

## Bir Ağır Vasıtanın Çok Akıllı Direksiyon Mekanizmasının Arı Algoritması Kullanılarak Optimizasyonu

A. Erdemir\*  
MPG Makine Prodüksiyon Grubu  
Makine İmalat San. ve Tic. A.Ş.  
Konya

M. Kalyoncu\*\*  
Selçuk Üniversitesi  
Konya

**Özet**—Bu çalışmada, bir ağır vasıtanın çok akıllı direksiyon mekanizmasının optimizasyonu yapılmıştır. Optimizasyonda Ackerman direksiyonlama hatasından türetilen hata fonksiyonunu minimize etmek için Arı Algoritması kullanılmıştır. Analitik bir matematiksel model kullanmak yerine CAD (Solidworks) programı tarafından sağlanmış çok akıllı direksiyon mekanizmasının katı modeli kullanılarak Ackerman hatası hesaplanmıştır. Bu sayede direksiyon mekanizmasının matematiksel modelinin elde edilmesindeki karmaşıklıktan kurtulmuş ve matematiksel model yerine doğrudan vasıtanın imalatında kullanılan CAD ortamındaki hali hazır gerçek katı model kullanılmıştır. Katı model kullanılarak gerçekleştirilen optimizasyon işlemi sürecinde her bir iterasyonda çok akıllı direksiyon mekanizmasının güncel konfigürasyonu kullanıcı tarafından görülebilmektedir. Ackerman direksiyonlama hatasının optimizasyonunda Arı Algoritmasını çalıştırabilmek için VisualBasic.NET dilinde bir yazılım geliştirilmiştir. Solidworks ile geliştirilen yazılım arasındaki bilgi aktarımı için Solidworks API (Uygulama Programlama Arayüzü) kullanılmıştır. Geliştirilen teknik beş akıllı bir ağır vasıtanın direksiyonlama sisteminin Ackerman direksiyonlama hatasının optimizasyonu için kullanılmıştır. Direksiyonlama açısı optimizasyonunda elde edilen nümerik sonuçlar grafiksel olarak verilmiş, tartışılmış ve yorumlanmıştır.

**Anahtar kelimeler:** Çok akıllı direksiyon mekanizması, çok değişkenli optimizasyon, Arı Algoritması, Solidworks API

**Abstract**—This paper presents an optimization scheme for a multi-axle heavy vehicle steering system. The Bees Algorithm is used in the optimization process in order to minimize the error function which is derived from the Ackerman steering error. Ackerman error is calculated by using the solid model of the multi-axle steering system obtained by a CAD program instead of using an analytical model. Since some assumptions are made in order to simplify the analytical model in mathematical formulation of the steering system, steering angle calculation by using an actual solid model is expected to give more accurate results. Solidworks CAD program is used in Ackerman steering error calculations.

By using the solid model, user can see the current configuration of the multi-axle steering system in each iteration of the optimization process. A software in VisualBasic.Net language is developed in order to implement Solidworks API is used in data transfer from Solidworks to the developed software. The developed technique is used in optimization of Ackerman steering error of a 5 axle heavy vehicle steering system. Obtained numerical results are discussed and comments on use of the presented technique in the steering angle optimization are presented.

**Keywords:** Multi Axle Steering, Multivariable Optimization, the Bees Algorithm, Solidworks API

### I Giriş

1970 yılından beri bilgisayar kaynaklarının gelişmesi ile beraber çok gövdeli karmaşık sistemlerin mekanizma sentezi problemleri çok gözde bir konu olmuştur [1]. Mekanizmaların karmaşıklaşması ile beraber analitik çözüm elde edilmesi de zorlaşmıştır.

Çok akıllı vasıtalar genellikle büyük boyutlu ve yüksek yük kapasiteli vasıtalarlardır. Yer ile vasıta arasındaki basıncı indirgeme gereksiniminden dolayı çok aksların kullanılması bir ihtiyaç duyulmuştur. Çok aksların kullanımı sonucunda da bu tip vasıtalarda, Ackerman hatası lastiklerin aşınmasına sebep olmaktadır. Bu, aynı zamanda aracın istenen yörüngede manevra yapamamasına ve manevra esnasında titreşimlere sebep olmaktadır. Çok değişkenli mekanizmalarda, istenilen yörüngenin takip edilebilmesi için, hata fonksiyonunu minimize eden değişken değerlerini tespit edebilmek gerekir. Bunu yapabilmek için, istenilen tolerans aralığında hatayı minimize eden değişken dizisi bulunana kadar değişken dizileri üretilmeli ve bunun getirdiği hata fonksiyonu hesaplanmalıdır. Bir optimizasyon işleminde n adet değişken olsun. Bu aslında n boyutlu bir uzaydaki grafiğin minimum değerini bulma işidir. Ve n boyutlu uzaydaki minimum değişken konfigürasyonunu öngörebilmek neredeyse imkânsızdır. Bu yüzden çok değişkenli optimizasyon tekniklerinin kullanılması gerekir.

