

## Değiştirilmiş Karınca Kolonisi Optimizasyon Algoritması ile Redüktör Tasarımının Simülasyonu

### Simulation of Speed Reducer Design with the Modified Ant Colony Optimization Algorithm

<sup>1</sup>Kürşat TANRIVER , <sup>2</sup>Mustafa AY 

<sup>1</sup>Marmara Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul, Türkiye

<sup>2</sup>Marmara Üniversitesi, Teknoloji Fakültesi, İstanbul, Türkiye

<sup>1</sup>k.tanriver@hotmail.com, <sup>2</sup>muay@marmara.edu.tr

Araştırma Makalesi/Research Article

#### ARTICLE INFO

##### Article history

Received :11 February 2024

Accepted: 10 March 2024

##### Keywords:

Ant Colony, Engineering Design, MATLAB, Optimization, Speed Reducer Design

#### ABSTRACT

This paper focuses on the innovative application of the Modified Ant Colony Optimization (DEKKO) algorithm for the optimization of speed reducer engineering problem. The main contribution of this study is the development of DEKKO, which combines the advantageous features of Ant Colony Algorithm (KKO). The aim of DEKKO is to achieve better results than those previously solved with different techniques in the literature by modifying KKO. The algorithm was run 20 times until the most effective result was achieved, with the best performance outcome of 3105. 8779 vector unit obtained at 14 iterations. This process utilized 100 ants and was completed in 66.81 seconds. When compared with similar results in the literature, DEKKO has achieved success with a solution that stands out among the literature results. Users can easily obtain information about speed reducer design and pre-production through simulation using the DEKKO algorithm. This aims to contribute to cost and time savings.

© 2024 Bandırma Onyedi Eylül University, Faculty of Engineering and Natural Science. Published by Dergi Park. All rights reserved.

#### MAKALE BİLGİSİ

##### Makale Tarihleri

Gönderim : 11 Şubat 2024

Kabul : 10 Mart 2024

##### Anahtar Kelimeler:

Karınca Kolonisi, Mühendislik Tasarımı, MATLAB, Optimizasyon, Redüktör Tasarımı

#### ÖZET

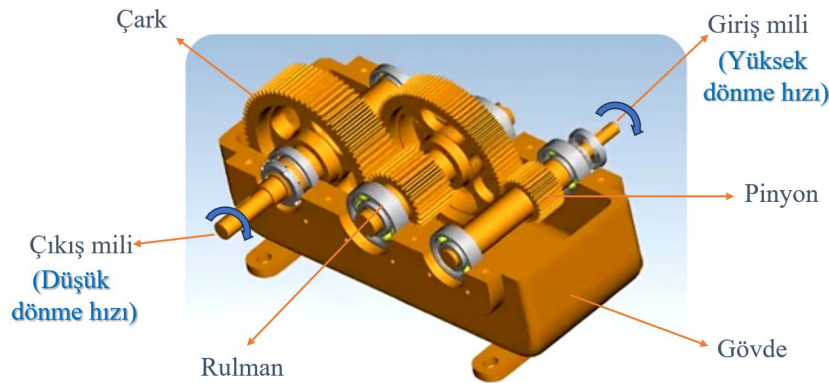
Bu makale, değiştirilmiş karınca kolonisi optimizasyonu (DEKKO) algoritmasının redüktör mühendislik probleminin çözümüne yeniden odaklanılmasına dayanmaktadır. DEKKO, Karınca Kolonisi Algoritmasının (KKO) avantajlı özelliklerinin birleştirilmesiyle oluşturulmuştur. DEKKO ile KKO' da değişiklik yapılarak daha önceden literatürde farklı tekniklerle yapılan çalışmalardan daha iyi sonuçların elde edilmesi amaçlanmıştır. Algoritma, en etkili sonuç elde edilene kadar 20 kez çalıştırılmıştır. İterasyon sayısı 14 olmak üzere en iyi performans sonucu olarak 3105,8779 vektör birim sonucu elde edilmiştir. Bu işlem, algorithmada 100 adet karınca kullanılarak 66,81 saniyede tamamlanmıştır. Literatürdeki sonuçlarla karşılaştırıldığında DEKKO, literatür sonuçları arasında olduğu ve başarılı bir çözümle sonuçlandırıldığı gözlemlenmiştir. Kullanıcılar, DEKKO algoritmasını kullanarak simülasyon yoluyla redüktör tasarımı ve ön üretimi hakkında kolaylıkla bilgi edinebilmektedir. Böylelikle maliyet ve zaman tasarrufu açısından kullanıcılara katkıda bulunulması amaçlanmıştır.

© 2024 Bandırma Onyedi Eylül Üniversitesi, Mühendislik ve Doğa Bilimleri Fakültesi. Dergi Park tarafından yayınlanmaktadır. Tüm Hakları Saklıdır.

## 1. GİRİŞ

Optimizasyon, belirlenen hedefler ve kısıtlamalar içinde en iyi sonucu elde etmek için genellikle kullanılan bir yöntemdir. Optimizasyon adımları, temel yapılandırmaları oluşturmayı, tasarım değişkenlerini tanımlamayı, amaç fonksiyonunu formüle etmeyi, uygun optimizasyon problemlerini seçmeyi ve uygulamayı içermektedir [1]. Optimizasyon teknikleri genellikle matematiksel ve meta-sezgisel yöntemler olmak üzere iki kategoriye ayrılmaktadır. Matematiksel yöntemler, en doğru analitik çözümü bulmayı amaçlarken, meta-sezgisel yöntemler daha pratik bir yaklaşım sergilemektedir [2-3]. Mühendislik problemlerini çözmek için bazı meta-sezgisel yöntemler sıklıkla kullanılmaktadır [4]. Bunlar arasında karınca kolonisi optimizasyonu (KKO) [6], genetik algoritma (GA) [6], parçacık sürü optimizasyonu (PSO) [7], yarasa algoritması (YA) [8], ateşböceği algoritması (ABA) [9], kelebek algoritması (KA) [10], yapay arı kolonisi (YAK) [11], yapay bağışıklık sistemi (YBS) [12] ve kızıl tilki optimizasyonu (KTO) [13] gibi algoritmalar bulunmaktadır. Görüldüğü gibi meta-sezgisel yaklaşımlar içinde karınca, yarasa ve ateşböceği gibi hayvanların içgüdüsel davranışları üzerine çalışmalar yapılabilmektedir. Bu içgüdüler, karıncaların yiyecek arama, kuşların toplanma ve balıkların sürüler halinde hareket etme gibi doğal davranışlarını organize etmelerine olanak tanımaktadır [14]. Bu bağlamda bilim insanları böcek davranışlarını incelemiş ve başarılı optimizasyon algoritmaları geliştirmişlerdir. Mühendislik problemlerini çözmede yüksek esneklik düzeyine sahip bu teknikler, birçok bilimsel alan ve mühendislik problemlerinin çözümünde başarıyla uygulanmıştır [15].

Son yıllarda çözülen problemin özelliklerine ve karmaşıklığına bağlı olarak meta-sezgisel algoritmaların nispeten daha iyi yönlerini bir araya getirilerek karma ve iyileştirilmiş algoritmalar oluşturulmuştur [16]. Bu algoritmalar redüktör tasarımı gibi mühendislik problemlerini çözümünde sıklıkla kullanılmıştır [17]. Havacılık ve imalat sanayindeki aktarma sistemlerini içeren farklı türdeki mekanik ekipmanların çalışma verimliliği ve hassasiyeti dikkate alındığında redüktör tasarımı dikkat çekmektedir [18]. Redüktörler, aktarma sistemlerindeki hızı azaltarak yüksek tork, yüksek verimlilik ve düşük gürültü sağlamaktadır. Temel prensip olarak bir redüktörün giriş miline yüksek hızda gelen dönme hareketi, piyon ve çark yardımıyla çıkış miline daha düşük hızlarda aktarılabilmektedir. Örnek bir redüktör tasarımı Şekil 1’ de gösterilmiştir [19].



Şekil 1. Örnek bir redüktör tasarımı [19].

