



## **Çelik Yapıların Kaynaklı Birleşimlerinin Metasezgisel Yöntemlerle Optimum Tasarımı**

### **Optimum Design of Welded Connections of Steel Structures Using Metaheuristic Methods**

**Soner Sezer<sup>1\*</sup>**, **Murat Hiçyılmaz<sup>2</sup>**

<sup>1</sup> Afyon Kocatepe Üniversitesi, Fen Bilimleri Enstitüsü, İnşaat Mühendisliği Anabilim Dalı, Afyonkarahisar, TÜRKİYE

<sup>2</sup> Afyon Kocatepe Üniversitesi, Mühendislik Fakültesi, İnşaat Mühendisliği Bölümü, Afyonkarahisar, TÜRKİYE

Sorumlu Yazar / Corresponding Author \*: [sonersezer@outlook.com.tr](mailto:sonersezer@outlook.com.tr)

Geliş Tarihi / Received: 01.05.2021

Kabul Tarihi / Accepted: 30.09.2021

Atıf şekli/How to cite: SEZER, S., HIÇYILMAZ, M.(2022).Çelik Yapıların Kaynaklı Birleşimlerinin Metasezgisel Yöntemlerle Optimum Tasarımı. DEUFMD, 24(70), 277-290.

Araştırma Makalesi/Research Article

DOI:10.21205/deufmd.2022247025

#### **Öz**

Bu çalışmada; Jaya, TLBO ve Sinüs-Kosinüs algoritmaları kullanılarak çelik yapıların kaynaklı birleşimlerinin optimum tasarımının yapılması, kullanılan metasezgisel algoritmaların bu çalışmadaki sayısal modeller ekseninde birbirlerine göre avantaj ve dezavantajlarının belirlenmesi ve bu çalışmada ortaya konan Adil Ceza Yöntemi'nin kısıtlı optimizasyon problemlerinin çözümü için kullanılabilirliğinin araştırılması hedeflenmiştir. İlgili hedefler doğrultusunda köşe kaynaklı bir bağlantıdan, Çelik Yapıların Tasarım, Hesap ve Yapımına Dair Esaslar - 2018'de belirtilen gereklilikleri sağlayan üç adet sayısal model oluşturulmuştur. Yönetmelikte tanımlanan Yük ve Dayanım Katsayıları ile Tasarım (YDKT) ve Güvenlik Katsayıları ile Tasarım (GKT) yöntemleri için üç farklı popülasyon büyüklüğü kullanılarak 54 farklı analiz gerçekleştirilmiştir. Yapılan incelemeler neticesinde, ele alınan algoritmaların çelik yapıların kaynaklı birleşimlerin optimizasyonu için uygun olduğu anlaşılmış olup algoritmaların birbirlerine göre avantaj ve dezavantajları ortaya konulmuştur. Bununla beraber, bu çalışmada sunulan Adil Ceza Yöntemi'nin kısıtlı optimizasyon problemlerinin çözümünde kullanılabilir olduğu görülmüştür.

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

#### **Abstract**

In this study, it is aimed to make optimum design of welded joints of steel structures using Jaya, TLBO and Sine-Cosine algorithms, to determine the advantages and disadvantages of the metaheuristic algorithms used in this study in terms of numerical models and to investigate the usability of the "Fair Penalty Method" presented in this study for solving constrained optimization problems. For these goals, three numerical models are created from the fillet welded connection in accordance with the ÇYTHYDE-2018 regulation. 54 different analyzes are carried out using three different population sizes for the Load and Resistance Factor Design (LRFD) and Allowable Strength Design (ASD) methods defined in the regulation. As a result of the investigations, it is understood that the algorithms considered are suitable for the optimization of welded joints of steel structures, and the advantages and disadvantages of the algorithms compared to each other are revealed. In addition, it is observed that the Fair Penalty Method presented in this study is useable in solving constrained optimization problems.

**Keywords:** Steel Structures, Welded Connections, Optimization, Metaheuristic Algorithms

## 1. Giriş

Artan dünya nüfusu ve buna bağlı olarak azalan doğal kaynaklar neticesinde optimizasyon kavramı, günümüzde popüler bir çalışma konusu haline gelmiştir. Optimizasyon, yüzyıllar boyunca çeşitli yöntemlerle yapılagelse de günümüzde bilgisayar teknolojileri ve yapay zekâ algoritmalarındaki gelişimle ortaya çıkan yeni yöntemler daha hızlı ve daha verimli çözümler sunmaktadır. Pek çok disiplin gibi yapı mühendisliği de optimizasyon yöntemlerindeki gelişmelerden ve bu yöntemlerin 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.

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 [1]. Metasezgisel algoritmalar bu noktadaki ihtiyaca pratik bir çözüm üretmektedirler. Metasezgisel (metaheuristic) kelimesi, Yunanca yeni strateji bulma sanatı anlamına gelen "heuriskein" sözcüğünün üst seviye anlamındaki "meta" ön ekiyle birleşmesinden oluşmuş ve özel optimizasyon problemlerinin çözümü için ortaya konulmuş, belirli stratejileri temel alan üst düzey genel metodolojileri ifade etmektedir [2]. Bu algoritmalar doğadan, fizikten, bir canlının av ararken ya da eş seçerken sergilediği davranışlardan, matematikten ve birçok diğer alandan belirli stratejileri temel almaktadırlar. İlham kaynakları neticesinde ortaya çıkan 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 [3]. En çok bilinen sürü tabanlı algoritmalar Parçacık Sürü Optimizasyonu [4], Karınca Kolonisi [5], Yapay

Arı Kolonisi [6], Ateş Böceği [7] ve Cuckoo Arama Algoritması [8]; evrimsel temelli algoritmalarından en yaygın kullanılanlar Genetik Algoritma [9], Diferansiyel Evrim [10], Evrim Stratejisi [11] ve Genetik Programlama [12]; fizik tabanlı algoritmaların en bilinenleri ise Tavlama Benzetimi [13], Büyük Patlama Büyük Çöküş [14], Kara Delik [15] ve Işın Optimizasyonu [16]'dur.

Yapısal optimizasyon kavramı ilk defa Schmit [17] tarafından yapılan bir 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 indireyecek 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 [18].

İlgili literatür incelendiğinde, çelik yapıların optimizasyonu alanında pek çok çalışma yapıldığı görülmektedir. Saka [19] tarafından yapılan çalışmada matematik tabanlı bir algoritma ile AISC (American Institute of Steel Construction) yönetmeliğindeki gerilme ve yer değiştirme kısıtları kullanılarak çelik çerçeve sistemlerin boyutlandırılması gerçekleştirilmiştir. İlgili çalışma amaç fonksiyonu ve kısıtların oluşturularak algoritmalarla tanıtılması gibi konularda pek çok araştırmacıya yol göstermiştir. Daloğlu ve Armutçu [20] tarafından yapılan çalışmada TS648 esas alınarak düzlem çelik çerçeve sistemler genetik algoritma ile boyutlandırılmıştır. Daloğlu ve Aydın [21] genetik algoritma ile düzlem kafes sistemleri optimize etmişlerdir. Kameshki ve Saka [22] tarafından, yarı rijit bağlanmış çerçeveler için P-Δ etkisini de dikkate alarak lineer olmayan bir

yaklaşım ile boyut optimizasyonu yapılmıştır. Hayalioğlu ve Değertekin [23] tarafından yapılan çalışmada yarı rijit bağlı doğrusal olmayan çelik çerçeveler genetik algoritma ile optimize edilmiştir. Kaveh ve Talatahari [24] tarafından, birçok algoritma ile düzlem çelik çerçevelerin optimizasyonu gerçekleştirilmiştir. Saka [25] tarafından yapılan çalışmada Harmoni Arama Algoritması ile BS-5950 yönetmeliğini kullanarak düzlem çelik çerçeve optimizasyonu gerçekleştirilmiştir. Benzer şekilde Toğan [26] tarafından TLBO (Teaching-Learning Based Optimization) algoritması kullanılarak AISC yönetmeliğinin LRFD (Load and Resistance Factor Design) tasarım prensibi için AISC'de listelenmiş W profillerden optimum seçimler yaptırılmıştır. TLBO algoritması, Karınca Kolonisi Optimizasyonu, Harmoni Arama Algoritması, Geliştirilmiş Karınca Kolonisi Optimizasyonu yöntemleri ile karşılaştırılarak güçlü bir algoritma olduğu ortaya konmuştur.