Mekanizma sentezi 3 konu başlığından oluşmaktadır: Grafik, analitik ve optimizasyon metotlarıdır [2]. Eğer mekanizmanın az sayıdaki noktalardan geçmesi isteniyorsa grafik metodu hızlı bir çözüm bulur. Analitik metotta ise cebrik ifadelerle mekanizmanın matematiksel

\* abduherdemir@gmail.com

\*\* mkalyoncu@selcuk.edu.tr

modeli çözülerek hassas noktalardan geçecek şekilde sonuca varılır, fakat bu hassas noktalar birbirleri ile uyumlu değilse veya sayıca fazla ise analitik metot sonucunda karmaşık sayılar elde edilebilir [3]. Bu da sonucun fiziksel olarak üretilemeyeceği anlamına gelir. Nümerik teknikler genellikle çeşitli optimizasyon metotları ile beraber kullanılır. Sürü optimizasyon algoritmaları bu alanda çokça kullanılmaktadır. Sürü optimizasyon algoritmaları şunlardır:

1. Evrimsel Algoritmalar
2. Parçacık Sürü Optimizasyonu
3. Karınca Koloni Optimizasyonu
4. Arılardan esinlenen algoritmalar (Arı Sistemi (BE), Arı Koloni Optimizasyonu(BCO), Yapay Arı Kolonisi (ABC) ve Arı Algoritması (BA)) [4]

En çok kullanılan optimizasyon teknikleri Evrimsel Algoritması (EA), Genetik Algoritması (GA) ve Parçacık Sürü Optimizasyonu (PSO)'dur. Nüfus temelli optimizasyon teknikleri çoğunlukla global minimuma erişebilir. Ayrıca nüfus temelli optimizasyonlarda bir başlangıç şartına da ihtiyaç yoktur, sadece değişken aralığına ihtiyaç vardır [5].

Boyutsal sentezde 2 yaklaşım vardır: Hassas noktalar ve en uygun sentez [6], [7]. Hassas nokta sentezi mekanizmanın belli bir noktasının belli sayıdaki noktalardan geçmesini hedefler. Hedef noktaların sayısı arttıkça problem çok nonlineer bir hal alır ve çözmesi de çok zorlaşır. Analitik yöntemlerle mekanizmanın aşırı fazla noktadan geçmesi istendiğinde mekanizma boyutları karmaşık sayı şeklinde çıkabilir, bu da mekanizmanın gerçekte mümkün olmadığını gösterir. Diğer bir taraftan, en uygun sentez ise rastgele üretilmiş mekanizma boyutlarını kullanarak en iyi sonucu veren çözümlü bulma işidir ve boyutsal sentezde çokça kullanılmaktadır.

Hata fonksiyonuna, kısıt şartları da dahil edilebilir. Mesela; uzuv boyutunun negatif olamaması veya bir uzvun boyu, bir sayıdan büyük veya küçük olması istendiği durum gibidir. Bunu yapabilmek için hata fonksiyonunun içine cezalandırma fonksiyonları eklenebilir [6], [8], [9].

Optimizasyon birçok alternatif arasından en iyi çözümü bulma işlemidir. Optimizasyon, nümerik metotlara dayanır ve genellikle nonlineer problemler için kullanılırlar. Optimizasyonda yerel arama metotları ve global arama metotları vardır. Optimizasyon tekniklerinin çoğu kısıt şartlarının kullanımı üzerine kuruludur [8].

Prasanna Kumar G.V [10], Parçacık Sürü Optimizasyonunu ve çeşitlerini 5 uzuvlu bir mekanizmada karşılaştırmıştır. Bu çalışmada, 11 ila 25 arasındaki noktalardan oluşan yörünge sentezi çalışması yapılmıştır.

Bazı algoritmalar dizi sınırları içinde kalacak şekilde rastgele diziler üretir. Bu tip algoritmalar yerel minimum veya maksimumları bulmakla birlikte global minimum veya maksimumu da bulması mümkündür [13].