Optimizasyon problemlerine uygulanabilecek başarılı meta-sezgisel uygulamalardan biri KKO algoritmasıdır [13]. Optimizasyon problemlerine çözüm bulmak, mühendislik problemlerinin yüksek uygulama alanları ve değişkenliği nedeniyle zorlaşmaktadır. Bunun kolaylaştırılması için bilgisayar programlarının [21] dolayısıyla MATLAB [22] gibi programlardan dillerinin hesaplama gücünden faydalanılmaktadır.

Bu makaledeki çalışma, redüktör tasarımı gibi mühendislik tasarım problemlerini karınca kolonisi optimizasyonu (KKO) kullanarak çözme fikri ile oluşturulmuştur. Karınca kolonisi algoritması, literatürde farklı tekniklerle önceden çözülen bu tür problemlerden daha iyi sonuçlar almak için değiştirilmiştir. DEKKO adı verilen bu yeni algoritma, MATLAB yazılımı ile çalıştırılmış ve sonuçlar literatürdeki diğer tekniklerle karşılaştırılmıştır. DEKKO adı verilen bu çalışmanın literatüre katkısı ve avantajlı yönleri aşağıda sıralanmıştır.

- Bu yeni algoritma, KKO 'nun avantajlı yönlerini birleştirerek elde edilmiştir.
- Önerilen algoritma, uyarlanabilir parametreler ve özel bir prosedür kullanarak karınca feromon izinden en iyi sonuç elde edilene kadar çözüm güncellenmektedir. Böylece DEKKO hassas ayarlanmış parametrelerle daha az hesaplama yapmakta ve daha kısa sürede daha iyi sonuçlara ulaşabilmektedir.
- Kullanıcılar DEKKO algoritması kullanılarak simülasyon yardımıyla redüktör tasarımı ve üretimi öncesinde kolayca bilgi edinebilmektedirler.
- Yeni algoritma ile endüstri, profesyoneller ve bu alanda çalışan kullanıcılar için maliyet ve zaman tasarrufuna katkı sağlanmaktadır.
- DEKKO algoritmasının kodu, MATLAB yazılımı kullanılarak işlevselliği test edilmiş ve literatürdeki sonuçlara göre işlevselliği açıkça paylaşılmıştır.

- Ayrıca DEKKO, diğer algoritmaların güçlü yanları ile birleştirilerek hibrit bir algoritma oluşturulması yoluyla daha iyi sonuçlar alınabilme potansiyeline sahiptir.

## 2. LİTERATÜR TARAMASI

Üretim ve havacılık endüstrisinde çeşitli zorluklarda mühendislik problemleri bulunmaktadır. Bu zorlu mühendislik problemlerinin geniş uygulama alanları nedeniyle bunların çözümleri için birçok çalışma yapılmıştır [23]. Problemlerdeki değişken sayısının artmasıyla beraber geleneksel yöntemlerle çözüm bulmak zorlaşmıştır. Çözümü basitleştirmek ve en iyi çözümü bulmak için çaba harcanmış ve bu da meta-sezgisel algoritmaların geliştirilmesine yol açmıştır. Bu türden bir meta-sezgisel algoritma olan KKO [24], günlük yaşamın farklı alanlarında birçok uygulama problemini çözmek için kullanılmıştır [25]. KKO ilk kez 1930'da analitik olarak ifade edilmiş ve 1950'den sonra sıkça kullanılmaya başlanmıştır. O zamandan beri çeşitli problemleri başarıyla çözmek için kullanılmıştır [26].

Redüktör tasarımı gibi mühendislik problemleri üzerine bir literatür taraması yapılmıştır. Yapılan araştırmada amaç fonksiyonları, kısıtlamaları ve değişkenleri dikkate alınarak bu problemlerin çeşitli optimizasyon tekniklerini kullanarak çözülmeye çalışıldığı görülmüştür. Çözümlerin geliştirilmesi için bu çalışmalar incelenmiş ve aşağıda verilmiştir.

Yay Tasarımı daha önce Ray & Saini [27], Belegundu [28], Ray & Liew [29], Grković & Bulatović [30], Coelho & Mariani [31] ve Cagnina ve ark. [32] tarafından tanımlanmıştır. Tasarım problemi, yayın ağırlığını minimuma indirirken minimum eğilme, kayma gerilmesi, salımlı frekans ve dış çap ile tasarım değişkenlerinde belirli kısıtlamalara uygun olmasını içermektedir.

Kaynaklı Kiriş daha önce Ray & Liew [29], Grković & Bulatović [30], Cagnina ve ark. [32] ve Rao [33] tarafından tanımlanmıştır. Sunulan tasarım problemleri, kirişteki eğilme gerilmesi, bükülme gerilmesi, kirişin burkulma yükü ve kirişin uç eğilimi nedeniyle minimum maliyetin belirlenmesini amaçlamaktadır.

Redüktör tasarımında yapılan bazı çalışmalar incelenmiştir. Örneğin; Li ve ark. [34], redüktör içindeki akış alanının dağılımını sayısal olarak simüle etmişlerdir. Bu simülasyonda dişli hızını belirleyen yasaları, yağlayıcının dinamik viskozitesi ve ısı iletkenlik katsayısının dişli ısı dağıtım verimliliği üzerindeki etkilerini ortaya koymuşlardır. Maccioni ve ark. [35], yaptıkları çalışmada yenilikçi bir redüktör tasarımı sunmuş ve farklı redüktör tasarımlarıyla karşılaştırılmışlardır. Ayrıca yeni tasarımın avantajları ortaya koymuşlardır. Farhat ve ark. [36], yaptıkları çalışmada çalısır ve azalı durumdaki redüktörlerin üzerinde bir modelleme yaparak bir araştırma sunmuşlardır. Burada hız, yük değişimlerinin performansın nasıl etkileyeceğini incelemişlerdir. Bunun için hızlı Fourier dönüşümü, kare zarf spektrumu, kısa Süreli Fourier dönüşümü ve kurtogram metodlarını kullanmışlardır. Tatar [37], yaptığı çalışmada robotik sistemler için içinde redüktör bulunan tahrik ünitesi tasarlamıştır. Bu işlemler için MATLAB/Simulink programı kullanılmıştır. Demir & Güner [38], MATLAB yazılımı kullanarak geometrik bir redüktör model tasarımı yapmışlardır. Sonlu elemanlar analizi yapılan modelin mukavemet değerini incelemişlerdir. Koçak ve ark. [39], yaptıkları çalışmada geleneksel bir redüktör ile yeni tasarladıkları redüktörü deneysel olarak karşılaştırmışlardır. Deneyler sonucunda yeni tasarımın sabit devir ve frenleme momenti açısından geleneksel redüktöre göre daha yüksek verim sergilediğini belirtmişlerdir.

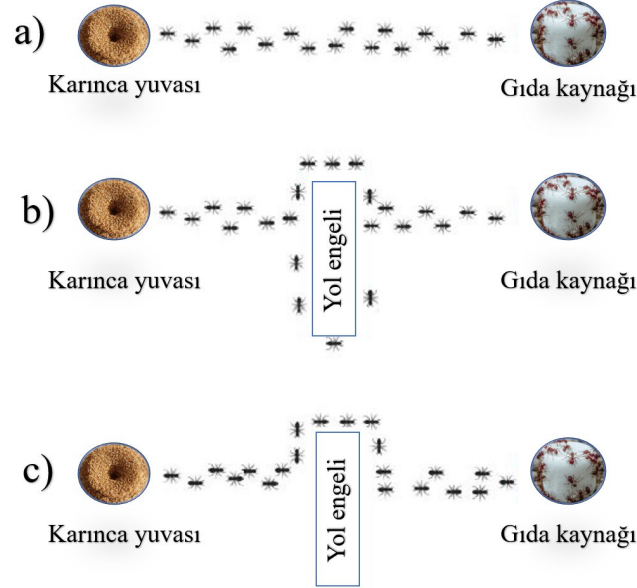
Ayrıca aynı amaç fonksiyonları, kısıtlamaları ve değişkenleri dikkate alınarak bu makaledeki redüktör tasarım problemlerin çeşitli optimizasyon teknikleri ile çözümünü sunan bazı çalışmalar incelenmiştir. Örneğin; aynı redüktör tasarımı daha önce Ray & Saini [27], Belegundu [28], Ray & Liew [29], Grković & Bulatović [30], Coelho & Mariani [31], Cagnina ve ark. [32], Saruhan & uyrur [40], Eskander ve ark. [41], Baykasoglu & Ozsoydan'ın [42], Guedria [43], Akhtar ve ark. [44], Mezura-Montes ve ark. [45], Aguirre ve ark. [46], Tomassetti [47], Akay & Karaboga [48], Gandomi ve ark. [49], Brajevic & Tuba [50], Yang & Gandomi [51], Mezura & Coello [52], Siddall [53], Golinski [54] ve Elaziz ve ark. [55] tarafından tanımlanmıştır. Ayrıca, sunulan tasarım problemleri, dişli dişlerinin bükülme gerilmesi, yüzey gerilmeleri ve millerin enine eğilmesinden kaynaklanan minimum ağırlığın bulunmasına olanak sağlamaktadır. Bunların arasında KKO optimizasyon çalışmaları yapanlar Ray & Liew [29], Grković & Bulatović [30], Coelho & Mariani [31] ve Cagnina ve ark. [32] olduğu görülmüştür.