Kaynaklı birleşimlerin optimizasyonuna dair literatürde en çok bilinen problem Ragsdell ve Phillips [27] tarafından ortaya konulmuştur. Problemden kurulum, kaynak işçiliği ve malzeme maliyetlerinin toplamından oluşan amaç fonksiyonunun minimize edilmesi hedeflenmiştir. Tanıtılan kaynaklı giriş problemi Geometrik Programlama yöntemi ile optimize edilmiştir. Bu problem farklı araştırmacılar tarafından da yaygın olarak kullanılmıştır [28, 29, 30, 31, 32]. Bunun neticesinde yeni geliştirilen algoritmaların eski algoritmalar ile mukayesesi noktasında bir test fonksiyonu olarak kabul görmüştür.

Alberdi vd. [33] tarafından çelik çerçevelerin optimize edildiği çalışmada, birleşimler için dört farklı topoloji tanımlanıp çerçeve optimizasyonu ile birlikte birleşim topoloji optimizasyonu da gerçekleştirilmiştir. Jin vd. [34] tarafından yapılan çalışmada, çerçeve ile birlikte kaynaklı birleşimlerin de optimizasyonunu sağlamak amacıyla, maliyet hesabına dayanan ve literatürdeki diğer çalışmalar ile kıyaslandığında nispeten kolay bir yöntem önerilmiştir. Çalışmada Genetik Algoritma kullanılmış ve Japon deprem yönetmeliği dikkate alınmıştır.

Çelik yapıların optimizasyonuna dair literatür incelendiğinde çalışmalarda 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 [22, 23, 25, 26]. Bu tür çalışmalarda birleşim hesapları bazı kabullerle sınırlı kalarak bulunlu ya da kaynaklı birleşim detayları için optimizasyon yapılması yerine analizlerin genellikle rijit ya da yarı rijit kabul ekseninde ilerlediği görülmüştür.

Bu çalışmada; Jaya, TLBO ve Sinüs-Kosinüs (SCA) algoritmaları kullanılarak çelik yapıların kaynaklı birleşimlerinin optimum tasarımının gerçekleştirilmesi, metasezgisel algoritmalar ile elde edilen sonuçların karşılaştırılarak algoritmaların birbirine göre avantajlı ve dezavantajlı yönlerinin belirlenmesi ve bu çalışmada sunulan Adil Ceza Yöntemi'nin kısıtlı optimizasyon problemlerinin çözümünde kullanılabilirliğinin araştırılması amaçlanmıştır. Bu amaçlar doğrultusunda ÇYTHYDE-2018 yönetmeliğinde belirtilen gereklilikleri sağlayan üç adet sayısal model oluşturularak yönetmelikte tanımlanan iki farklı tasarım yöntemi ve üç farklı popülasyon büyüklüğü için 54 ayrı analiz yapılmıştır.

## 2. Materyal ve Metot

Bu çalışmada, detayları bu kısımda verilmiş olan köşe kaynaklı birleşimlerin kaynak alanlarının optimize edilmesi amacıyla Jaya, TLBO ve SCA algoritmaları kullanılmıştır. Analizler Çelik Yapıların Tasarım, Hesap ve Yapımına Dair Esaslar - 2018'de (ÇYTHYDE-2018) tanımlanan yük ve dayanım katsayıları ile tasarım (YDKT) ve güvenlik katsayıları ile tasarım (GKT) yöntemleri esas alınarak gerçekleştirilmiştir.

### 2.1. ÇYTHYDE-2018'e Göre Köşe Kaynaklı Birleşimlerin Tasarımı

ÇYTHYDE-2018'e göre birleşen çelik elemanlar arasındaki açı  $60^\circ$  ile  $120^\circ$  aralığında olan kaynaklar köşe kaynak olarak değerlendirilerek tasarlanacaktır. Köşe kaynaklarda belirlenecek minimum etkin kalınlık, hesaplanmış olan kuvvetin emniyetli şekilde aktarılmasına

**Tablo 1.** Köşe kaynakların minimum kalınlıkları [35].

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

<sup>a</sup>:Tek geçişli kaynaklar kullanılmalıdır.

yetecek minimum kalınlıktan ve Tablo 1’de verilen minimum kalınlıklardan daha az seçilemez [35].

Maksimum kaynak kalınlığı için kaynak ile birleştirilen elemanın kenar kalınlığı  $t$  olmak üzere şu şartlara uyulacaktır: Kaynaklanan ince elemanın kenar kalınlığı olan  $t$ , 6 mm’den küçük ise kaynak kalınlığı  $a$ , 0,7 $t$ ’den daha büyük seçilemez. Kaynaklanan ince elemanın kenar kalınlığı olan  $t$ , 6 mm’den büyük ise kaynak kalınlığı  $a$ , 0,7( $t-2$ ) mm’den daha büyük seçilemez. Köşe kaynakları için minimum etkin uzunluk, kaynak kalınlığı olan  $a$ ’nın 6 katından veya 40 mm’den daha küçük seçilmemelidir. Çelik elemanların kaynaklı uç birleşimlerindeki etkin kaynak uzunluğu aşağıdaki şartlar göz önüne alınarak denklem (1), (2) ve (3) ile hesaplanmaktadır [35].

$$L \leq 150a \text{ için } L_e = L \quad (1)$$

$$150a < L \leq \text{ için } L_e = \beta L \quad (2)$$

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

$$400a < L \text{ için } L_e = 250a \quad (3)$$

Bu denklemlerde,  $L$  kaynağın uzunluğu,  $L_e$  etkin kaynak uzunluğu,  $a$  kaynak etkin kalınlığı,  $\beta$  ise azaltma katsayısıdır. Kaynaklı birleşim noktalarının tasarım dayanımları ( $\phi R_n$ ) veya güvenli dayanımları ( $R_n/\Omega$ ) esas metalin çekme ve kayma etkisindeki kırılma sınır durumları ve kaynak metalinin kırılma sınır durumları hesaplanarak kritik olan değer esas alınarak yapılacaktır. Esas metalin karakteristik dayanımı  $R_{nBM}$  ve kaynak metalinin karakteristik

dayanımı  $R_{nw}$  sırasıyla denklem (4) ve (5) ile hesaplanacaktır [35].

$$R_{nBM} = F_{nBM} A_{BM} \quad (4)$$

$$R_{nw} = F_{nw} A_{we} \quad (5)$$

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 alanıdır.  $\phi$ ,  $\Omega$ ,  $F_{nBM}$  ve  $F_{nw}$  için alınacak değerler ve kullanılacak yöntemler Tablo 2’de verilmiştir. Köşe kaynakların karakteristik gerilmesi olan  $F_{nw}$  kaynağın boyuna ekseninin kuvvetin doğrultusuyla arasındaki açı dikkate alınmadan Tablo 2’den kullanılabilir [35].

## 2.2 Çalışmada Kullanılan Metasezgisel Algoritmalar

Bu çalışmada kullanılmak üzere güncel ve popüler üç farklı metasezgisel algoritma seçilmiştir. Algoritmalar seçilirken birbirinden farklı çalışma prensiplerine sahip olmalarına dikkat edilmiştir. Bu bağlamda; TLBO algoritması çift fazlı ve parametresiz, Jaya algoritması tek fazlı ve parametresiz, SCA ise tek fazlı ve parametrelidir. İlgili algoritmaların hesaplamalarına ilişkin detaylar, akış diyagramları ve örnek problem çözümleri ayrıntılı olarak Sezer [36] tarafından verilmiştir.