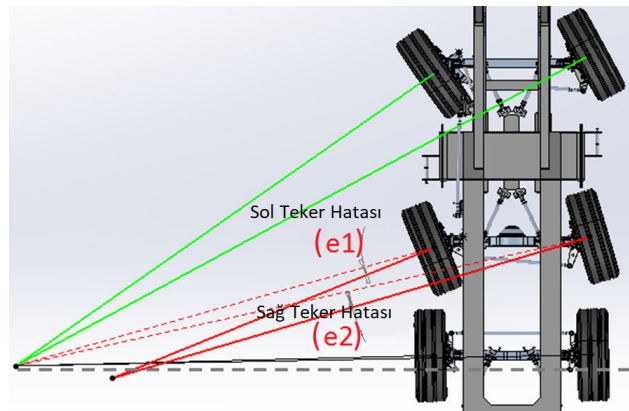
D.T. Pham ve Mete Kalyoncu [14], tek uzuvlu esnek robot kolunun bulanık mantık kontrolcü parametrelerinin optimizasyonunda Arı Algoritması kullanmışlardır. D. T. Pham ve M. Castellani[14], Arı Algoritması ile Parçacık Sürü Optimizasyonu, Evrimsel Algoritması ve Yapay Arı Koloni algoritmalarını karşılaştırmışlardır. Birçok karşılaştırmalarda en iyi sonucu Arı Algoritması, bunun ardından da Yapay Arı Kolonisi gelmiştir.

Qiang Long ve Changzhi Wu [15], Genetik Algoritması ile Hooke-Jeeves algoritmasını birleştirmişlerdir. Bu iki algoritma birlikte çalıştığında, daha iyi sonuçların elde edildiği görülmüştür. Yerel arama algoritmasının global optimizasyon algoritmasına yerel bölgelerde yardım ettiği görülmüştür. Coelho, Sacco ve Henderson [16] Metropolis algoritması ile Hooke-Jeeves algoritmasından bir kombinasyon oluşturmuş ve bu optimizasyon işlemi hızlandırmıştır.

Kuang-Hua Chang ve Sung-Hwan Joo [17], bir aracın dinamik etkiler altında süspansiyon konumlarının CAD programları ile optimize edilmesi çalışması yapmışlardır. Bu çalışmalarında ProEngineer ve Solidworks API kullanarak CAD modelinden yararlanmışlardır.

Bu çalışmada, direksiyonlama mekanizması Solidworks CAD programında modellenmiş olup, bu mekanizma VisualBasic.NET kullanılarak Arı Algoritması ile optimize edilecektir. Hata fonksiyonu, Solidworks API aracılığıyla CAD modeli üzerinden direkt olarak hesaplanacaktır. Mekanizma optimizasyonu işleminde Arı Algoritması kullanılarak global minimum ve komşu yerel minimum alternatif çözümleri de bulunabilecektir. Diğer bir ifadeyle, Ackerman hatasını minimize eden birden fazla CAD konfigürasyonu bulunmuş olacaktır.

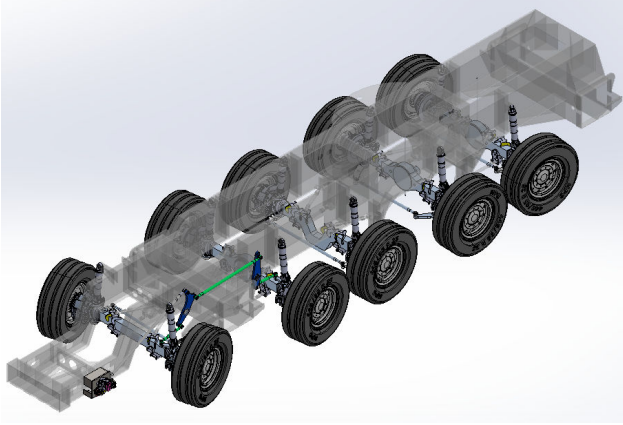
Ackerman direksiyonlama hatası, hedeflenen tekerlek doğrultusu ile gerçekleşen tekerlek doğrultusu arasındaki açıdır [18-21]. Genellikle bu hatanın  $1^\circ$  veya  $1,5^\circ$ 'nin altında olması istenir. Bizim bu çalışmadaki hedefimiz  $1^\circ$ 'nin altına inebilmektir. Direksiyonlama hatası aşağıdaki şekilde gösterilmiştir.