## 3. KARINCA KOLONİSİNİN ALGORİTMASININ TEMELİ

Wilson ve Hölldobler [56], karıncaların feromon adı verilen sinyaller aracılığıyla iletişim kurabildiklerini keşfetmişlerdir. Feromonlar, karıncaların salgıladığı kimyasallardır ve salgılandıktan sonra izleri kısa bir süre kalmaktadır. Karıncalar, aynı yere ne kadar sık gittiklerini, o yol üzerine ne kadar çok feromon bıraktıklarını belirleyebilmektedirler. Feromon izlerinin tespit edilmesiyle, karıncaların nasıl yiyecek buldukları ve nasıl iş birliği yaptıkları açık hale gelmektedir. Karınca davranış kalıbı şu şekilde açıklanabilmektedir. Karıncalar feromon izini bulana kadar rastgele hareket etmektedir. Ardından, izdeki feromon konsantrasyonuna bağlı olarak karıncalar, izi takip edip etmeyeceklerine karar vermektedirler. Dolayısıyla, bir izi takip eden karınca sayısı ne kadar çoksa, diğer karıncaların izi takip etme olasılığı da o kadar yüksek olmaktadır [21].

Karıncalar gıda kaynaklarından yuvalarına en kısa yolu görme duyusunu kullanmadan bulma becerilerinin yanı sıra, adaptasyon yeteneklerine sahiptirler. Eğer takip ettikleri mevcut yol dış etkenler nedeniyle artık en kısa yol değilse, yeni en kısa yolu bulabilmektedirler. Başlangıçta, karıncalar düz bir çizgiyi takip etmekte ve aynı anda feromonları yolu bırakarak takip eden karıncalara yardımcı olmaktadır.

Şekil 2' de, doğal olarak salgıladıkları feromonların yardımıyla karıncaların hareketleri ve izledikleri yol gösterilmiştir. Karıncaların yolları engellendiğinde feromon izlerini takip edememekte ve genellikle takip edebilecekleri iki yoldan birini seçmektedirler. Karıncaların kısa yola geçiş yapma sayısı daha fazla olacağından, bırakılan feromon miktarı da daha fazla olmaktadır. Dolayısıyla, zamanla daha fazla karınca kısa yolu tercih eder hale gelmektedir. İzleri kontrol ederek, önceden rastgele hareket eden karıncalar, muhtemel olarak daha yoğun bir feromon izinin yönünü almayı tercih etmektedirler [57].



**Şekil 2.** Gerçek karınca davranışları; (a) Karıncalar yuva ve gıda kaynağı arasında bir yol izlemekteler, (b) Yolda bir engel belirlediğinde karıncalar, eşit olasılıkla sola veya sağa dönme seçeneğine sahiptirler, (c) Karıncalar kısa yola daha yüksek miktarda feromon bırakmakta ve çoğunluğunun kısa yolu seçmesine olanak sağlamaktadırlar.

Bu keşif karıncaların bir yiyecek kaynağı bulduklarında nasıl ulaştıklarını ve engelleri nasıl aştıklarını anlamaya yardımcı olmaktadır. Feromonlar, karıncalar arasında iletişim kurmanın bir aracı olarak işlev görmektedir. Bu davranışlar incelenerek bilim adamları bir algoritma geliştirilmiştir. Bu algoritma, karınca kolonilerinden ilham alınarak geliştirildiği için karınca kolonisi optimizasyon algoritması (KKO) olarak adlandırılmıştır. Klasik KKO temel olarak gerçek karıncalardan esinlenen yapay karıncalar kullanılarak formüle edilmiştir. Bu algorithmada da karıncalar feromon adı verilen izleri kenarlara bırakarak iletişim kurabilmektedirler. Bu algoritma ilk olarak Dorigo ve ark. [58] tarafından bir referans makalesinde incelenmiştir. Klasik KKO, sürü zekâsından ortaya çıkan davranışlar aracılığıyla optimizasyon problemlerini çözmeyi amaçlayan bir ajan popülasyonuna dayalı meta sezgisel bir yöntemdir. KKO algoritmasının tek yapay kontrol mimarisi bazı birimler için aynı ve ölçeklenebilir özelliklere sahiptir.

## 4. METODOLOJİ

Yapay koloni karıncaları tarafından bırakılan feromon izleri şeklinde biriken bilgilerden faydalanılarak, ardışık olarak daha kısa turlar oluşturulması mümkün olmaktadır. Bu sayede belirtilen iterasyonun sonunda belirlenen en iyi sonuçlar kaydedilmekte ve problem çözümüne ulaşılmaktadır.

Algoritmanın çalışma sürecinde, karıncalar tarafından güncellenen feromon izleri aracılığıyla iyi bir çözüm bulmak için bir veri seti oluşturulmuş ve bu veri seti her iterasyonda güncellenmiştir. Karıncaların sayısını artırmak çözümün iyileştirilmesine yardımcı olabilmektedir. Ancak işlem sayısını arttırdığı için, karıncaların sayısını çok fazla arttırmak işlem süresini uzatmaktadır. Problem boyutuna göre karmaşık bir hal alabilen KKO algoritmasının hızını artırmak için alfa ve beta gibi bazı parametrelerin güncellenmesi yoluyla başarılı bir şekilde test edilen çeşitli çalışmalar bulunmaktadır [59]. Programlama dilleri kullanılarak yapılan bilgisayar simülasyonları, KKO problemlerin iyi çözümler bulunduğunu göstermektedir [60].

Bu çalışmada, daha etkili ve güvenilir bir çözüme ulaşma hedefiyle bir KKO algoritması geliştirilmiştir. Bu hedef doğrultusunda, temel KKO yönteminin kodu değiştirilerek DEKKO adı verilen bir algoritma oluşturulmuştur. Bu DEKKO algoritmasının kodu, MATLAB yazılımı kullanılarak işlevselliği test edilmiş ve literatürdeki sonuçlarla karşılaştırılmıştır.

### 4.1. Karınca Koloni Algoritması

Bu algoritma, optimizasyon sürecinde yapılan feromon güncellemesine ek olarak, yerel bir feromon güncellemesi başlatmaktadır. Bu güncelleme aynı zamanda çevrimdışı feromon güncellemesi olarak da adlandırılır. Her

iterasyonun ardından tüm karıncalar yerel feromon güncellemesi yapar ve her bir karınca yalnızca kapsanan son kenara uygular. Yerel feromon güncellenmesi  $\tau_{ij}$ , Denklem 1' de gösterilmiştir

$$\tau_{ij} = \tau_{ij} \cdot (1 - Q) + \tau_0 \cdot Q \quad (1)$$

Burada, Q buharlaşma katsayısıdır ve miktarı  $0 \leq Q \leq 1$  olarak tanımlanır.  $\tau_0$ , feromonun başlangıç değeridir. Yerel feromon güncellemesinin ana amacı, her iterasyondan sonra karıncalar tarafından gerçekleştirilen aramanın çeşitliliğini sağlamaktır. Kapsanan kenarlardaki feromon konsantrasyonunu azaltarak, sonraki karıncaların başka bir kenarı seçmeleri teşvik edilmekte ve dolayısıyla farklı bir çözüm üretilmektedirler. İterasyon sırasında, birkaç karıncanın aynı çözümü üretme olasılığı düşer. Yerel feromon güncellemesi, maksimum ve minimum karınca sistemi ile aynı prensipte çalışmaktadır. Denklem 2' deki gibi her iterasyonun sonunda en iyi güncelleme göz önüne alınmaktadır.

$$\tau_{ij} = \begin{cases} \tau_{ij} \cdot (1 - q) + \Delta \tau_{ij} \cdot q, & \text{eğer } (i, j) \text{ en iyi tura aitse} \\ \tau_{ij}, & \text{aksi takdirde} \end{cases} \quad (2)$$

Karıncalar sistemleri ve karınca kolonisi sistemleri arasındaki önemli bir fark, karıncaların çözüm sürecinde kullandıkları karar kuralında görülmektedir. KKO' da aşağıdaki kural kullanılmaktadır.