### Jaya Algoritması

Rao [37] tarafından geliştirilmiştir. Jaya kelime olarak Sanskritçe’de zafer anlamına gelmektedir. Diğer pek çok metasezgisel algoritmadan farklı olarak herhangi bir doğa olayı, sosyal hayat,

**Tablo 2.** Köşe kaynakların hesabında kullanılacak değerler ve ÇYTHYDE-2018 yönetmeliği içerisinde esas alınacak bölümler [35].

KÖŞE KAYNAKLAR					
Yük Tipi ve Kaynak Eksenine Göre Doğrultusu	İlgili Metal	$\phi$ ve $\Omega$	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			Bölüm 13.2.6
	Kaynak Metali	$\phi = 0,75$ $\Omega = 2,00$	$0,60F_E$	Bölüm 13.2.2.1	
Kaynak eksenine paralel çekme ve basınç	Kaynak eksenine paralel çekme ve basınç etkilerinin kaynak tasarımında göz önüne alınmasına gerek yoktur.				

hayvan davranışları gibi bir esin kaynağı bulunmamaktadır. Algoritma, başarıya yaklaştıkça (en iyi çözüme ulaşmaya) ve başarısızlıktan kaçmaya (en kötü çözümden uzaklaşmaya) dayanan bir matematiksel felsefe ile çalışmakta olup algoritmaya özgü bir parametre içermemektedir. Tek fazlı yapıdadır. Yani her iterasyon için popülasyondaki elemanlar bir defa güncellenmektedir. Popülasyon güncellemesi denklem (6) ile gerçekleştirilmektedir.

$$X'_{j,k,i} = X_{j,k,i} + r_{1,j,i}(X_{j,best,i} - |X_{j,k,i}|) - r_{2,j,i}(X_{j,worst,i} - |X_{j,k,i}|) \quad (6)$$

Yukarıdaki denklemde;  $X_{j,k,i}$   $i$ . iterasyonda  $k$ . çözümün  $j$ . değişkeni (güncellenecek değişken),  $X'_{j,k,i}$   $X_{j,k,i}$ 'in denklem (6) 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,  $X_{j,worst,i}$  ise en kötü çözümü veren çözüm kümesinin  $j$ . değişkenidir.

#### Öğretme - Öğrenme Tabanlı Optimizasyon (TLBO)

Öğretme - Öğrenme Tabanlı Optimizasyon (TLBO: Teaching - Learning Based Optimization), bir derslik içerisindeki öğretme ve öğrenme sürecini taklit eden popülasyon tabanlı bir algoritmadır. Rao vd. [38] tarafından geliştirilmiştir. Derslikteki öğrenci grubu popülasyon olarak kabul edilmektedir. Öğrencilere okutulan farklı dersler optimizasyon probleminin farklı tasarım değişkenlerini ifade etmektedir. 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. Öğretmen aşamasındaki popülasyonu geliştirmek için denklem (7), öğrenci aşamasındaki popülasyonu geliştirmek için ise minimizasyon problemlerinde denklem (8) ve (9) kullanılır. 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.

$$X'_{j,k,i} = X_{j,k,i} + Difference\_Mean_{j,k,i} \quad (7)$$

Bu denklemde;  $i$  iterasyon,  $X_{j,k,i}$   $i$ . iterasyonda  $k$ . çözümün  $j$ . değişkeni ( $k$ . öğrencinin aldığı  $j$ . derse karşılık gelmektedir),  $X'_{j,k,i}$  popülasyondaki her bir  $X_{j,k,i}$  değerinin denklem (7) ile güncellenmiş hali,  $Difference\_Mean_{j,k,i}$  ise her ders için öğretmenin sonucundan mevcut ortalama sonucun  $T_f$  mislinin çıkarılıp, elde edilen değer [0,1] aralığında rastsal bir sayı ile çarpılmasıyla bulunur. Burada  $T_f$  değeri 1 veya 2 olarak kendiliğinden seçilir.

$$X''_{j,p,i} = X'_{j,p,i} + r_i(X'_{j,p,i} - X'_{j,q,i}) \quad (8)$$

$$; X'_{total-p,i} < X'_{total-q,i}$$

$$X''_{j,q,i} = X'_{j,q,i} + r_i(X'_{j,q,i} - X'_{j,p,i}) \quad (9)$$

$$; X'_{total-q,i} < X'_{total-p,i}$$

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 amaç fonksiyonundan geçmiş halidir.

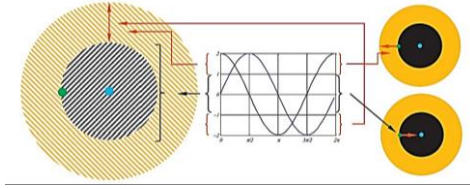
#### Sinüs-Kosinüs Algoritması (SCA)

Sinüs - Kosinüs Algoritması Mirjalili [39] 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 1'de gösterilmiştir. Bu çalışmada kullanılan diğer algoritmaların aksine SCA, çalışmak için kendine özgü bazı parametrelere ihtiyaç duymaktadır. Bu durum doğru parametrelerin seçilebilmesi için bir dizi deneme ihtiyacını ortaya çıkarmaktadır. SCA için popülasyon güncellemesi denklem (10) ile yapılmaktadır.

$$X_i^{t+1} = \begin{cases} x_i^t + r_1 * \sin(r_2) * |r_3 * P_i^t - X_i^t| & ; r_4 < 0,5 \\ x_i^t + r_1 * \cos(r_2) * |r_3 * P_i^t - X_i^t| & ; r_4 \geq 0,5 \end{cases} \quad (10)$$

Bu denklemde;  $P_i^t$  elde edilen en iyi çözümün ilgili tasarım değişkeni,  $r_1$  bir sonraki pozisyon bölgesi veya hareket yönünü belirlemektedir

( $r_1=a-ta/T$ ). Keşif ve sömürü arasındaki dengeyi sağlamak için kullanılmaktadır yani lokal ekstremum noktalarında takılmayı önleme görevine sahiptir.  $a$  algoritma parametresi,  $T$  toplam iterasyon sayısı,  $t$  içinde bulunulan iterasyon numarasıdır.  $r_2$  içe ya da dışa doğru hareket miktarını belemektedir.  $[0, 2\pi]$  aralığında radyan cinsinden rastgele açı ölçüsüdür.)  $r_3$  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.



Şekil 1. SCA için arama modeli [39].

### 2.3 Problemlerin Tanıtımı ve Optimizasyon Modellerinin Oluşturulması

Bu çalışmada 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 ve uzunlukları eşit kabul edilmiştir. Bunun sebebi kaynak gruplarındaki asimetrik ölçülerin, kaynak düzleminde eksantrisite oluşturması 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 modeline ait genel durum Şekil 2’de gösterildiği gibidir. Çelik malzeme sınıfı olarak tüm profiller için S275 tercih edilmiştir. Kaynak metali için  $F_E=480$  N/mm<sup>2</sup> seçilmiştir. Sayısal modellerde kullanılacak bütün yükler sabit yük olarak kabul

edilmiştir. Genel hali tanımlanan 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) yöntemleri için; moment etkin, normal kuvvet etkin ve kesme kuvveti etkin yükleme durumları altında çözülecektir. Sayısal modeller için isimlendirme Tablo 3’de, farklı yükleme durumlarını temsil eden yük dağılımları ise Tablo 4’de verilmiştir. Yük ve momentler birleşimin kapasitesini mümkün olduğunca yüksek oranda kullanacak şekilde seçilmiştir.

Tablo 3. Tasarım esasları ve yükleme durumlarına göre sayısal modeller.