Şekil 1. Ackerman hatası

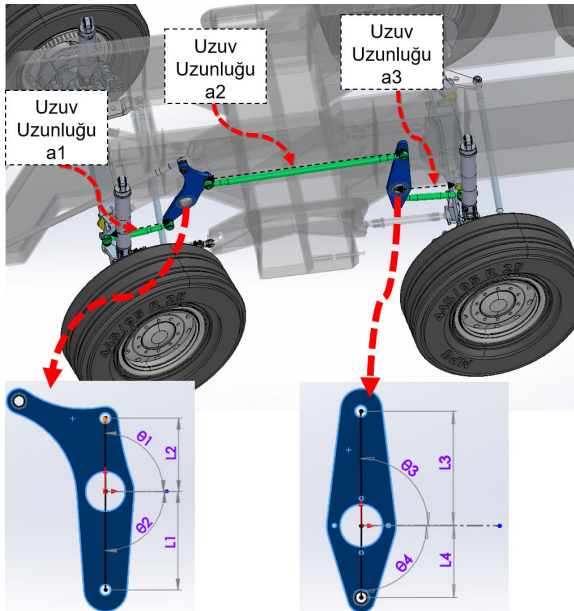
## II. CAD Parametreleri ve Cezalandırma Fonksiyonları

Şekil 2 de görüldüğü gibi uygulama yapılan ağır vasıtada 5 aks vardır.



Şekil 2. Çok akslı ağır vasıta

Çok akslı ağır vasıtalar, tüm teker direksiyonlama (all-wheel-steering) ve yengeç direksiyonlama (crab-steering) modlarına ihtiyaç duyar. Bu iki modu birden sağlayabilmesi için aracın üçüncü, dördüncü ve beşinci akslarının birinci ve ikinci akslardan mekanik olarak bağımsız olması gerekmektedir. Bu durumda sadece birinci ve ikinci akslar mekanik olarak bağlantılıdır. Fakat üçüncü, dördüncü ve beşinci akslar elektro hidrolik eyleyiciler tarafından tahrik edilmektedir. Bu sentez çalışmasında mekanik olarak bağımlı olan birinci ve ikinci aks arasındaki direksiyon mekanizması optimize edilecektir.



Şekil 3. Dizayn parametreleri

CAD parametreleri şunlardır: a1, a2, a3, L1, L2, L3, L4, θ1, θ2, θ3 ve θ4. Bu parametrelerin başlangıç şartları Tablo 1 de verilmiştir.

Parametreler	Başlangıç Şartları
a1 [mm]	787,80
L1 [mm]	250,60
θ1 [°]	89,00
L2 [mm]	294,38
θ2 [°]	89,62
a2 [mm]	1457,36
L3 [mm]	415,31
θ3 [°]	90,00
L4 [mm]	200,00
θ4 [°]	89,78
a3 [mm]	487,19

TABLO 1. Başlangıç şartları

Bir değişken dizisinin fiziksel olarak mümkün olmayan mekanizmaları önlemek için cezalandırma fonksiyonları kullanılmıştır. Bu cezalandırma fonksiyonları aktif konfigürasyonu kontrol eder ve şarta bağlı olarak hatayı artırır.

Mesela, a1 değişkenine ait sınır şartları şu şekildedir:

$$623 \leq a1 \leq 823$$

Bu sınır şartlarının uyulmaması durumunda ceza katsayısı 1000 olsun. Bu durumda hata fonksiyonu:

$$g1 = (a1 < 623) * 1000$$

Yukarıdaki ifadede parantez içindeki değer doğru ise 1 değerini üretir, bu durumda g1 1000 olur. Eğer parantez içindeki değer doğru değilse 0 değerini üretir ve bu durumda da g1 değeri 0 olur.

Tüm cezalandırma fonksiyonları Tablo 2 de verilmiştir. Tablo 2 deki fonksiyonlara, eğer varsa başka cezalandırma fonksiyonları da eklenebilir.

$g1 = (a1 < 623) * 1000$	$g12 = (a > 823) * 1000$
$g2 = (L1 < 170) * 1000$	$g13 = (L1 > 370) * 1000$
$g3 = (\theta1 < 84^\circ) * 1000$	$g14 = (\theta1 > 96^\circ) * 1000$
$g4 = (L2 < 100) * 1000$	$g15 = (L2 > 300) * 1000$
$g5 = (\theta2 < 84^\circ) * 1000$	$g16 = (\theta2 > 96^\circ) * 1000$
$g6 = (a2 < 1290) * 1000$	$g17 = (a2 > 1490) * 1000$
$g7 = (L3 < 200) * 1000$	$g18 = (L3 > 450) * 1000$
$g8 = (\theta3 < 84^\circ) * 1000$	$g19 = (\theta3 > 96^\circ) * 1000$
$g9 = (L4 < 91) * 1000$	$g20 = (L4 > 291) * 1000$
$g10 = (\theta4 < 84^\circ) * 1000$	$g21 = (\theta4 > 96^\circ) * 1000$
$g11 = (a3 < 390) * 1000$	$g22 = (a3 > 590) * 1000$