Bir karıncanın i şehriden j şehrine taşınma olasılığı,  $0 \leq q \leq 1$  aralığında bir rasgele değişkene bağlıdır. Eğer bu değer yeni oran  $q_0$ 'a eşit veya küçükse,  $j = \text{argmax}_{C_{ij} \in N(S^P)} \{\tau_{ij} \eta_{ij}^\beta\}$ , işlemi uygulanmaktadır. Aksi halde, aşağıdaki Denklem 3 kullanılır.

$$P_{ij}^k = \begin{cases} \frac{\tau_{ij}^\alpha \eta_{ij}^\beta}{\sum_{C_{ij} \in N(S^P)} \tau_{ij}^\alpha \eta_{ij}^\beta}, & \text{eğer } C_{ij} \in N(S^P) \\ 0, & \text{aksi takdirde} \end{cases} \quad (3)$$

#### 4.2. Değiştirilmiş Karınca Koloni Optimizasyonu (DEKKO)

Önerdiğimiz algoritma, Karınca Kolonisi Optimizasyonu (KKO) algoritmasının bir modifikasyonudur. Bu değişiklik, algoritmanın uyum kabiliyetini artırmayı ve çözüm kalitesini iyileştirmeyi amaçlamaktadır. Bu yeni algoritmanın ana noktaları ve nasıl çalıştığı aşağıda gösterilmiştir.

**Uyarlanabilir  $\alpha$  ve  $\beta$  Değerleri:** Algoritmanın her iterasyonunda, feromonun ( $\alpha$ ) ve meta-sezgisel bilginin ( $\beta$ ) önemi dinamik olarak ayarlanmaktadır. Bu, çözüm arama sürecini daha esnek hale getirmekte ve yerel sıkışmayı önlemeye yardımcı olabilmektedir.

**Feromon Güncellemesi:** Her iterasyonda, tüm feromon izleri ( $\tau$ ) güncellenmektedir. Yeni feromon değeri, o iterasyondaki en iyi çözümün mevcut feromon değeri ve işlev değerine dayanarak artırılmaktadır. Böylelikle daha iyi çözümlere daha fazla ağırlık verilmektedir.

**Kraliçe Karınca (queen. Value):** En iyi çözüm "kraliçe karınca" olarak saklanmakta ve bu çözümün değeri diğer karıncaların gelecekteki arama yönlerini etkilemektedir.

**Durum Kontrolü ve Çıktı:** Algoritma, her iterasyonda en iyi çözümü kontrol etmekte ve eğer bu çözüm önceki en iyi çözümden daha iyiyse, yeni en iyi çözüm olarak kabul edilmektedir.

Bu önerilen modifikasyonları mevcut karınca optimizasyonuna entegre etmek için, belirtilen değişiklikleri yansıtan fonksiyonları güncellemek ve algoritmanın temel yapısını korumak gereklidir. Bu durum, hem feromon güncelleme mekanizmasını hem de  $\alpha$  ve  $\beta$ 'nin uyarlanabilir ayarlarını içerecektir. Ayrıca, ihtiyaç duyulduğunda en iyi çözümü ("kraliçe karınca") saklamak ve güncellemek gerekecektir.

Bu yaklaşım, redüktör tasarımı gibi çeşitli mühendislik tasarım problemlerinin karmaşık optimizasyon çalışmasında etkili olabilmektedir. Parametre tanımları Tablo 1'de verilmiştir.

Değiştirilmiş karınca kolonisi algoritmasında, yalnızca en iyi karınca feromon izinden elde edilen çözüm güncellenmekte ve belirlenen modifikasyonlarla DEKKO algoritması elde edilmektedir. İlk aşamada, m sayıda karınca parametre olarak belirlenmelidir. İlk çözümün kalitesi ve ikinci döngüde değişim sayısını temsil eden I iterasyon numarası bu parametreye dayanmaktadır.

**Tablo 1.** DEKKO parametre tanımlamaları

Tanımlama	
<b>max iterations</b>	Algoritmadaki maksimum iterasyon sayısı
<b>ant population</b>	Karıncalar popülasyon sayısı
<b>evaporation rate (<math>\rho</math>)</b>	Feromon buharlaşma oranı
<b>alpha (<math>\alpha</math>)</b>	Karıncanın karar verme sürecinde feromonun önemini kontrol eden parametre
<b>beta (<math>\beta</math>)</b>	Karıncanın karar verme sürecinde mesafenin önemini kontrol eden parametre
<b>queen. Value</b>	En iyi çözüm bulunduğunda "kraliçe karınca" olarak saklanmaktadır

İlk döngüde, tüm karıncalar rastgele bir konum seçmekte ve her bir değişken için bir değer almaktadır. Her bir karınca konumunu bulduğunda, karıncalar tarafından seçilen değişken değerleri için optimize edilmiş  $\min_{val}$  değerleri aracılığıyla fonksiyonun değeri belirlenir. Her bir karınca için fonksiyon değerleri elde edildiğinde, hangi karıncanın en iyi çözümü seçtiği değerlendirilmekte ve bu çözümün değeri  $\min_{function}$  olarak kaydedilmektedir. İkinci döngüde, tüm karıncalar ilk aşama  $\min_{function}$ 'dan en iyi çözüm yerelini elde etmektedir. En iyi değer için feromon güncellemesi Denklem 4' te gösterildiği şekilde yapılır.

$$\tau(i) = 0.1 (1 - \eta_{ij}). \tau(i) \quad (4)$$

Her iterasyonun çözümünü düzeltmek için  $\alpha_j$  ve  $\beta_j$  değerleri verilmektedir.. İlk iterasyonda, bu değer bir kodda verilir ve bu değer, değişkenlerin olası çözüm aralıklarının alt ve üst sınırlarına bağlı olarak seçilmektedir. Ayrıca, her takip eden iterasyon için  $\alpha$  ve  $\beta$ 'nin düzeltme değerleri Denklem 5' te olduğu gibi yapılmaktadır.

$$\alpha_j = 0,05 . \alpha_i ; \quad \beta_j = 0,05 . \beta_i \quad (5)$$

İkinci adımda açıklanan bu prosedür, ikinci döngüde bir iterasyon tekrarıdır ve maksimum iterasyon sayısına kadar çözümleri düzeltmek için tekrarlamak üzere programlanmıştır. İsteğe bağlı olarak, en iyi değer kodda tanımlanabilmektedir. Bir değer tanımlanmışsa ve belirli bir iterasyon sayısından sonra gereksinimden daha iyi bir çözüm elde edilemezse, tüm döngü tekrarlanabilmektedir. En iyi çözüm değerleri elde edildikten sonra, bu değerler kaydedilmekte ve işlev değerleri tanımlanmaktadır. DEKKO algoritmasının basit bir optimizasyon sözde kodu Şekil 3' te verilmiştir.

DEKKO Algoritması
<pre> for t = 1 : maxiteration MACO= createMaco ( graph, colony , antnumber, tau, eta, alpha, beta); function [ graph ] = createGraph() Update alpha and beta; alpha_j = 0,05 . alpha_i ; beta_j = 0,05 . beta_i  Determine optimum value and best function value for i = 1 : antnumber tau(j) = tau(i) + 0,005 . minfunction  if minfunction &lt; bestfunction; function = new function; end  queen.value= bestvalue; to update tau ; tau = 0.1*( 1 - rho ) .* tau;  output = [ 'MACO (Modified ACO) #', 'Iteration #', value1 , ' optimum Function value = ', value2]; disp(output) </pre>

Şekil 3. DEKKO sözde kodu.