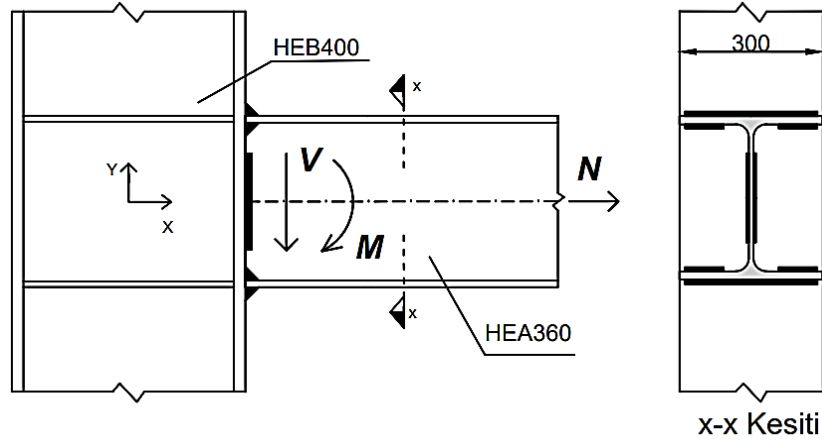
Tasarım Prensipleri	Yükleme Durumları		
	Moment Etkin	Normal Kuvvet Etkin	Kesme Kuvveti Etkin
GKT	SM-1A	SM-2A	SM-3A
YDKT	SM-1B	SM-2B	SM-3B

Tablo 4. 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

### Tasarım Değişkenleri

Tüm sayısal modellerde başlık dış, başlık iç ve gövde olmak üzere üç adet kaynak grubu bulunmaktadır. Optimizasyon algoritmalarının verimsiz bölgelerde arama yapmasının önüne geçilmesi maksadıyla her bir kaynak grubu için eşit kaynak kalınlığı ve eşit kaynak uzunluğu seçilmiştir. Yani her şartta kaynak alanları  $x$  ve  $y$  eksenine göre simetriktir. Bu sayede kaynak düzlemindeki eksantrisitenin önüne geçilmiştir.



Şekil 2. Birleşim modelinin genel görünümü.

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 Tablo 5'de açıklanarak Şekil 3'de gösterilmiştir.

Tablo 5. Tasarım değişkenleri.

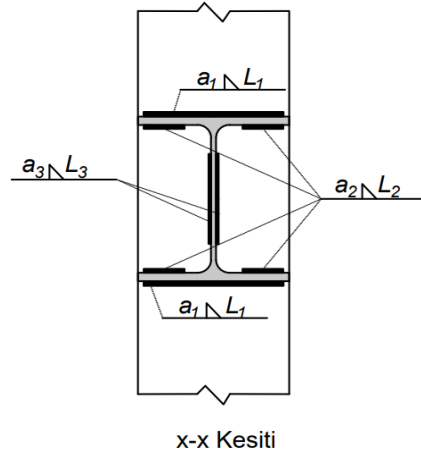
Tasarım Değişkeni	Açıklama
$a_1$	Başlık dış kaynak grubu için kaynak kalınlığı.
$a_2$	Başlık iç kaynak grubu için kaynak kalınlığı.
$a_3$	Gövde kaynak grubu için kaynak kalınlığı.
$L_1$	Başlık dış kaynak grubu için kaynak uzunluğu.
$L_2$	Başlık iç kaynak grubu için kaynak uzunluğu.
$L_3$	Gövde kaynak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

### Amaç Fonksiyonu

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 alanının minimizasyonu seçilmiştir.

Kısıtlı optimizasyon yöntemi için amaç fonksiyonu denklem (11)'de gösterilmiştir.

$$\min F_{(n)} = (2a_1L_1 + 4a_2L_2 + 2a_3L_3) + p_{(n)} \quad (11)$$



Şekil 3. Tasarım değişkenleri.

### Kısıtlar

Optimize edilecek sayısal modeller GKT ve YDKT tasarım yöntemleri için ayrı iki süreç olarak kodlanmıştır. Güvenlik katsayıları ile tasarım ile yük ve dayanım katsayılarına göre tasarım yöntemlerinde kullanılan idealize edilmiş kısıt fonksiyonları Tablo 6'da gösterildiği gibidir. Kısıtların kontrol edilmesine dair detaylar ve hesap prosedürleri Sezer [36] tarafından yapılan çalışmada verilmiştir.

**Tablo 6.** GKT ve YDKT tasarım yöntemleri için kısıt fonksiyonları.

Kısıt Açıklaması	GKT İçin Kısıt Fonksiyonları	YDKT İçin Kısıt Fonksiyonları
Kaynak kalınlığı kontrolü.	$if\ a_{min} < a < a_{max} \Rightarrow g_1 = 0\ else\ g_1 = 1 ;\ g_1 < 1$	
Etkin kaynak uzunluğu kontrolü.	$if\ Le_{min} < Le < Le_{max} \Rightarrow g_2 = 0\ else\ g_2 = 1 ;\ g_2 < 1$	
Başlık dış kaynak grubuna gelen gerilmelerin kontrolü.	$g_3 = \frac{Md/2I_w + N/A_{we}}{0,6F_e/\Omega} \leq 1$	$g_7 = \frac{Md/2I_w + N/A_{we}}{0,6F_e\phi} \leq 1$
Gövde üst ucuna gelen gerilmelerin kontrolü.	$g_4 = \frac{MLE_3/2I_w + N/A_{we}}{0,6F_e/\Omega} \leq 1$	$g_8 = \frac{MLE_3/2I_w + N/A_{we}}{0,6F_e\phi} \leq 1$
Gövde kaynak grubuna gelen kayma gerilmelerinin kontrolü.	$g_5 = \frac{V/A_{we3}}{0,6F_e/\Omega} \leq 1$	$g_9 = \frac{V/A_{we3}}{0,6F_e\phi} \leq 1$
Gövde kaynak grubuna gelen bileşke gerilmelerin kontrolü.	$g_6 = \sqrt{\frac{(\frac{MLE_3}{2I_w} + \frac{N}{A_{we}})^2 + (\frac{V}{A_{we3}})^2}{0,6F_e/\Omega}} \leq 1$	$g_{10} = \sqrt{\frac{(\frac{MLE_3}{2I_w} + \frac{N}{A_{we}})^2 + (\frac{V}{A_{we3}})^2}{0,6F_e\phi}} \leq 1$

Bu tabloda,  $a$ : kaynak kalınlığı,  $F_e$ : kaynak metalinin karakteristik çekme dayanımı,  $\Omega$ : güvenlik katsayısı,  $\phi$ : dayanım katsayısı,  $I_w$ : toplam etkin kaynak dikisi alanının kirişin kuvvetli eksenine göre toplam atalet momenti,  $d$ : kirişin yüksekliği,  $M$ : moment,  $N$ : normal kuvvet,  $V$ : kesme kuvveti,  $L_e$ : etkin kaynak uzunluğu,  $A_{we}$ : etkin kaynak alanıdır.