TABLO 2. Cezalandırma fonksiyonları

### III. Hata Tanımı ve Ağırlıklar

Direksiyonlama hatası Şekil 1'deki gibi tanımlanmıştır. Bu mekanizmada direksiyonlama hatasının her durumda  $1^\circ$ 'nin altında olması istenir. Yaklaşık her  $5^\circ$ 'de bir hata fonksiyonu hesaplanacaktır. Ve elde edilen hataların karelerinin toplamının karekökü alınarak ortak bir hata elde edilecektir.  $e_1$  ikinci aksın sol tekerine ait direksiyonlama hatası olsun ve  $e_2$  de ikinci aksın sağ tekerine ait direksiyonlama hatası olsun. Bu durumda hata aşağıdaki gibidir:

$$e = \sqrt{\sum_{i=1}^n (e_1^2 + e_2^2)_i} + \sum_{j=1}^m g_j \quad (1)$$

Bazı hata bölgelerinin önemli olduğu durumlarda, ağırlık çarpanlarının kullanılması gerekir. Mesela, direksiyonlama hatasının her durumda  $1^\circ$ 'nin altında olmasını istendiği için, burada  $1^\circ$ 'nin üzerindeki hatalar önem arz etmektedir. Eğer hata fonksiyonunu şu şekilde değiştirilirse; bu hata fonksiyonu  $1^\circ$ 'nin üzerindeki hataları azaltmaya daha fazla odaklanacaktır.

$$e_1 = \begin{cases} 1 + 10(e_1 - 1), & e_1 > 1 \\ e_1, & e_1 \leq 1 \end{cases} \quad (2a)$$

$$e_2 = \begin{cases} 1 + 10(e_2 - 1), & e_2 > 1 \\ e_2, & e_2 \leq 1 \end{cases} \quad (2b)$$

2a ve 2b denklemlerine göre hatanın  $1^\circ$ 'nin üzerindeki kısmı 10 ile ağırlıklı olarak toplanır.

$$e_1 = \begin{cases} 1 + 10(e_1 - 0.80), & e_1 > 1 \\ e_1, & e_1 \leq 1 \end{cases} \quad (3a)$$

$$e_2 = \begin{cases} 1 + 10(e_2 - 0.80), & e_2 > 1 \\ e_2, & e_2 \leq 1 \end{cases} \quad (3b)$$

$$e = \sqrt{\sum_{i=1}^n (e_1^2 + e_2^2)_i} + \sum_{j=1}^m g_j \quad (3c)$$

3a ve 3b denklemlerinde, eğer hata  $1^\circ$ 'nin altında ise hata aynen alınmakta fakat hata  $1^\circ$ 'nin üzerinde ise  $0.80^\circ$  altındaki hata aynen alınır ve buna ek olarak da  $0.80^\circ$ 'nin üzerindeki kalan kısmı da 10 kat ağırlıklı olarak toplanır. Bu durumda eğer direksiyonlama hatası  $1^\circ$ 'nin üzerinde olursa 3a ve 3b'deki hata fonksiyonu 2a ve 2b'deki fonksiyonuna göre daha fazla hata sonucu üretecektir.

Birinci aksın sol tekeri  $35^\circ$  sol ile  $28,5^\circ$  sağ aralığında çalışmaktadır. Toplam açı menzili  $35 + 28.5 = 63,5^\circ$ 'dir.  $35^\circ$ 'den  $-28,5^\circ$ 'ye kadar 13 adım oluşturulmuştur. Her

bir adım  $63,5 / 12 = 5.29^\circ$ 'dir. Hata fonksiyonundaki n değeri 13'tür. m ise cezalandırma fonksiyonlarının sayısıdır ve bu problemde 22 tane cezalandırma fonksiyonu olduğu için değeri 22'dir.

### IV. Optimizasyon Algoritması

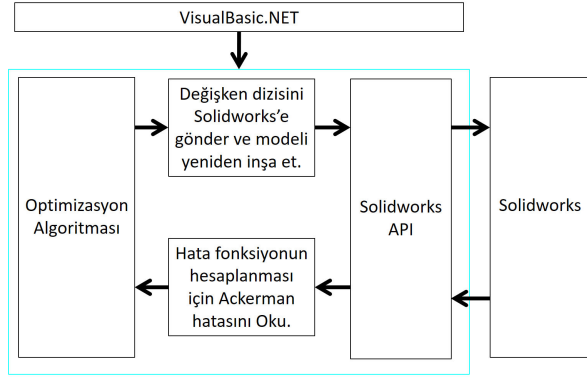
Arı Algoritması bir global arama algoritmasıdır. Algoritma arama uzayına rastgele yerleşmiş n adet keşif arısı ile başlar. Bu arıların her birinin hatası hesaplanır. Bu arıların oluşturduğu hata dizisi küçükten büyüğe doğru sıralanır. m adet en iyi arı, komşuluk araştırması için seçilir. Bu en iyi arılar etrafında ngh radius komşuluğunda yeni diziler üretilir ve bunların hata sonuçları hesaplanır. Fakat bu m adet arı içindeki e adet en iyi olanı kıdemli sonuçlardır ve bu sonuçların etrafına daha fazla arı gönderilir. Bu m arıdan her birinin etrafında gönderilen araştırmacı arılar da kendi aralarında tekrar küçükten büyüğe doğru sıralanır ve en küçük olanı en iyi arının yerini alır. Kalan (n-m) arı yerine, sınır şartları içindeki uzaya rastgele yayılmış arılar üretilir. Ve tüm arıların hatası tekrar en küçükten büyüğe doğru sıralanır ve döngü durma şartı ile karşılaşıncaya kadar devam eder. Arı Algoritmasının en basit hali ile ifadesi aşağıdaki gibidir:

Kısıt şartları dâhilindeki uzayda rastgele üretilmiş bir nüfus oluştur.

1. Nüfusun hatasını hesapla ve en küçükten en büyüğe doğru sırala.
2. Durdurma kriteri gerçekleşmediği sürece döngüye devam et.
3. En iyi nüfusa ait bölgeleri komşuluk araştırması için seç.
4. En iyi nüfus bölgelerinin komşuluklarına arılar gönder (kıdemli e adet bölgeye daha fazla arı gönder) ve her bir arının hatasını hesapla.
5. Her bir komşuluk grubunun hatasını en küçükten en büyüğe doğru sırala.
6. Kalan arıları kısıt şartları dâhilinde uzaya tekrar rastgele dağıt ve hatalarını hesapla
7. Döngüye tekrar başla (2. maddeye git)

### V. Uygulama Arayüzü

Bu optimizasyon için kullanılacak yazılım VisualBasic.NET kullanılarak geliştirilmiştir ve bu yazılım ile Solidworks CAD programı arasındaki iletişim Solidworks API aracılığıyla sağlanmıştır. Yazılım ölçü dizileri üretmekte ve Solidworks'e bu dizileri göndermektedir. Direksiyonlama mekanizması soldan sağa doğru direksiyonlama esnasında kontrol edilmekte ve eğer mekanik bir hata yoksa yazılım Ackerman hatalarını okumaktadır.



Şekil 4. Geliştirilen yazılım

## VI. Sonuçlar

Bu çalışmada, uygulama yapmak için kullanılan bilgisayar özellikleri ve programlar şu şekildedir:

- CPU: Intel Xeon CPU E3-1270- v3 @ 3.50 GHz
- RAM: 16 GB hafıza
- CAD Programı: Solidworks 2014 Premium
- Yazılım Geliştirme: Microsoft Visual Studio 2008
- İşletim sistemi: Microsoft Windows 7 Professional

Hata fonksiyonu (3c) 'ye göre Arı Algoritması sonuçları Tablo 3 te verilmiştir. Elde edilen uygulama sonuçlarına ait tasarım parametreleri ise Tablo 4 de verilmiştir.

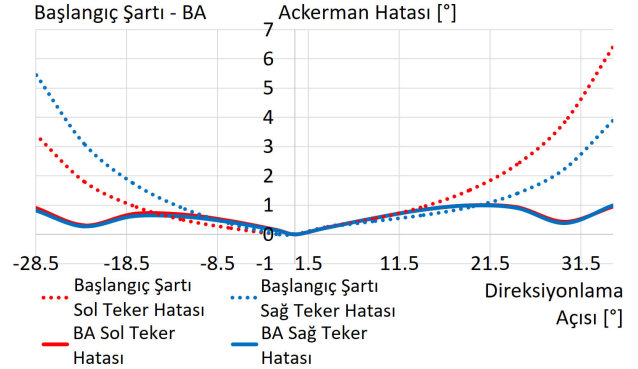
Hata Sonuçları	Orijinal	BA
<b>Toplam hata</b>	222,14	11,18
<b>Ortalama sol teker hatası</b>	1,75°	0,62°
<b>Ortalama sağ teker hatası</b>	1,65°	0,59°
<b>En yüksek sol teker hatası</b>	6,39°	0,99°
<b>En yüksek sağ teker hatası</b>	5,46°	1,00°

TABLO 3. Optimizasyon Sonuçları (Ackerman Hataları)

Parametre	Orijinal	BA
<b>Toplam Hata</b>	222,14	11,18
<b>a1 [mm]</b>	787,8	781,33
<b>L1 [mm]</b>	250,6	352,56
<b>θ1 [°]</b>	89	88,83
<b>L2 [mm]</b>	294,38	230,61
<b>θ2 [°]</b>	89,62	93,56
<b>a2 [mm]</b>	1457,36	1466,52
<b>L3 [mm]</b>	415,31	274,6
<b>θ3 [°]</b>	90	94,52
<b>L4 [mm]</b>	200	189,08
<b>θ4 [°]</b>	89,78	92,1
<b>a3 [mm]</b>	487,19	529,43

TABLO 4. Optimizasyon sonucu eski parametreler ve elde edilen yeni parametreler

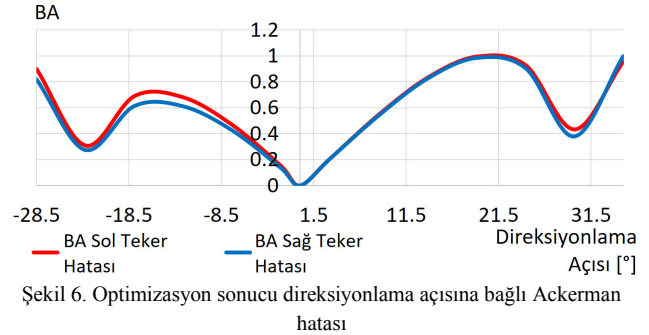
Dönüş esnasında orijinal Ackerman hatası ve optimize edilen Ackerman hatası kıyaslaması Şekil 5'te görülmektedir.



Şekil 5. Optimizasyon sonucu direksiyonlama açısına bağlı Ackerman hatası (Orjinal ve Arı Algoritması Karşılaştırma)

Sadece Arı Algoritmasının bulduğu sonuca ait Ackerman hatası Şekil 6 de görülmektedir.

Şekil 6'da görüldüğü gibi optimizasyon sonrasında, tüm direksiyonlama açılarında Ackerman hatası 1° 'nin altına indirilmiştir.



## VII. Tartışma

Boyutsal sentezde doğrudan CAD modeli kullanmak, optimizasyon işleminin hazırlık aşamasını hızlandırmaktadır. CAD model kullanmak aynı zamanda optimizasyon sonucun simülasyonunu da yapmış olmaktadır. Ayrıca CAD modellerinin kullanılması yanlış matematiksel model geliştirme riskini de ortadan kaldırmaktadır. Direksiyonlama hatasının tüm durumlarda 1° 'nin altında kalmasını sağlamak için, hata fonksiyonuna 0.80° ve üzerine ağırlık verilmiştir. Hata fonksiyonu her durumundaki Ackerman hatasının toplamına dayalı bir fonksiyon olduğundan dolayı, 1° üzerine ağırlık verilmemesi durumunda bazı direksiyonlama açılarında hata 2° ve bazı direksiyonlama açılarında da 0° 'ye yakın olup hata fonksiyonunun düşük sonuç vermesi optimizasyon amacının gerçekleştirilmesini engellemektedir. Özellikle toplama dayalı hata fonksiyonlarında ağırlık kullanmanın önemli olduğu görülmüştür.

## Teşekkür

Bu çalışmada optimizasyonu yapılan çok akslı ağır vasıtaya ait direksiyonlama mekanizmasının gerçekleştirilip denemesi için gerekli verilerin teminindeki katkı ve desteklerinden dolayı tüm MPG Makine İmalat Sanayi ve Tic. A.Ş. çalışanlarına ve Genel Müdürü Veysel ALVER 'e teşekkür ederiz.

Bu bildiri Abdullah ERDEMİR'in yüksek lisans tez çalışmalarından türetilmiştir.