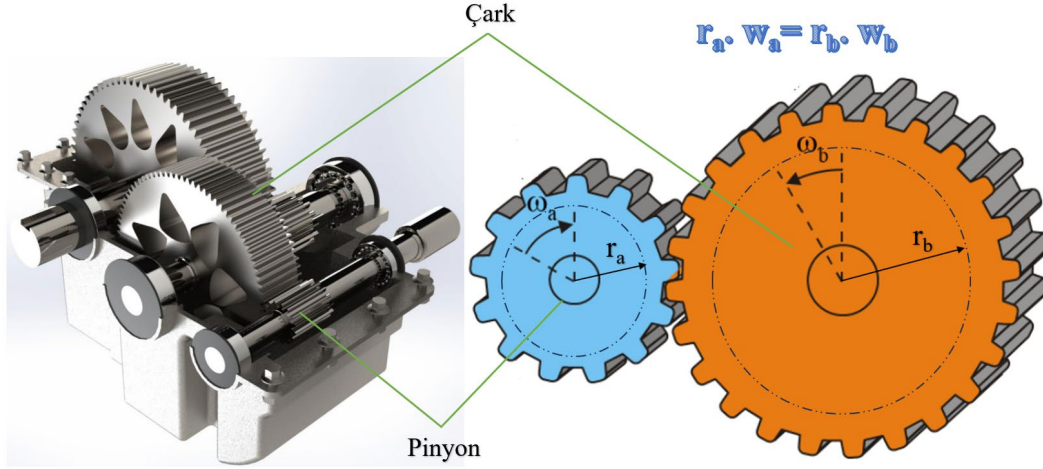
## 5. BUGULAR VE TARTIŞMA

Literatürde bulunan redüktör tasarımı uygulamalarında uygulanan KKO çözümleri incelenmiştir. Buradaki tasarımlar birebir olarak DEKKO algoritması kullanılarak yürütülmüş ve sonuçlar karşılaştırma amacıyla okuyucuya sunulmuştur.

### 5.1. Redüktör Tasarımı

Bir dönme hareketini dişliler yardımı ile değiştiren ve aktaran makine elemanlarına redüktör denmektedir. İngilizcedeki “Reduce” kelimesinden gelmektedir. Temel olarak Şekil 4' te gösterilen redüktörler ana gövde, dişli, rulmanlar, giriş milleri ve çıkış millerinden oluşmaktadır. Redüktör bir motor ile bir makine parçası arasına yerleştirilmekte ve motorun sağladığı hızı azaltarak sisteme aktarılmasını sağlamaktadır. Bir redüktör dönen ve döndüren olmak üzere en az iki dişliden meydana gelmektedir. Bu iki dişliden küçük olanına pinyon, büyük olanına çark adı verilmektedir. Bunlar yardımıyla hız düşerken diğer taraftan sistemin torqu artmaktadır. Böylelikle prosesin daha iyi, daha hassas şekilde kontrol edilmesi ve daha iyi sonuçlar elde edilmesine olanak sağlanmaktadır [61]. Redüktör bloklarına göre Monoblok redüktör ve yatık tip redüktör olmak üzere temelde ikiye ayrılabilir. Monoblok redüktör bloklarında yataklar, bir kapak veya gövde parçasına bağlı değil ve doğrudan gövde üzerinde konumlanmaktadır. Yatık redüktör blokları, ağır iş koşullarında kullanılmak üzere özel olarak tasarlanmış redüktörlerdir. Bu tip redüktörlerde, redüktör ile tahrik mekanizması arasında farklı bağlantı şekilleri bulunabilmektedir. Gövdeleri ve kapakları yüksek kaliteli çelik dökümden üretilirken, dişliler sementasyon çeliklerinden imal edilmekte ve yüzeyleri modern üretim yöntemleriyle sertleştirilerek

taşlanmaktadır. Her iki dişli çarkın yuvarlanma daireleri üzerindeki çevre hızlarının eşit olduğu bilinmektedir. Buradan hareketle pinyon yuvarlanma dairesi çapı ve açısal hızlarının çarpımının çark yuvarlanma dairesi çapı ve açısal hızlarının çarpımına eşit olduğu sonucu çıkmaktadır. Geleneksel bir redüktör çalışma prensibini Şekil 4' te gösterilmiştir [62]. Burada; pinyon yuvarlanma dairesi çapı  $r_a$ , pinyon açısal hızı  $w_a$ , çark yuvarlanma dairesi çapı  $r_b$  ve çark açısal hızı  $w_b$  olarak gösterilmiştir.

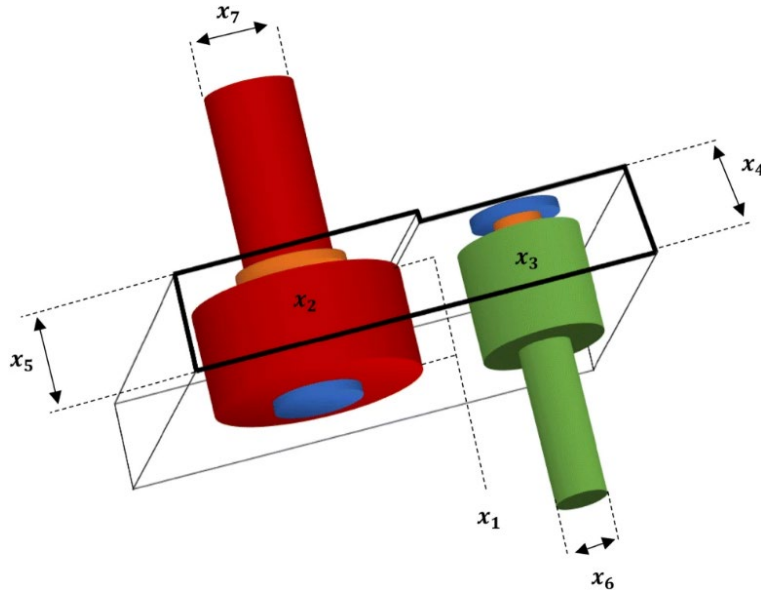


Şekil 4. Geleneksel bir redüktörün çalışma prensibi [62].

Redüktörlerin tercih edilme sebepleri arasında güç aktarımının yanı sıra farklı dönme yönleri elde edilmesi, küçük bir hacimden büyük bir çevrim oranı sağlanması, az bakım maliyetine sahip olması ve bakımları doğru yapıldığında kullanım ömürleri olması gösterilebilmektedir.

## 5.2. Probleminin Tanımlanması

Redüktörlerde dişler üzerindeki eğilme gerilimi, yüzey gerilmeleri, enine sapmalar ve mildeki gerilmelerden dolayı oluşan birtakım kısıtlamalar bulunmaktadır. Bunlar dikkate alınarak DEKKO algoritması ile minimum ağırlıkta bir redüktör tasarımına ulaşılması amaçlanmıştır. Yanak genişliği  $b$  ( $x_1$ ), diş modülü  $m$  ( $x_2$ ), pinyonun diş sayısı  $z$  ( $x_3$ ), ilk mil ile rulmanlar arasındaki uzunluk  $l_1$  ( $x_4$ ), ikinci mil ile rulmanlar arasındaki uzunluk  $l_2$  ( $x_5$ ), birinci mil çapı  $d_1$  ( $x_6$ ) ve ikinci mil çapı  $d_2$  ( $x_7$ ) olarak tasarlanmıştır [27-32, 40-55]. Redüktör tasarımı Şekil 5' te gösterilmiştir [55].



Şekil 5. Problem çözümünde kullanılan redüktör yapısı [55].