#### Adil Ceza Yöntemi ve Ceza Fonksiyonu

Kısıtlı optimizasyon problemlerinin çözülebileceği için problemin yapısına göre seçilebilecek pek çok farklı cezalandırma metodu bulunmaktadır. Bu çalışmada "Adil Ceza Yöntemi" olarak isimlendirilen özgün bir yöntem 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. Adil Ceza Yöntemi'nde her bir kısıt her eleman için kontrol edilerek uygun olup olmadıkları kısıt kontrol vektörlerinde saklanır. Uygunluk kontrolü yapılırken temel bilgisayar mantığındaki ikili sistemden faydalanılır. İlgili kontrol mekanizması denklem (12)'de gösterilmiştir. Bu denklemde;  $g_{k(n)}$  kısıt kontrol vektörü (kısıtlar için kısıta uymama sayısı yani

hata sayacı),  $k$  kısıt numarası,  $n$  popülasyon eleman numarasıdır.

$$g_{k(n)} = 0 ; \text{ ilgili kısıta uyuluyorsa} \quad (12)$$

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

Kısıt kontrolleri yapıldıktan sonra kısıt kontrol vektörleri toplanarak her bir popülasyon elemanının kısıtları toplamda ne kadar ihlal ettiği elde edilerek  $g_{top(n)}$  vektöründe saklanır. Bu işlem denklem (13)'de gösterilmiştir.

$$g_{top(n)} = \sum_{x=1}^{k_{max}} (g_x(n)) \quad (13)$$

$g_{top(n)}$  vektörü ceza fonksiyonunda bir parametre olarak kullanılır. İlgili ceza fonksiyonu denklem (14)'te verilmiştir. Amaç fonksiyonundaki tasarım değişkenlerinin çarpım durumunda olduğu optimizasyon problemlerinde, algoritma bütün tasarım değişkenlerini sıfıra götürdüğünde ceza verilemeyecektir. Bu sorunun önüne geçilebilmesi maksadıyla ceza fonksiyonuna toplam durumunda 1000 sabiti bulunmaktadır.

$$p(n) = kc_k(F(n) + 1000) * g_{top(n)} \quad (14)$$



Bu denklemlerde;  $p_{(n)}$  her bir eleman için verilecek ceza miktarı,  $k$  kısıt no,  $k_{C_k}$  her bir kısıt için kısıt ceza çarpanı (Gerektiğinde kısıtlardan herhangi birisi için verilecek cezanın arttırılıp azaltılması amacıyla kullanılmaktadır.),  $F_{(n)}$  ceza öncesi hesaplanmış amaç fonksiyonu değeri,  $J_{top(n)}$  her bir eleman için kısıtların toplam ihlal edilme sayısıdır. Adil Ceza Yöntemine ilişkin detaylar Sezer [36] tarafından verilmiştir.

### Optimizasyon Algoritmaları İçin Belirlenen Parametreler

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 büyüklüğü ( $n$ ), 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 sınırlardır ( $p_{üst}$ ). Seçilen ortak parametreler Tablo 7'de verilmiştir. Farklı kıyaslama durumları elde edebilmek amacıyla üç farklı popülasyon büyüklüğü ( $n$ ) ile analiz yapılmıştır. Farklı yükleme durumları altında elde edilen 3 adet sayısal model, 2 farklı tasarım prensibi, 3 farklı popülasyon büyüklüğü ve 3 farklı optimizasyon algoritması ile toplamda birbirinden farklı 54 adet analiz gerçekleştirilmiştir. Bütün algoritmaların global optimuma ulaşabilmesi amacıyla iterasyon sayısı

**Tablo 7.** Ortak çalıştırma parametreleri.

Sembol	Parametre Değeri
$n$	250, 500 ve 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

( $T$ ) mümkün olduğunca büyük seçilmiştir. Arama uzayının sınırlarını belirleyen parametreler olan  $p_{alt}$  ve  $p_{üst}$  algoritmaların gereksiz aramaların yapılacağı bölgelerden mümkün olduğunca uzak tutacak kadar dar, çözümü içerebilecek bölgelere mümkün olduğunca kapsayacak kadar geniş bir aralıkta seçilmeye çalışılmıştır.

SCA için yukarıda açıklanan ortak çalışma parametrelerine ek olarak algoritmaya özel bazı parametrelere de ihtiyaç duyulmaktadır. Bu çalışmada,  $b$  parametresi rastlantısallığa eşit şans vermek amacıyla 2 olarak seçilmiştir.  $a$  parametresinin seçilebilmesi için bir dizi analiz yapılması ihtiyacı ortaya çıkmıştır. Bu sebeple  $0,1 \leq a \leq 3,0$  aralığında bazı deneyler gerçekleştirilmiş ve nihayetinde en iyi sonuçların elde edildiği parametre olan  $a=0,5$  seçilmiştir.

### 2.4 Algoritmaların 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, en kötü ve ortalama kaynak alanı değerleri, standart sapmalar ve iterasyonların işlem süreleridir. 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. Tüm analizler, Matlab [40] yazılım dilinde oluşturulmuş kodlar vasıtasıyla gerçekleştirilmiştir. Bunlara ilaveten, karşılaştırmalar için 10 çalıştırmanın her bir iterasyonundaki amaç fonksiyonu değerlerinin ortalamaları elde edilerek iterasyon-toplam kaynak alanı grafikleri oluşturulmuştur.

### 3. Bulgular

Sayısal modeller optimizasyon algoritmaları ile çözümlenmiş ve elde edilen sonuçlar bu bölümde özetlenmiştir. Tüm Sonuçlar ÇYHYDE-2018'e bağlı kalınarak tanımlanan tüm kısıtları sağlamaktadır. Tablo 8'de analizler neticesinde elde edilen en iyi sonuç değerleri verilmiştir. Bu tabloda  $A_w$  toplam kaynak alanı, GS-1 başlık dış kaynak grubunun üst ucuna gelen normal gerilmelerin kapasite kullanımı, GS-2 gövde kaynak grubunun üst ucunda oluşan normal gerilmelerin kapasite kullanımı, GS-3 gövde kaynak grubuna gelen kayma gerilmelerinin kapasite kullanımı, GS-4 ise gövde

**Tablo 8.** Elde edilen en iyi sonuç değerleri.

SM	Alg.	$A_w$ ( $mm^2$ )	GS-1	GS-2	GS-3	GS-4	SM	Alg.	$A_w$ ( $mm^2$ )	GS-1	GS-2	GS-3	GS-4
SM-1A	Jaya	3955,36	1,00	0,13	0,22	0,25	SM-2B	Jaya	1977,21	1,00	0,74	0,17	0,76
	SCA	3961,83	1,00	0,13	0,22	0,25		SCA	1981,86	1,00	0,74	0,17	0,76
	TLBO	3955,25	1,00	0,13	0,22	0,25		TLBO	1975,99	1,00	0,74	0,17	0,76
SM-1B	Jaya	3717,05	1,00	0,13	0,17	0,22	SM-3A	Jaya	1927,64	0,50	0,16	0,99	1,00
	SCA	3720,53	1,00	0,13	0,17	0,22		SCA	1923,27	0,51	0,16	0,99	1,00
	TLBO	3716,73	1,00	0,13	0,17	0,22		TLBO	1919,34	0,51	0,16	0,99	1,00
SM-2A	Jaya	2291,05	1,00	0,78	0,22	0,81	SM-3B	Jaya	1776,03	0,47	0,12	0,99	1,00
	SCA	2299,11	1,00	0,78	0,21	0,81		SCA	1780,83	0,47	0,12	0,99	1,00
	TLBO	2290,19	1,00	0,78	0,22	0,81		TLBO	1775,88	0,47	0,12	0,99	1,00

kaynak grubuna gelen bileşke gerilmelerin kapasite kullanımıdır.

Algoritmalar tarafından elde edilen sonuçlar incelendiğinde; SM-1'de beklendiği şekilde başlık dış kaynak grubunun dış ucuna gelen normal gerilmelerin kapasitesi (GS-1) neredeyse %100 oranında kullanılmıştır. SM-2'de de GS-1 kapasitesi neredeyse %100 kullanılırken gövdeye gelen bileşke gerilmelerin kapasitesinin (GS-4) de yüksek oranda kullanıldığı görülmektedir. Bu kapasite kullanımı kayma gerilmelerinden ziyade gövde kaynak grubunun

dış ucuna gelen normal gerilmelerin kapasitesinin (GS-2) yüksek oranda kullanılmasından kaynaklanmaktadır. SM-3 için gövde bileşke kapasitesi (GS-4) neredeyse %100 oranında kullanılmaktayken ilgili kapasite kullanımı beklendiği şekilde gövde kayma gerilmesi kapasitesinin %99 oranında kullanımından oluşmaktadır. Sonuçlar incelendiğinde, algoritmaların kaynak alanlarını minimize ederek, maksimum kapasite kullanım hedefine başarılı bir şekilde ulaştıkları anlaşılmaktadır.

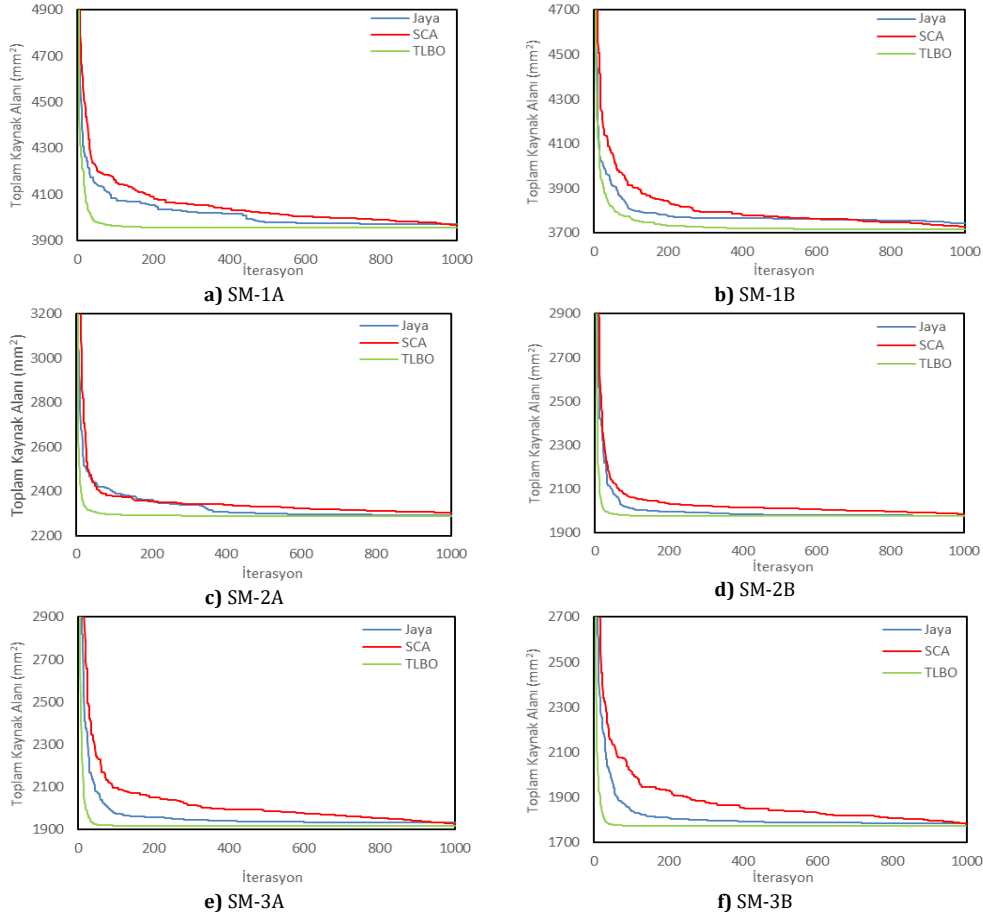
**Tablo 9.** Karşılaştırma parametreleri.

n	Durum	SM-1A			SM-1B			SM-2A		
		Jaya	SCA	TLBO	Jaya	SCA	TLBO	Jaya	SCA	TLBO
250	Eİ ( $mm^2$ )	3955,28	3960,86	3955,25	3716,75	3724,01	3716,73	2290,66	2298,23	2290,19
	$\bar{x}$ ( $mm^2$ )	4021,31	3971,43	3955,26	3776,97	3733,04	3716,73	2294,84	2305,16	2290,19
	EK( $mm^2$ )	4231,91	3979,33	3955,29	4105,20	3744,39	3716,74	2322,69	2311,21	2290,20
	S	101,92	6,29	0,01	131,62	7,36	0,00	9,86	3,59	0,00
	$t_w$ (sn)	63,68	61,40	139,24	63,85	61,04	139,02	58,52	57,56	137,41
500	Eİ ( $mm^2$ )	3955,28	3965,59	3955,25	3716,76	3725,12	3716,73	2291,42	2298,95	2290,19
	$\bar{x}$ ( $mm^2$ )	4010,78	3971,33	3955,34	3749,95	3729,64	3716,73	2296,70	2304,51	2290,19
	EK( $mm^2$ )	4184,49	3978,12	3955,82	3902,78	3740,36	3716,73	2337,66	2309,44	2309,44
	S	78,99	4,97	0,20	56,53	4,74	0,00	14,40	2,99	0,00
	$t_w$ (sn)	124,61	116,48	272,18	98,82	94,96	249,01	91,94	97,67	250,86
1000	Eİ ( $mm^2$ )	3955,36	3961,83	3955,25	3717,05	3720,53	3716,73	2291,05	2299,11	2290,19
	$\bar{x}$ ( $mm^2$ )	3971,51	3967,09	3955,25	3743,90	3728,15	3716,75	2292,79	2302,36	2290,19
	EK( $mm^2$ )	4022,45	3976,53	3955,25	3852,48	3736,46	3716,81	2295,91	2308,87	2290,20
	S	25,50	5,09	0,00	44,24	4,85	0,03	1,30	3,06	0,00
	$t_w$ (sn)	153,59	184,18	486,79	182,14	187,90	481,23	178,97	181,51	479,73
n	Durum	SM-2B			SM-3A			SM-3B		
		Jaya	SCA	TLBO	Jaya	SCA	TLBO	Jaya	SCA	TLBO
250	Eİ ( $mm^2$ )	1976,18	1981,60	1975,99	1919,36	1924,37	1919,34	1775,90	1782,00	1775,88
	$\bar{x}$ ( $mm^2$ )	1976,77	1987,80	1975,99	1935,08	1931,13	1919,34	1785,94	1786,08	1775,88
	EK( $mm^2$ )	1978,04	1996,47	1975,99	1961,93	1938,78	1919,34	1806,49	1789,52	1775,88
	S	0,48	4,89	0,00	15,10	3,93	0,00	9,64	2,59	0,00
	$t_w$ (sn)	57,51	56,89	135,71	55,91	56,32	138,22	54,58	61,73	139,97
500	Eİ ( $mm^2$ )	1976,44	1984,37	1975,99	1919,36	1926,50	1919,34	1776,10	1780,94	1775,88
	$\bar{x}$ ( $mm^2$ )	1977,42	1987,35	1975,99	1928,09	1929,47	1919,34	1786,57	1783,87	1775,88
	EK( $mm^2$ )	1979,62	1991,01	1975,99	1940,01	1934,81	1919,34	1797,34	1787,17	1775,88
	S	1,20	2,38	0,00	7,60	2,71	0,00	6,80	2,11	0,00
	$t_w$ (sn)	98,36	94,82	278,97	107,67	118,71	330,81	91,90	95,09	249,28
1000	Eİ ( $mm^2$ )	1977,21	1981,86	1975,99	1927,64	1923,27	1919,34	1776,03	1780,83	1775,88
	$\bar{x}$ ( $mm^2$ )	1978,37	1985,94	1975,99	1932,35	1927,15	1919,34	1784,98	1783,69	1775,88
	EK( $mm^2$ )	1982,47	1990,98	1975,99	1941,08	1932,00	1919,34	1793,10	1786,69	1775,88
	S	1,60	2,90	0,00	4,51	2,70	0,00	5,85	1,85	0,00
	$t_w$ (sn)	181,89	182,18	487,40	175,20	183,24	484,65	177,70	181,72	478,73

Tablo 9'da üç farklı popülasyon büyüklüğü için elde edilmiş karşılaştırma parametreleri verilmiştir. En iyi sonuçlar (EI) değerlendirildiğinde, tüm varyasyonlarda TLBO algoritması ile daha başarılı sonuçlar elde edildiği görülmektedir. Varyasyonların büyük çoğunluğunda Jaya algoritması da TLBO'ya oldukça yakın sonuçlar bulurken SCA en düşük performansı göstermiştir. Ortalama ( $\bar{x}$ ) ve en kötü (EK) sonuçlar değerlendirildiğinde, TLBO en başarılı sonuçları bulurken bunu genellikle SCA izlemiştir. Jaya algoritması ise bu parametrelerde çoğu varyasyon için en başarısız değerleri elde etmiştir. En iyi değerleri elde etmede oldukça başarılı olan Jaya algoritmasının bu parametreler için en başarısız sonuçları vermesinin sebebinin, çoğu durum için diğer algoritmalarından daha yüksek standart sapmalı sonuçlar elde etmesinden kaynaklandığı düşünülmektedir.

Standart sapmalar (S) değerlendirildiğinde, TLBO algoritması sıfıra oldukça yakın standart sapmalar elde ederek en kararlı davranışı göstermiştir. SCA ise genellikle Jaya'dan daha küçük standart sapmalar ile daha kararlı değerler elde etmiştir.

İşlem süreleri ( $t_w$ ) değerlendirildiğinde, Jaya algoritması ile SCA'nın performansları birbirine yakın bulunmuştur. TLBO algoritması ise eşit çalışma koşullarında diğer algoritmaların iki ila üç katı kadar fazla sürede çözüm bulabilmiştir. TLBO özdeş geliştirmenin yapıldığı durumlarda (diğer algoritmaların yarısı kadar popülasyon elemanı ile) da süre bakımından en başarısız algoritma iken çalışma koşullarının TLBO aleyhine olduğu durumlar incelendiğinde sonuç değişmektedir. TLBO algoritması ile  $n=250$  eleman ile elde edilen sonuçlar Jaya ve SCA için  $n=1000$  eleman



Şekil 4. İterasyon-toplam kaynak alanı grafikleri.

ile elde edilen sonuçlardan hem daha başarılıdır hem de analizleri daha kısa sürelerde tamamlanmıştır.

Şekil 4'te, her biri 1000 iterasyonluk 10 çalıştırma için ortalama değerler ile elde edilen iterasyon - toplam kaynak alanı grafikleri verilmiştir. Grafikler  $n=1000$  elaman için yapılan analizler esas alınarak oluşturulmuştur. Grafikler incelendiğinde, TLBO algoritmasının her durumda global optimuma ilk ulaşan algoritma olduğu açıkça görülmektedir. Jaya algoritması ve SCA kıyaslandığında; SM-1A ve SM-1B için iterasyonun erken aşamalarında Jaya algoritması daha başarıyla, iterasyon ilerledikçe SCA'nın öne geçtiği gözlenmiştir. SM-2A ve SM-2B için iki algoritmanın performansı iterasyonun ilk aşamalarında birbirine yakinken sürecin genelinde Jaya algoritmasının daha başarılı ilerlediği görülmüştür. SM-3A ve SM-3B için ise Jaya algoritmasının TLBO'ya yakın bir performans göstererek SCA'dan pozitif ayrıştığı gözlenmiştir.

#### 4. Tartışma 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 Yöntemi'nin kısıtlı optimizasyon problemlerinin çözümü için kullanılabilirliğinin araştırılması hedeflenmiştir. Bu hedef doğrultusunda ÇYTHYDE-2018 için tanımlanan gereklilikleri yerine getiren üç adet sayısal model oluşturularak yönetmelikte tanımlanan iki farklı tasarım prensibi ve üç farklı popülasyon büyüklüğü için toplamda 54 adet analiz gerçekleştirilmiştir. Bu çalışma neticesinde ulaşılan sonuçlar şu şekildedir:

a. Yapılan 54 farklı analizin sonuçları değerlendirildiğinde; bulunan optimum kaynak alanlarının son derece tutarlı olduğu, ÇYTHYDE-2018 yönetmeliğinde belirtilen tüm şartları yerine getirdiği ve her bir sayısal model için maksimum kapasite kullanımını sağlayan minimum kaynak alanlarının elde edildiği 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, TLBO algoritması tüm durumlar için Jaya ve Sinüs - Kosinüs algoritmalarından daha başarılı sonuçlara ulaştığı görülmüştür. TLBO algoritması ile diğer algoritmalar karşılaştırılırken TLBO'nun çift fazlı, SCA ve Jaya'nın ise tek fazlı olduğu dikkate alınmalıdır. Çalıştırma parametreleri eşit seçilirse TLBO çift fazlı süreç sayesinde iki kat daha fazla geliştirme yapmaktadır. Daha adil bir karşılaştırma yapılabilmesi için çalıştırma parametreleri eşit değil geliştirme sayısı özdeş olacak şekilde seçilebilir. Aynı geliştirme sayılarının elde edilebilmesi için TLBO'nun diğer algoritmaların yarısı kadar popülasyon elemanı ile yaptığı optimizasyonlar esas alınmalıdır. Eşit geliştirme sayısı ile yapılan karşılaştırmalarda, analiz süreleri hariç tüm parametrelerde TLBO algoritmasının daha başarılı sonuçlara ulaştığı görülmüştür dolayısıyla TLBO için hesaplama maliyetinin daha yüksek olmasından başka bir dezavantaj tespit edilememiştir. Çalıştırma parametreleri TLBO algoritmasının aleyhinde seçildiğinde bile (TLBO için popülasyon eleman sayısı  $n=250$ , SCA ve Jaya için  $n=1000$ ), TLBO'nun diğer algoritmaların yarısı kadar geliştirme sayısı ile daha başarılı sonuçlara ulaştığı, eşit geliştirme sayısı ile yapılan karşılaştırmaların aksine hesaplama maliyetlerinin de daha düşük olduğu görülmüştür. Ayrıca tüm analizler için TLBO algoritmasının sıfıra oldukça yakın standart sapma değerleri elde edebilecek kadar kararlı çalıştığı gözlenmiştir. TLBO algoritmasının çok yüksek kararlılıkta çalışmasında 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 başarıyla SCA ise daha küçük standart sapmalar vererek daha kararlı sonuçlar bulmuştur. 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 değerlendirilirken SCA'nın parametrelili bir algoritma olduğu ve

farklı parametre ayarlamaları ile daha iyi ya da daha kötü sonuçlar alınabilme ihtimalinin bulunduğu unutulmamalıdır. Bununla beraber, algoritmalar arasındaki karşılaştırmaların bu çalışmada sunulan problemler özelinde yapılmış olduğu ve farklı optimizasyon problemleri için yapılacak karşılaştırmalarda algoritma performanslarının değişkenlik gösterebileceği bilinmelidir.

d. Elde edilen optimizasyon sonuçlarının tamamı algoritma kısıtlarını, dolayısıyla ÇYTHYDE-2018 için tanımlanan sınırlayıcıları sağlamaktadır. İlgili kısıtların sağlanabilmesi için bu çalışmada sunulan “Adil Ceza Yöntemi” başarılı bulunmuştur. Yöntemin kısıtlı optimizasyon problemlerinin çözümünde cezalandırma işlevi sağlamak üzere kullanılabilir olduğu anlaşılmıştır.

### Kaynakça

- [1] Pham, D. T., Karaboga, D. 2000. Intelligent Optimisation Techniques. Springer-Verlag, London 302s.
- [2] Talbi, E. G. 2009. Metaheuristic: from Design to Implementation. 2nd Edition. Wiley, New Jersey, 624s.
- [3] Yildiz, A. R., Abderazek, H., Mirjalili, S. 2020. A Comparative Study of Recent NonTraditional Methods for Mechanical Design, Optimization Archives of Computational Methods in Engineering, Cilt. 27, s. 1031-1048. DOI: 10.1007/s11831-019-09343-x
- [4] Kennedy, J., Eberhart, R. C. 1995. Particle Swarm Optimization. Proceedings of IEEE International Conference on Neural Networks. November 27-December 1, Perth, 1942-1948.
- [5] Dorigo, M., Birattari, M., Stutzle, T. 2006. Ant Colony Optimization—Artificial Ants as a Computational Intelligence Technique, IEEE Computational Intelligence Magazine, Cilt. 1, s. 28-39. DOI: 10.1109/CI-M.2006.248054
- [6] Karaboğa, D., Baştürk, B. 2007. A Powerful and Efficient Algorithm for Numerical Function Optimization: Artificial Bee Colony (ABC) Algorithm, Journal of Global Optimization, Cilt. 39, s. 459-471. DOI: 10.1007/s10898-007-9149-x
- [7] Yang, X. S. 2010. Nature-Inspired Metaheuristic Algorithms. 2nd Edition. Luniver Press, Fome, 147s.
- [8] Gandomi, A. H., Yang, X. S., Alavi, A. H. 2013. Cuckoo Search Algorithm: A Metaheuristic Approach to Solve Structural Optimization Problems, Engineering with Computers, Cilt. 29, s. 17-35. DOI: 10.1007/s00366-011-0241-y
- [9] Holland, J. H. 1975. Adaptation in Natural And Artificial Systems. University of Michigan Press, Ann Arbor, 194s.
- [10] 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.
- [11] Beyer, H. G., Schwefel, H. P. 2002. Evolution Strategies a Comprehensive Introduction, Natural Computing, Cilt. 1, s. 3-52. DOI: 10.1023/A:1015059928466
- [12] Koza, J. R. 1994. Genetic Programming II, Automatic Discovery of Reusable Subprograms. MIT Press, Cambridge, 746s.
- [13] Kirkpatrick, S., Gelatt, C. D., Vecchi, M. P., 1983. Optimization by Simulated Annealing, Science, Cilt. 220, s. 671-680. DOI: 10.1126/science.220.4598.671
- [14] Erol, O. K., Eksin, I. 2006. A New Optimization Method: Big Bang- Big Crunch, Advances in Engineering Software, Cilt. 37, s. 106-111. DOI: 10.1016/j.advengsoft.2005.04.005
- [15] Hatamlou, A. 2013. Black hole: A New Heuristic Optimization Approach for Data Clustering, Information Sciences, Cilt. 222, s. 175-184. DOI: 10.1016/j.ins.2012.08.023
- [16] Kaveh, A., Khayatizad, M. 2012. A New Meta-Heuristic Method: Ray Optimization, Computers & Structures, Cilt. 112, s. 283-294. DOI: 10.1016/j.compstruc.2012.09.003
- [17] Schmit, L. A. 1960. Structural Design by Systematic Synthesis. 2nd Conference on Electronic Computation, September 8-9, Pittsburgh, 105-132.
- [18] Saka, M. P., Hasançebi, O., Geem, Z. W. 2016. Metaheuristics in Structural Optimization and Discussions on Harmony Search Algorithm, Swarm and Evolutionary Computation, Cilt. 28, s. 88-97. DOI: 10.1016/j.swevo.2016.01.005
- [19] Saka, M. P. 1991. Optimum Design of Steel Frames with Stability Constraints, Computers and Structures, Cilt. 41, s. 1365-1377. DOI: 10.1016/0045-7949(91)90274-P
- [20] Daloğlu, A., Armutçu, M. 1998. Genetik Algoritma ile Düzlem Çelik Çerçevelerin Optimum Tasarımı, Teknik Dergi, Cilt. 116, s. 1601-1615.
- [21] Daloğlu, A., Aydın, Z. 1999. Kafes Sistemlerin Uygulamaya Yönelik Optimum Tasarımı, Pamukkale Üniversitesi Mühendislik Fakültesi Mühendislik Bilimleri Dergisi, Cilt. 5, s. 951-957.
- [22] Kameshki, E. S., Saka, M. P. 2001. Optimum Design of Nonlinear Steel Frames with Semi-Rigid Connections Using a Genetic Algorithm, Computers and Structures, Cilt. 79, s. 1593-1604. DOI: 10.1016/S0045-7949(01)00035-9
- [23] 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, Cilt. 83, s. 1849-1863. DOI: 10.1016/j.compstruc.2005.02.009
- [24] 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, Cilt. 9, s. 563-575.

- [25] Saka, M. P. 2009. Optimum Design of Steel Sway Frames to BS5950 Using Harmony Search Algorithm, *Journal of Constructional Steel Research*, Cilt. 65, s. 36-43. DOI: 10.1016/j.jcsr.2008.02.005
- [26] Toğan, V. 2012. Design of Planar Steel Frames Using Teaching-Learning Based Optimization, *Engineering Structures*, Cilt. 34, s. 225-232. DOI: 10.1016/j.engstruct.2011.08.035
- [27] 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*, Cilt. 98, s. 1021-1025. DOI: 10.1115/1.3438995
- [28] Coello, C. A. C. 2000. Use of a Self-Adaptive Penalty Approach for Engineering Optimization Problems, *Computers in Industry*, Cilt. 41, s. 113-127. DOI: 10.1016/S0166-3615(99)00046-9
- [29] 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*, Cilt. 194, s. 3902-3933. DOI: 10.1016/j.cma.2004.09.007
- [30] Kaveh, A., Talatahari, S. 2010. A novel heuristic optimization method: charged system search, *Acta Mechanica*, Cilt. 213, s. 267-289. DOI:10.1007/s00707-009-0270-4
- [31] Mirjalili, S., Mirjalili, S. M., Lewis, A. 2014. Grey Wolf Optimizer, *Advances in Engineering Software*, Cilt. 69, s. 46-61. DOI:10.1016/j.advengsoft.2013.12.007
- [32] David, D. C. N., Stephen, S. E. A., Ajoy, J. A. 2016. Cost Minimization of Welded Beam Design Problem Using PSO, SA, PS, GOLDLIKE, CUCKOO, FF, FP, ALO, GSA and MVO, *International Journal of Applied Mathematics*, Cilt. 5, s. 1-14.
- [33] Alberdi, R., Murren, P., Khandelwal, K. 2015. Connection Topology Optimization of Steel Moment Frames Using Metaheuristic Algorithms, *Engineering Structures*, Cilt. 100, s. 276-292. DOI: 10.1016/j.engstruct.2015.06.014
- [34] 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*, Cilt. 135, s. 292-301. DOI: 10.1016/j.jcsr.2017.03.020
- [35] Çelik Yapıların Tasarım, Hesap ve Yapımına Dair Esaslar. 2018. Resmî Gazete, 15 Şubat 2018, Sayı 30333.
- [36] Sezer, S. 2021. Çelik Yapıların Kaynaklı Birleşimlerinin Metasezgisel Yöntemlerle Optimum Tasarımı, Afyon Kocatepe Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 110s, Afyonkarahisar.
- [37] Rao, R. V. 2016. Jaya: A Simple and New Optimization Algorithm for Solving Constrained and Unconstrained Optimization Problems, *International Journal of Industrial Engineering Computations*, Cilt. 7, s. 19-34. DOI: 10.5267/j.ijiec.2015.8.004
- [38] 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*, Cilt. 43, s. 303-315. DOI: 10.1016/j.cad.2010.12.015
- [39] Mirjalili, S. 2015. SCA: A Sine Cosine Algorithm for Solving Optimization Problems, *Knowledge-Based Systems*, Cilt. 96, s. 120-133. DOI: 10.1016/j.knosys.2015.12.022
- [40] MATLAB. 2016. version 9.0.0 (R2016a), The MathWorks Inc., Natick, Massachusetts.