## Kaynakça

- [1] Ouezdou, F. B., and S. Regnier. "General method for kinematic synthesis of manipulators with task specifications." *Robotica* 15(06): 653-661,15 June 1997
- [2] Sheu, Jinn-Biau, Sheng-Lun Hu, and Jyh-Jone Lee. "Kinematic synthesis of a four-link mechanism with rolling contacts for motion and function generation." *Mathematical and Computer Modelling* 48.5:805-817, 2008.
- [3] Peñañuri, F., et al. "Synthesis of mechanisms for single and hybrid tasks using differential evolution." *Mechanism and Machine Theory* 46(10): 1335-1349,2011.
- [4] Yuce, Baris, et al. "Honey Bees Inspired Optimization Method: The Bees Algorithm." *Insects*,4(4):646-662,2013.
- [5] Portilla Flores Edgar A., Morales-Cruz Cuauhtemoc Suarez Santillan Rosaura A., Villarreal-Cervantes Miguel G., Nino Suarez Paola A., and Calva Yanez Maria B., "Optimal Synthesis of The Continuously Variable Transmission Input Mechanism Using An Evolutionary Approach", *International Journal of Innovative Computing, Information and Control* Volume 9, Number 5, ICIC International 2013 ISSN 1349-4198, May 2013.
- [6] Bulatović, Radovan R., and Stevan R. Đorđević. "Optimal synthesis of a four-bar linkage by method of controlled deviation." *Theoretical and applied mechanics* 31.(3-4):265-280,2004.
- [7] Collard, Jean-François, Pierre Duysinx, and Paul Fiset. "Optimal synthesis of planar mechanisms via an extensible-link approach." *Structural and Multidisciplinary Optimization* 42(3):403-415,2010.
- [8] Zhang, Wei, Xudong Shi, and Liwen Wang. "Lamarckism and mechanism synthesis: approaching constrained optimization with ideas from biology." *arXivpreprint arXiv:1109.6717*,2011.
- [9] Bulatović, Radovan R., and Stevan R. Đorđević. "Optimal synthesis of a path generator six-bar linkage." *Journal of mechanical science and technology* 26(12): 4027-4040,2012.
- [10] Prasanna Kumar G. V., "Comparative Evaluation of Particle Swarm Optimization Algorithms for the Optimal Dimensional Synthesis of Planar Four-bar Mechanism", *Invited Lecture on Dynamic Analysis of Mechanisms* ,2013.
- [11] Fernandes, Florbela P., et al. "Multi start Hooke and Jeeves filter method for mixed variable optimization." 2013.
- [12] Luo, Youxin, et al. "Hyper-chaotic mapping newton Iterative method to mechanism synthesis." *Strojnikivestnik* 54(5): 372-378,2008.
- [13] Moser, Irene. "Hooke-Jeeves revisited." *Evolutionary Computation*, 2009. CEC'09. IEEE Congress on. IEEE, 2009.
- [14] Pham, Duc Truong, and M. Castellani. "The bees algorithm: modelling foraging behavior to solve continuous optimization problems." *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*,223(12):2919-2938,2009.
- [15] Long, Qiang, and Changzhi Wu. "A HYBRID METHOD COMBINING GENETIC ALGORITHM AND HOOKE-JEEVES METHOD FOR CONSTRAINED GLOBAL OPTIMIZATION." *MANAGEMENT* 10(4): 1279-1296, 2014.
- [16] Rios-Coelho, A. C., W. F. Sacco, and N. Henderson. "A Metropolis algorithm combined with Hooke-Jeeves local search method applied to global optimization." *Applied Mathematics and Computation* 217(2): 843-853,2010.
- [17] Chang, Kuang-Hua, and Sung-Hwan Joo. "Design parameterization and tool integration for CAD-based mechanism optimization." *Advances in Engineering Software* 37(12):779-796,2006.
- [18] Qin, Gang, et al. "Analysis and Optimization of the Double-Axle Steering Mechanism with Dynamic Loads." *The Open Mechanical Engineering Journal* 5:26-39,2012.
- [19] Zhou, Bing, Dongsheng Li, and Fan Yang. "Optimization design of steering linkage in independent suspension based on genetic algorithm." *Computer-Aided Industrial Design & Conceptual Design*, 2009. CAID & CD 2009. IEEE 10th International Conference on. IEEE, 2009.
- [20] Felzien, Mark L., and D. L. Cronin. "Steering error optimization of the MacPherson strut automotive front suspension." *Mechanism and machine theory* 20(1): 17-26,1985.
- [21] Rahmani Hanzaki, A., P. V. M. Rao, and S. K. Saha. "Kinematic and sensitivity analysis and optimization of planar rack-and-pinion steering linkages." *Mechanism and Machine theory* 44(1) :42-56,2009.