Literatürde yukarıda tasarım ve parametreleri verilen redüktörün minimum ağırlık problemlerinin çözümü için amaç fonksiyonu, kısıtlama fonksiyonları ve tasarım değişkenleri tanımlanmıştır. Bu redüktör problemi, dişli dişlerinin bükülme gerilimi, yüzey gerilimi, millerin enine sapmaları ve millerdeki gerilmeler üzerindeki kısıtlamalar ile ifade edilmektedir. Bunlar ayrı ayrı aşağıdaki denklemlerde ifade edilmiştir [27-32, 40-55]. Redüktörün Amaç fonksiyonu Denklem 6' de gösterilmiştir.

$$f(x) = (0,7854 x_1 x_2^2 (3,3333 x_3^2 + 14,933 x_3 - 43,0934) - 1,508 x_1 (x_6^2 + x_7^2) + 7,4777 x_1 (x_6^3 + x_7^3) + 0,7854 (x_4 x_6^2 + x_5 x_7^2)) \quad (6)$$

Kısıtlama fonksiyonları ve tasarım değişkenleri Denklem 7-24 arasında gösterilmiştir.

$$g_1(x) = 27 / (x_1 x_2^2 x_3) - 1 \leq 0 \quad (7)$$

$$g_2(x) = (397,5 / (x_1 x_2^2 x_3^2)) - 1 \leq 0 \quad (8)$$

$$g_3(x) = (1,93 x_4^3 / (x_2 x_6^4 x_3)) - 1 \leq 0 \quad (9)$$

$$g_4(x) = (1,93 x_5^3 / (x_2 x_7^4 x_3)) - 1 \leq 0 \quad (10)$$

$$g_5(x) = \left( \frac{1}{110x_6^3} \right) \sqrt{(745,0 x_4 / x_2 x_3)^2 + 16,9 10^6} - 1 \leq 0 \quad (11)$$

$$g_6(x) = \left( \frac{1}{85x_7^3} \right) \sqrt{(745,0 x_5 / x_2 x_3)^2 + 157,5 10^6} - 1 \leq 0 \quad (12)$$

$$g_7(x) = (x_2 x_3 / 40) - 1 \leq 0 \quad (13)$$

$$g_8(x) = (5x_2 / x_1) - 1 \leq 0 \quad (14)$$

$$g_9(x) = (x_1 / 12x_2) - 1 \leq 0 \quad (15)$$

$$g_{10}(x) = (1,5x_6 + 1,9) / x_4 - 1 \leq 0 \quad (16)$$

$$g_{11}(x) = (1,1x_7 + 1,9) / x_5 - 1 \leq 0 \quad (17)$$

$$2,6 \leq x_1 \leq 3,6 \quad (18)$$

$$0,7 \leq x_2 \leq 0,8 \quad (19)$$

$$17,0 \leq x_3 \leq 28,0 \quad (20)$$

$$7,3 \leq x_4 \leq 8,3 \quad (21)$$

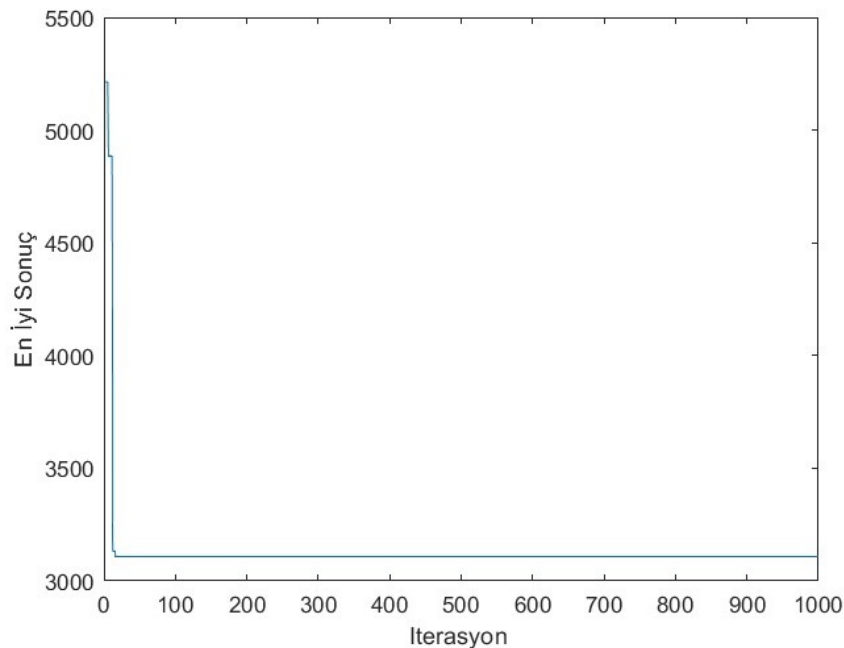
$$7,8 \leq x_5 \leq 8,3 \quad (22)$$

$$2,9 \leq x_6 \leq 3,9 \quad (23)$$

$$5,0 \leq x_7 \leq 5,5 \quad (24)$$

### 5.3. Redüktör Tasarımı için DEKKO Algoritma Simülasyonu

Algoritma hedef ve kısıt fonksiyonları göz önünde bulundurularak en iyi sonuca ulaşmak için 20 kez çalıştırılmıştır. En iyi sonuç, 14 iterasyon, 100 karınca ve 66,81 saniye süreyle elde edilmiştir. İterasyon-en iyi sonuç grafiği Şekil 6 ve MATLAB sonuç ekranı Şekil 7' de gösterilmiştir. Ayrıca, DEKKO için elde edilen en iyi sonuç, literatürde aynı problem için elde edilen sonuçlarla karşılaştırılmıştır. Sonuç özetleri Tablo 2' de verilmiştir.



Şekil 6. İterasyon-en iyi sonuç grafiği.



```

1 function engineeringproblem_modified
2 % Parametreler
3 numAnts = 100;
4 numVariables = 7;
5 maxIter = 1000;
6 evaporationRate = 0.2;
7 alpha = 0.5; % Başlangıç Alpha değeri
8 beta = 0.5; % Başlangıç Beta değeri
9 maxPheromoneIndex = 500;
10
11 % Değişkenler için sınırlar
12 lb = [2.6 0.7 17 7.3 7.8 2.9 5.0];
13 ub = [3.6 0.8 28 8.3 8.3 3.9 5.5];
14
15
16 % Optimizasyon
17 [bestSol, bestCost, bestIter, elapsedTime] = modifiedACO(numAnts, numVariab.
18
19 % Sonuçları Göster
20 fprintf('En İyi Çözüm: %s\n', mat2str(bestSol));
21 fprintf('En İyi Sonuç: %.4f\n', bestCost);
22 fprintf('Bulunan İterasyon: %d\n', bestIter);
23 fprintf('Geçen Süre: %.2f saniye\n', elapsedTime);
24 fprintf('Karıncaya Sayısı: %d\n', numAnts);
25

```

```

Command Window
Karıncaya Sayısı: 100
>> engineeringproblem_modified
En İyi Çözüm: [3.53386773547094 0.705410821643287 17 8.27394789579158 8.05350701402806 3.
En İyi Sonuç: 3105.8779
Bulunan İterasyon: 14
Geçen Süre: 66.81 saniye
Karıncaya Sayısı: 100
fx >>

```

Şekil 7. Redüktör tasarımı için MATLAB sonuç ekranı.

Tablo 2. Redüktör problem çözümlerinin karşılaştırılması

Definition	Eskandar ve ark. (WCA)	Baykasoglu & Ozsoydan (FA)	Guedria (IAPSO)	Grković & Bulatović (MACA)	Önerilen Metot (DEKKO)
$x_1$	3,500000000	3,507495	3,501313	3,499999999470	3,533867735
$x_2$	0,700000000	0,7001	0,700000	0,699999999999	0,705410821
$x_3$	17,000000000	17,00000	18,00000	17,000000000000	17,0000000
$x_4$	7,300000000	7,719674	8,127814	8,051780184365	8,27394789
$x_5$	7,715319000	8,080854	8,042121	8,084677071037	8,053507014
$x_6$	3,350214000	3,351512	3,352446	3,351689274675	3,39268997
$x_7$	5,286654000	5,287051	5,287076	5,286782111706	5,35698359
Best f(x)	2994,4710660	3010,137492	3187,630486	3009,669238634	3105,8779
İterasyon	-	-	-	-	-
Karıncaya Sayısı	-	-	-	10000	100

Ayrıca, DEKKO için elde edilen en iyi sonuç literatürde aynı problem için elde edilen sonuçlarla karşılaştırılmıştır. Eskandar ve ark. su döngüsü algoritması (WCA), Baykasoglu & Ozsoydan'ın ateşböceği algoritması (FA), Guedria geliştirilmiş bir hızlandırılmış parçacık sürüsü optimizasyon algoritması (IAPSO) ve Grković & Bulatović ise değiştirilmiş bir karınca kolonisi algoritması (MACA) kullanarak çalışmalarını gerçekleştirmişlerdir. DEKKO algoritması uygulandığında, en iyi çözümün 3105,8779 olduğu bulunmuştur.

DEKKO algoritması sonucu, Eskander ve ark.'nın [41] çalışmasına göre (WCA) %3,58 daha fazla bulunmuştur. DEKKO 14 iterasyonla çözüme ulaşırken, WCA çalışmasında iterasyon bilgisi verilmemiştir.

DEKKO algoritması sonucu, Baykasoglu & Ozsoydan'ın [42] çalışmasına göre (FA) % 3,082 daha fazla bulunmuştur. DEKKO 14 iterasyonla çözüme ulaşırken, FA' da en iyi sonuçlar ortalama 3000 iterasyon-değerlendirme ile çözüme ulaşmıştır.

DEKKO algoritması sonucu, Guedria'nın [43] çalışmasına göre (IAPSO) % 2,63 daha düşük bulunmuştur. DEKKO 14 iterasyonla çözüme ulaşırken, IAPSO 6000 iterasyon-değerlendirme ile çözüme ulaşmıştır.

DEKKO algoritması sonucu, Grković & Bulatović'in [30] çalışmasına göre (MACA) %3,09 daha fazla bulunmuştur. DEKKO 100 karınca ile çözüme ulaşırken, MACA 10000 karınca ile çözüme ulaşmıştır.

Bundan dolayı, DEKKO' nun literatürdeki sonuçlar arasında başarı elde ettiği açıktır. Dolayısıyla, DEKKO' nun redüktör tasarımlarında güvenilir bir şekilde kullanılabilirliği açıktır.

## 6. SONUÇ

Son yıllarda çözülen problemin özelliklerine ve karmaşıklığına bağlı olarak meta-sezgisel algoritmaların nispeten daha iyi yönleri bir araya getirilerek karma ve iyileştirilmiş algoritmalar oluşturulmuştur. Bu algoritmalar redüktör

tasarımı gibi mühendislik problemlerini çözümünde sıklıkla kullanılmıştır. Havacılık ve imalat sanayindeki aktarma sistemlerini içeren farklı türdeki mekanik ekipmanların çalışma verimliliği ve hassasiyeti dikkate alındığında redüktör tasarımı dikkat çekmektedir.

Bu makale amaç fonksiyonları, kısıt fonksiyonları ve tasarım değişkenleri belirli olan redüktör probleminin değiştirilmiş bir karınca kolonisi optimizasyon algoritması (DEKKO) ile çözümüne odaklanmıştır. Literatürde farklı teknikler kullanılarak redüktör probleminin çözümlerinden daha iyi sonuçlar elde etmek amacıyla karınca koloni (KKO) algoritmasının değiştirilme yolun gidilmiştir. Bu yeni algoritma KKO' nun avantajlı yönlerini birleştirerek geliştirilmiş ve DEKKO adı verilmiştir. DEKKO kodu MATLAB yazılımı kullanılarak çalıştırılmıştır.

En iyi sonucu elde etmek için program 20 kez çalıştırılmıştır. En iyi sonuca 14 iterasyon, 100 karınca ve 66,81 saniyede ulaşılmıştır. Burada en iyi çözüm 3105,8779 vektör birim olduğu bulunmuştur. Ayrıca, DEKKO için elde edilen en iyi sonuç literatürde aynı problem için elde edilen sonuçlarla karşılaştırılmıştır. Eskandar ve ark. su döngüsü algoritması (WCA), Baykasoglu & Ozsoydan atesböceği algoritması (FA), Guedria geliştirilmiş bir hızlandırılmış parçacık sürüsü optimizasyon algoritması (IAPSO) ve Grković & Bulatović ise değiştirilmiş bir karınca kolonisi algoritması (MACA) kullanarak çalışmalarını gerçekleştirmişlerdir. DEKKO algoritması sonucu, WCA' ya göre % 3,58 daha fazla bulunmuştur. DEKKO 14 iterasyonla çözüme ulaşırken, WCA çalışmasında iterasyon bilgisi verilmemiştir. DEKKO algoritması sonucu, FA' ya göre % 3,082 daha fazla bulunmuştur. DEKKO 14 iterasyonla çözüme ulaşırken, FA en iyi sonuçları ortalama 3000 iterasyon-değerlendirme ile çözüme ulaşmıştır. DEKKO algoritması sonucu, IAPSO' ya göre %2,63 daha düşük bulunmuştur. DEKKO 14 iterasyonla çözüme ulaşırken, IAPSO 6000 iterasyon-değerlendirme ile çözüme ulaşmıştır. DEKKO algoritması sonucu MACA' ya göre %3,09 daha fazla bulunmuştur. DEKKO 100 karınca ile çözüme ulaşırken, MACA 10000 karınca ile çözüme ulaşmıştır.

Bu durum, DEKKO' nun literatürdeki sonuçlar arasında başarı elde ettiğini açıkça göstermektedir. Dolayısıyla, DEKKO' nun redüktör tasarım çözümlerinde güvenilir bir şekilde kullanılabilmesi açıktır. Kullanıcılar, DEKKO algoritmasını kullanarak redüktör tasarımı ve ön üretim hakkında kolayca bilgi edinebileceklerdir. Böylece endüstri, profesyoneller ve bu alanda çalışan kullanıcılar için maliyet ve zaman tasarrufuna katkıda bulunulması amaçlanmıştır. Gelecekte DEKKO algoritmasının daha da optimum sonuçlar vereceği inancıyla, algoritmanın daha fazla araştırılması ve geliştirilmesi gerektiği düşünülmektedir. Ayrıca, geliştirilen DEKKO algoritması diğer mühendislik çalışmalarında test edilebilir ve sonuçları literatürdeki çalışmalarla karşılaştırılarak okuyuculara sunulabilir.

## Yazar Katkıları

Kürşat Tanrıver- Kavramsal tasarım, Metodoloji, Veri düzenleme, Yazı- orijinal taslak.

Mustafa Ay- Görselleştirme, Metodoloji, Araştırma, Doğrulama, Denetim, Yazı- inceleme ve düzenleme.

## Çıkar Çatışması

Makale yazarları aralarında herhangi bir çıkar çatışması olmadığını beyan ederler.

## KAYNAKÇA

- [1] Ö. Akçay, "Structural Optimization of the Brake Pedal using Artificial Intelligence," International Journal of Automotive Science and Technology, vol. 7, no. 3, pp. 187–195, Sep. 2023.
- [2] K. Tanrıver, M. Ay, "Experimental, software and topological optimization study of unpredictable forces in bolted connections," Tehnicki Vjesnik-technical Gazette, vol. 30, no. 4, Aug. 2023.
- [3] W. Zhao, Y. Liu, Y. Li, C. Hu, and S. Rui, "Multi-robot coverage path planning for dimensional inspection of large free-form surfaces based on hierarchical optimization," The International Journal of Advanced Manufacturing Technology, vol. 127, no. 11–12, pp. 5471–5486, Jul. 2023.
- [4] C. Baştemur Kaya, E. Kaya, "Yüksek Boyutlu Nümerik Optimizasyon Problemlerinin Çözümünde Kelebek Optimizasyon Algoritmasının Performansının Değerlendirilmesi," Müh.Bil.ve Araş.Dergisi, 4:296–30, 2022.
- [5] A.M. Ebid, M.Y. Abdel-Kader, I.M. Mahdi, I.Abdel-Rasheed, "Ant Colony Optimization based algorithm to determine the optimum route for overhead power transmission lines," Ain Shams Engineering Journal, vol. 15, no 1,2024,102344.
- [6] V.K. Harikrishnan, A.I. Sivakumar, S. Sampath, and S. Paramasivam, "A Time-Performance Improvement Model with Optimal Ergonomic Risk Level Using Genetic Algorithm," Transactions of FAMENA, vol. 47, no. 4, pp. 109–128, Jan. 2023.
- [7] İ. Avcı and M.N. Yıldırım, "Solving Weapon-Target Assignment Problem with Salp Swarm Algorithm," Tehnicki Vjesnik-technical Gazette, vol. 30, no. 1, Feb. 2023.
- [8] S. Katiyar and A. Dutta, "Comparative analysis on path planning of ATR using RRT\*, PSO, and modified APF in CG-Space," Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, vol. 236, no. 10, pp. 5663–5677, Jan. 2022.
- [9] E.S. Eşsiz, V.N. Kiliç, and M. Oturakçi, "Firefly-Based feature selection algorithm method for air pollution analysis for Zonguldak region in

- Turkey,” Turkish Journal of Engineering, vol. 7, no. 1, pp. 17–24, Jan. 2023.
- [10] B. Irmak and Ş. Gülcü, “Training of the feed-forward artificial neural networks using butterfly optimization algorithm,” MANAS Journal of Engineering, vol. 9, no. 2, pp. 160–168, Dec. 2021.
- [11] M.C. Cihan, M.B. Çetinkaya, ve H. Duran “Performance comparison of artificial bee colony algorithm based approaches for retinal vessel segmentation”, BAUN Fen. Bil. Enst. Dergisi, vol. 23, no. 2, pp. 792–807, 2021.
- [12] J.D. Farmer, N.H. Packard, and A. S. Perelson, “The immune system, adaptation, and machine learning,” Physica D: Nonlinear Phenomena, vol. 22, no. 1-3, pp. 187-204, 1986.
- [13] E.K. Yaylacı, A.E. Yılmaz, H.N. Özdeş, “Kızıl Tilki Optimizasyon Algoritması ile Da-Da Alçaltıcı Tip Dönüştürücü Kontrolör Katsayılarının Optimizasyonu,” Müh.Bil.ve Araş.Dergisi, vol. 4, no. 2, pp. 129–140, 2022.
- [14] A. Tuncer, “Otonom mobil robotların Voronoi diyagramı ve karınca kolonisi optimizasyonuna dayalı yol planlaması,” Journal of Innovative Engineering and Natural Science, Dec. 2023.
- [15] D. Zhang, R. Luo, Y.-B. Yin, and S. Zou, “Multi-objective path planning for mobile robot in nuclear accident environment based on improved ant colony optimization with modified A\*,” Nuclear Engineering and Technology, vol. 55, no. 5, pp. 1838–1854, May 2023.
- [16] L. Sun, Y.S. Chen, W. Ding, and J. Xu, “LEFSA: label enhancement-based feature selection with adaptive neighborhood via ant colony optimization for multilabel learning,” International Journal of Machine Learning and Cybernetics, Aug. 2023.
- [17] A. Durmuş, Z. Yıldırım, “Synthesis of Linear Antenna Arrays with Physics Based AOA, CryStAl and LA Algorithms.” Müh.Bil.ve Araş.Dergisi, vol. 4, no. 2, pp. 164–172, 2022.
- [18] S. Wu, R.Guo, X. Li, "Quasi-Static Force Analysis and Tooth Profile Modification Optimization of the Cycloid Speed Reducer," Applied Sciences, vol. 14, no. 2, p. 845, 2024.
- [19] A. Lanzotti, M. Calise, M. Molaro, S. Patalano, F.Renno, ark. ‘Federica’s MOOC’ (Massive Open Online Course): a blended course in engineering drawing at Federico II,” Int J Interact Des Manuf vol. 13, pp. 1115–1128, 2019.
- [20] X. Zhou, W. Gui, A.A. Heidari, Z. Cai, G. Liang, and H. Chen, “Random following ant colony optimization: Continuous and binary variants for global optimization and feature selection,” Applied Soft Computing, vol. 144, p. 110513, Sep. 2023.
- [21] Ö. HiZaroğlu, “Diz Eklemi Simülator Prototipinin ISO 14243/3 Standardına Uygun Sistem Modelleme Çalışmasının Yapılması ve Matlab Simulink Ortamında Yürüyüş Profilinin Simule Edilmesi,” Jan. 01, 2024.
- [22] A. Hashemi, M. Joodaki, N.Z. Joodaki, and M. B. Dowlatshahi, “Ant colony optimization equipped with an ensemble of heuristics through multi-criteria decision making: A case study in ensemble feature selection,” Applied Soft Computing, vol. 124, p. 109046, Jul. 2022.
- [23] V. Maniezzo, L.M. Gambardella, and F. De Luigi, “Ant Colony Optimization,” in Studies in fuzziness and soft computing, pp. 101–121, 2004.
- [24] X. Zheng, Z. Wang, D. Liu, and H. Wang, “A path planning algorithm for PCB surface quality automatic inspection,” Journal of Intelligent Manufacturing, vol. 33, no. 6, pp. 1829–1841, Apr. 2021.
- [25] M. Das, A. Roy, S. Maity, and S. Kar, “A Quantum-inspired Ant Colony Optimization for solving a sustainable four-dimensional traveling salesman problem under type-2 fuzzy variable,” Advanced Engineering Informatics, vol. 55, p. 101816, Jan. 2023.
- [26] J. Heins, J. Bossek, J. Pohl, M. Seiler, H. Trautmann, P. Kerschke, “A study on the effects of normalized TSP features for automated algorithm selection,” Theoretical Computer Science, 940, Part B, pp. 123-145, 2022.
- [27] T. Ray and P. Saini, Engineering design optimization using swarm with an intelligent information sharing among individuals,” Engineering Optimization, vol. 33, no. 6, pp. 735–748, Aug. 2001.
- [28] A.D. Belegundu, “A Study of Mathematical Programming Methods for Structural Optimization,” Dept. of Civil and Environmental Engineering, University of Iowa, Iowa, IA, 1982.
- [29] T. Ray and K.M. Liew, “Society and civilization: an optimization algorithm based on the simulation of social behavior,” IEEE Transactions on Evolutionary Computation, vol. 7, no. 4, pp. 386–396, Aug. 2003,
- [30] V. Grković and R. Bulatović, “Modified Ant Colony Algorithm for Solving Engineering Optimization Problems,” IMK-14 – Research & Development, vol. 18, no. 4, EN115-122 UDK 621, 2012.
- [31] L.D.S. Coelho and V.C. Mariani, “Use of chaotic sequences in a biologically inspired algorithm for engineering design optimization,” Expert Systems With Applications, vol. 34, no. 3, pp. 1905–1913, Apr. 2008.
- [32] L. Cagnina, S.C. Esquivel, and C. a. C. Coello, “Solving Engineering Optimization Problems with the Simple Constrained Particle Swarm Optimizer,” Informatica, vol. 32, pp. 319–326, 2008.
- [33] S.S. Rao, “Engineering Optimization: Theory and Practice,” John Wiley & Sons, 1996.
- [34] Q. Li, P. Xu, L. Li, W. Xu, “Tan, D. Investigation on the Lubrication Heat Transfer Mechanism of the Multilevel Gearbox by the Lattice Boltzmann Method,” Processes, vol. 12, p. 381, 2024.
- [35] L. Maccioni, F. Concli, M. Blagojevic, “A new three-stage gearbox concept for high reduction ratios: Use of a nested-cycloidal architecture to increase the power density,” Mechanism and Machine Theory, vol. 181, p. 105203, 2023.
- [36] M.Habib Farhat, T. Hentati, X. Chimentin, F.Bolae rs, F. Chaari, M. Haddar, “Numerical model of a single stage gearbox under variable

- regime,” *Mechanics Based Design of Structures and Machines*, vol. 51, no. 2, pp. 1054-1081, 2023.
- [37] A.B. Tatar, “Planet Redüktörlü Robotik Aktüatör Tasarımı ve Üç Boyutlu (3B) Yazıcı ile İmalatı,” *IJ3DPTDI*, vol. 7, no. 2, pp. 161-168, 2023.
- [38] M. Demir, F. Güner, “Generating a Matlab Code with Parameter Optimization in Gearbox,” *Karadeniz Fen Bilimleri Dergisi*, vol. 12, no. 2, pp. 1098-1107, 2022.
- [39] S. Koçak, