaklaşık sıfır standart sapma ile en başarılı sonuçları bulduğu görülmektedir. Jaya algoritması ve SCA kıyaslandığında en iyi değerler bakımından her durumda Jaya algoritması daha başarılıdır. Her eleman sayısı için Jaya algoritmasının standart sapmaları daha büyüktür. Ortalamalar bakımından Jaya ile SCA'nın performansı birbirine yakındır. En kötü sonuçlara bakıldığında Jaya algoritmasının daha başarısız olduğu görülebilir. İşlem süreleri bakımından, Jaya algoritması az bir farkla SCA'dan daha kısa sürede analizleri tamamlanmıştır. TLBO'nun analizlerini tamamlaması ise diğer algoritmaların iki katından daha fazla zaman almıştır. Yakın sürelerde tamamlanan analizlerin sonuçları incelenecek olursa ($n=250$ için TLBO, $n=1000$ için Jaya ve $n=1000$ için SCA), TLBO ile daha az zamanda daha az eleman ile elde edilen sonuçların, diğer algoritmalarla daha çok zamanda daha fazla elemanla elde edilen sonuçlardan daha başarılı olduğu görülmektedir.

Çizelge 4.12 SM-3B için algoritma karşılaştırma parametreleri.

<i>n</i>	Durum	Jaya	SCA	TLBO
250	En İyi (mm ²)	1775,90	1782,00	1775,88
	Ortalama (mm ²)	1785,94	1786,08	1775,88
	En Kötü (mm ²)	1806,49	1789,52	1775,88
	Standart Sapma	9,64	2,59	0,00
	İşlem Süresi (sn)	54,58	61,73	139,97
500	En İyi (mm ²)	1776,10	1780,94	1775,88
	Ortalama (mm ²)	1786,57	1783,87	1775,88
	En Kötü (mm ²)	1797,34	1787,17	1775,88
	Standart Sapma	6,80	2,11	0,00
	İşlem Süresi (sn)	91,90	95,09	249,28
1000	En İyi (mm ²)	1776,03	1780,83	1775,88
	Ortalama (mm ²)	1784,98	1783,69	1775,88
	En Kötü (mm ²)	1793,10	1786,69	1775,88
	Standart Sapma	5,85	1,85	0,00
	İşlem Süresi (sn)	177,70	181,72	478,73

Çizelge 4.12'de 10 çalıştırma için en iyi sonuçları verilen çalıştırma koşulları için iterasyon – toplam kaynak alanı grafikleri aşağıda verilmiştir. Grafiklerde her bir çalıştırma koşulu için her biri 1000 iterasyonluk 10 çalıştırmanın ortalama değerleri gösterilmektedir. Şekil 4.6.a $n=250$ popülasyon elemanı için, Şekil 4.6.b $n=500$ popülasyon elemanı için, Şekil 4.6.c ise $n=1000$ popülasyon elemanı için çizilmiştir.



Şekil 4.6 SM-3B için 10 çalıştırmanın ortalama değerleri. **a.** $n=250$, **b.** $n=500$, **c.** $n=1000$.

SM-3B modeli için grafikler incelendiğinde, TLBO algoritmasının diğer algoritmalara göre ciddi bir farkla global optimuma çok erken ulaştığı görülmektedir. Bu farkın oluşmasında TLBO algoritmasının bir iterasyonda iki geliştirme yaparak ilerlemesinin ciddi payı olduğu düşünülmektedir. Popülasyon eleman sayısı (n) arttıkça Jaya algoritmasının TLBO algoritmasına daha erken yaklaşabildiği görülmektedir. SCA ise her popülasyon eleman sayısı için TLBO ve Jaya algoritmasının gerisinde kalmıştır. Bu farkın oluşmasında SCA'nın hassas parametre ayarlama ihtiyacının bulunmasının önemli bir rolü olduğu düşünülmektedir.

5. TARTIŞMA ve SONUÇ

Bu çalışmada; Jaya, TLBO ve Sinüs – Kosinüs algoritmaları ile çelik yapıların kaynaklı birleşimlerinin optimum tasarımlarının yapılması, ilgili metasezgisel algoritmaların sonuçlarının tanımlanan sayısal modeller ekseninde karşılaştırılarak hangi algoritmanın hangi spesifik konuda diğerlerinden daha avantajlı olduğunun belirlenmesi ve bu çalışmada sunulmuş olan Adil Ceza Yönteminin kısıtlı optimizasyon problemlerinin çözümü için kullanılabilirliğinin araştırılması hedeflenmiştir. Bu hedef doğrultusunda ÇYTHYE (2018) için tanımlanan gereklilikleri yerine getiren üç adet sayısal model oluşturularak yönetmelikte tanımlanan iki farklı tasarım prensibi için ve üç farklı popülasyon eleman sayısı için toplamda 54 adet analiz gerçekleştirilmiştir. Bu çalışma neticesinde ulaşılan sonuçlar şu şekildedir:

- a. Yapılan 54 farklı analizin sonuçlarından ilgili olanlar karşılaştırıldığında, bulunan optimum kaynak alanlarının son derece tutarlı olduğu görülmüştür. Buradan hareketle Jaya, TLBO ve Sinüs – Kosinüs algoritmalarının çelik yapıların kaynaklı birleşimlerinin optimum tasarımı için kullanılabilir ve hızlı çözümler sunan başarılı algoritmalar olduğu anlaşılmaktadır.
- b. Analizler algoritma bazında karşılaştırılacak olursa, her bir karşılaştırma parametresi için TLBO algoritması Jaya ve Sinüs-Kosinüs algoritmasından daha başarılı sonuçlar vermiştir. TLBO'nun özdeş çalışma koşulları altında optimizasyon için daha çok zaman harcamasından başka bir dezavantajı tespit edilememiştir. Çalıştırma parametreleri özdeş seçilmediğinde, TLBO algoritmasının her durum için daha az işlem adımıyla ve daha kısa sürede Jaya ve SCA'dan daha iyi sonuçlar bulabildiği gözlenmiştir. TLBO algoritmasının, çalıştırmalar arası neredeyse sıfıra yakın standart sapma değerleri elde edebilecek kadar kararlı ve her durumda en iyi sonucu verecek kadar başarılı sonuçlar bulmasında, bir iterasyon için iki geliştirme yapan çift fazlı sürecin büyük katkısı olduğu düşünülmektedir.

- c. Jaya ve Sinüs – Kosinüs algoritmaları karşılaştırıldığında ortalama performansları birbirine yakın bulunmuştur. Bu iki algoritma ile yapılan analizler birbirlerine yakın sürelerde tamamlanmıştır. Analizlerin büyük çoğunluğunda Jaya algoritması en iyi sonucu bulmakta daha başarılıyken SCA ise daha küçük standart sapmalar vererek daha kararlı sonuçlar bulunmuştur. Bu bağlamda, Jaya algoritmasının global optimumu bulmakta, SCA'nın ise her bir çalıştırma için yakın ve kendi içerisinde tutarlı sonuçlar tespit etmekte daha başarılı olduğu söylenebilir. Sonuçlar değerlendiriliyorken SCA'nın parametrelili bir algoritma olduğu ve farklı parametre ayarlamaları ile daha iyi ya da daha kötü sonuçlar alınabilme ihtimalinin bulunduğu unutulmamalıdır.
- d. SCA'nın performansının “ a ” parametresindeki değişikliklere çok duyarlı olduğu tespit edilmiştir. Optimizasyon problemlerinin çözümü için SCA tercih edildiğinde, hassas parametre ayarlamaları için denemeler yapılması gerektiği anlaşılmaktadır. Bu durumun SCA'nın pratikliğini ve bir problemin çözümü için tek başına kullanılabilirliğini sınırladığı görülmüştür.
- e. Bulgular kısmında verilen sonuçların tamamı algoritma kısıtlarını, dolayısıyla ÇYTHYE (2018) için tanımlanan sınırlayıcıları sağlamaktadır. İlgili kısıtların sağlanabilmesi için bu çalışmada sunulan “Adil Ceza Yöntemi” başarılı bulunmuştur. Yönteminin kısıtlı optimizasyon problemlerinin çözümünde cezalandırma işlevi sağlamak üzere kullanılabilir olduğu anlaşılmıştır.

6. KAYNAKLAR

- Alberdi R, Murren P, Khandelwal K, 2015, Connection Topology Optimization of Steel Moment Frames Using Metaheuristic Algorithms, *Engineering Structures*, 100, 276-292.
- Anders A, 2003, Tracking Down the Origin of Arc Plasma Science-II. Early Continuous Discharges, *IEEE Transactions on Plasma Science*, 31, 1060–1069.
- Anonim, 1994, *The Procedure Handbook of Arc Welding*, Lincoln Electric Press, 750p, Cleveland.
- Artar M, Daloglu A T, 2020, A Research on Optimum Designs of Steel Frames Including Soil Effects or Semi Rigid Supports Using Jaya Algorithm, *Structural Engineering and Mechanics*, 73, 153-165.
- Ayrton H, 1902, *The Electric Arc*, D. Van Nostrand Co., 479p, New York.
- Beyer H G, Schwefel H P, 2002, Evolution Strategies a Comprehensive Introduction, *Natural Computing*, 1, 3–52.
- Calado L, Mele E, 2000, Cyclic Tests on Bolted and Welded Beam-to-Column Connections, *ISET Journal of Earthquake Technology*, 37, 65–88.
- Carry H B, Helzer S C, 2005, *Modern Welding Technology*, Pearson/Prentice Hall, 787p, Upper Saddle River.
- Castiglioni C A, Bernuzzi C, Calado L, Cyclic Behaviour of Steel Beam-to Column Joints with Concrete Slab, 3rd International Conference on the Behaviour Of Steel Structures in Seismic Areas (STESSA), 2000, August 21-24, Montreal.
- Christensen P W, Klarbring A, 2009, *An Introduction to Structural Optimization*, Springer, 201p, Netherlands.
- Coello C A C, 2000, Use of a Self-Adaptive Penalty Approach for Engineering Optimization Problems, *Computers in Industry*, 41, 113–127.
- Coello C A C, 2008, Solving Engineering Optimization Problems with the Simple Constrained Particle Swarm Optimizer, *Informatica*, 32, 319–326.

- Çelik Yapıların Tasarım, Hesap ve Yapımına Dair Esaslar, 2018, Resmî Gazete, 15 Şubat 2018, 30333.
- Daloğlu A, Armutçu M, 1998, Genetik Algoritma ile Düzlem Çelik Çerçevelerin Optimum Tasarımı, Teknik Dergi, 116, 1601-1615.
- Daloğlu A, Aydın Z, 1999, Kafes Sistemlerin Uygulamaya Yönelik Optimum Tasarımı, Pamukkale Üniversitesi Mühendislik Fakültesi Mühendislik Bilimleri Dergisi, 5, 951-957.
- David D C N, Stephen S E A, Ajoy J A, 2016, Cost Minimization of Welded Beam Design Problem Using PSO, SA, PS, GOLDBLICK, CUCKOO, FF, FP, ALO, GSA and MVO, International Journal of Applied Mathematics, 5, 1–14.
- Dede T, Ayvaz Y, 2013, Structural Optimization with Teaching Learning-Based Optimization Algorithm, Structural Engineering and Mechanics, 47, 495-511.
- Dede T, Bekiroğlu S, Ayvaz Y, 2011, Weight Minimization of Trusses with Genetic Algorithm, Applied Soft Computing, 11, 2565–2575.
- Demiral M F, 2020, A Parameters Analysis of Sine Cosine Algorithm on Travelling Salesman Problem, El-Cezerî Journal of Science and Engineering, 7, 526-535.
- Deren H, Uzgider E, Piroğlu F, 2008, Çelik Yapılar, Çağlayan Kitabevi, 983s, İstanbul.
- Dorigo M, Birattari M, Stutzle T, 2006, Ant Colony Optimization—Artificial Ants as a Computational Intelligence Technique, IEEE Computational Intelligence Magazine, 1, 28–39.
- Engelhardt M D, Sabol T A, 1998, Reinforcing of Steel Moment Connections with Cover Plates: Benefits and Limitations, Engineering Structures, 20, 510–520.
- Erol O K, Eksin I, 2006, A New Optimization Method: Big Bang– Big Crunch, Advances in Engineering Software, 37, 106–111.
- Ertürk İ, 1987, Gazaltı Kaynak Teknikleri, Küçük Sanayi İşletmelerinde Danışmanlık Hizmetleri Projesi, Türkiye Halk Bankası AŞ. Yayınları, 78s, Ankara.

- Gandomi A H, Yang X S, Alavi A H, 2013, Cuckoo Search Algorithm: A Metaheuristic Approach to Solve Structural Optimization Problems, *Engineering with Computers*, 29, 17–35.
- Gaylord E H, Gaylord, C N, Stallmeyer, J E, 1992, *Design of Steel Structures*, McGraw-Hill Education, 816p, New York
- Haftka R T, Gürdal Z, 1991, *Elements of Structural Optimization* 3rd ed., Kluwer Academic Publishers, 468p, London.
- Hatamlou A, 2013, Black hole: A New Heuristic Optimization Approach for Data Clustering, *Information Sciences*, 222, 175–184.
- Hayalioğlu M S, Değertekin S Ö, 2005, Minimum Cost Design of Steel Frames with Semi-Rigid Connections and Column Bases via Genetic Optimization, *Computers and Structures*, 83, 1849-1863.
- Heredotus, 1998, *The Histories*, Çev.: Waterfield R, Oxford University Press, 848p, Oxford.
- Holland J H, 1975, *Adaptation in Natural And Artificial Systems*, University of Michigan Press, 194p, Ann Arbor.
- Jaspard J P, Weynand K, 2017, *Design of Joints in Steel Structures Eurocode 3: Design of Steel Structures*, European Convention for Constructional Steelwork, 374p, Maia.
- Jin S, Ohmori H, Lee S, 2017, Optimal Design of Steel Structures Considering Welding Cost and Constructability of Beam-Column Connections, *Journal of Constructional Steel Research*, 135, 292-301.
- Kahraman N, Gülenç B, 2016, *Modern Kaynak Teknolojisi*, Epa-Mat Basım Yayın, 331s, Ankara.
- Kameshki E S, Saka M P, 2001, Optimum Design of Nonlinear Steel Frames with Semi-Rigid Connections Using a Genetic Algorithm, *Computers and Structures*, 79, 1593–1604.

- Karaboğa D, Baştürk B, 2007, A Powerful and Efficient Algorithm for Numerical Function Optimization: Artificial Bee Colony (ABC) Algorithm, *Journal of Global Optimization*, 39, 459–471.
- Kaveh A ve Talatahari S, 2012, A Hybrid CBO and PSO Algorithm for Optimal Design of Structures, *Structural Engineering and Mechanics*, 42, 783-797.
- Kaveh A, Khayatazad M, 2012, A New Meta-Heuristic Method: Ray Optimization, *Computers & Structures*, 112, 283–294.
- Kaveh A, Talatahari S, 2007, A Discrete Particle Swarm Ant Colony Optimization for Design of Steel Frames, *Asian Journal of Civil Engineering Building and Housing*, 9, 563-575.
- Kaveh A, Talatahari S, 2009, Particle Swarm Optimizer, Ant Colony Strategy and Harmony Search Scheme Hybridized For Optimization of Truss Structures, *Computers & Structures*, 87, 267-283.
- Kazakov N F, 1985, *Diffusion Bonding of Materials*, Pergamon Press, 304p, Oxford.
- Kennedy J, Eberhart R C, Particle Swarm Optimization, *Proceedings of IEEE International Conference on Neural Networks*, 1995, November 27-December 1, Perth, 1942–1948.
- Kirkpatrick S, Gelatt C D, Vecchi M P, 1983, Optimization by Simulated Annealing, *Science*, 220, 671–680.
- Koza J R, 1994, *Genetic Programming II, Automatic Discovery of Reusable Subprograms*, MIT Press, 746p, Cambridge.
- Lagaros N D, 2018, The Environmental and Economic Impact of Structural Optimization, *Structural and Multidisciplinary Optimization*, 58, 1751-1768.
- Lazarev P, 1999, Historical Essay on the 200 Years of the Development of Natural Sciences in Russia, *Advances in Physical Sciences*, 42, 1247–1257.
- Lee K S, Geem Z W, 2005, A New Meta-Heuristic Algorithm for Continuous Engineering Optimization: Harmony Search Theory and Practice, *Computer Methods in Applied Mechanics and Engineering*, 194, 3902–3933.

- Lui E M, Lopes A, 1997, Dynamic Analysis and Responce of Semi Rigid Frames, *Engineering Structures*, 19, 644-654.
- Mahin S A, Hamburger R O, Malley J O, An Integrated Program to Improve the Performance of Welded Steel Frame Buildings, 11th World Conference on Earthquake Engineering (WCEE), 1996, June 23-28, Acapulco.
- MATLAB, 2016, version 9.0.0 (R2016a), The MathWorks Inc., Natick, Massachusetts.
- Mele E, 2002, Moment Resisting Welded Connections: An Extensive Review of Design Practice and Experimental Research in USA, Japan And Europe, *Journal of Earthquake Engineering*, 6, 111–145.
- Mirjalili S, 2015, SCA: A Sine Cosine Algorithm for Solving Optimization Problems, *Knowledge-Based Systems*, 96, 120-133.
- Pham D T, Karaboga D, 2000, *Intelligent Optimisation Techniques*, Springer-Verlag, 302p, London.
- Ragsdell K M, Phillips D T, 1976, Optimal Design of a Class of Welded Structures Using Geometric Programming, *ASME Journal of Manufacturing Science and Engineering*, 98, 1021–1025.
- Rao R V, 2016a, Jaya: A Simple and New Optimization Algorithm for Solving Constrained and Unconstrained Optimization Problems, *International Journal of Industrial Engineering Computations*, 7, 19–34.
- Rao R V, 2016b, *Teaching Learning Based Optimization Algorithm and Its Engineering Applications*, Springer International, 267p, Switzerland.
- Rao R V, Savasani V J, Vakharia D P, 2011, Teaching–Learning-Based Optimization: A Novel Method for Constrained Mechanical Design Optimization Problems, *Computer-Aided Design*, 43, 303-315.
- Rao S S, 1996, *Engineering Optimization: Theory and Practice*, 4th ed., Wiley-Interscience Publications, 903p, New York.
- Saka M P, 1991, Optimum Design of Steel Frames with Stability Constraints, *Computers and Structures*, 41, 1365-1377.

- Saka M P, 2009, Optimum Design of Steel Sway Frames to BS5950 Using Harmony Search Algorithm, *Journal of Constructional Steel Research*, 65, 36–43.
- Saka M P, Çarbaş S, Aydoğdu I, Akin A, 2015, Use of Swarm Intelligence in Structural Steel Design Optimization, *Metaheuristics and Optimization in Civil Engineering*, 7, 43-73.
- Saka M P, Hasançebi O, Geem Z W, 2016, Metaheuristics in Structural Optimization and Discussions on Harmony Search Algorithm, Swarm and Evolutionary Computation, 28, 88–97.
- Schmit L A, Structural Design by Systematic Synthesis, 2nd Conference on Electronic Computation, 1960, September 8-9, Pittsburgh, 105–132.
- Schwartz M, 2011, Innovations in Materials Manufacturing, Fabrication and Environmental Safety, CRC Press, 300p, Boca Raton.
- Storn R, Price K, 1995, Differential Evolution—A Simple and Efficient Adaptive Scheme for Global Optimization Overcontinuous Spaces, Report No: TR-ICSI 95-012, 15p.
- Talbi E G, 2009, Metaheuristic: from Design to Implementation, 2nd Edition, Wiley, 624p, New Jersey.
- Tama Y S, 2009, Çelik Yapı Uygulamalarında Kullanılan Kaynak Yöntemleri, Üstünlükleri ve Sakıncalı Yönleri, Türk Yapısal Çelik Derneği, E-Makale.
- Tanaka A, Kato B, Kaneko H, Sakamoto S, Takahashi Y, Teraoka M, Seismic Damage of Steel Beam-to-Column Connections — Evaluation from Statical Aspects, 2nd International Conference on the Behaviour Of Steel Structures in Seismic Areas (STESSA), 1997, August 3-8, Kyoto.
- Tejani G G, Savsani V J, Patel V K, Savsani P V, 2018, Size, Shape, and Topology Optimization of Planar and Space Trusses Using Mutation-Based Improved Metaheuristics, *Journal of Computational Design and Engineering*, 5, 198–214.
- Toğan V, 2012, Design of Planar Steel Frames Using Teaching–Learning Based Optimization, *Engineering Structures*, 34, 225–232.

- Toğan V, Daloğlu A T, Karadeniz H, 2011, Optimization of Trusses Under Uncertainties with Harmony Search, Structural Engineering and Mechanics, 37, 543- 560.
- TS EN 1090-2, 2018, Çelik ve Alüminyum Yapı Uygulamaları- Bölüm 2: Çelik Yapılar İçin Teknik Gerekler, Türk Standartları Enstitüsü, Ankara.
- Weman K, 2007, Welding Processes Handbook 2nd ed., Woodhead Publishing, 280p, New York.
- Yang X S, 2010, Nature-Inspired Metaheuristic Algorithms, Second Edition, Luniver Press, 147p, Frome.
- Yildiz A R, Abderazek H, Mirjalili S, 2020, A Comparative Study of Recent Non-Traditional Methods for Mechanical Design, Optimization Archives of Computational Methods in Engineering, 27, 1031–1048.
- Yorgun C, Topkaya C, Vatansever C, 2017, Çelik Yapıların Tasarım, Hesap ve Yapım Esasları Yönetmeliği – 2016 Eğitim Notları, Çevre ve Şehircilik Bakanlığı, 472s, Ankara.

İnternet Kaynakları

- 1- <https://en.wikipedia.org/w/index.php?title=Welding&oldid=89525524> , 29.12.2020
- 2- <http://weldinghistory.org> , 29.12.2020
- 3- <http://www.weldinghistory.org/whfolder/folder/wh1900.html> , 29.12.2020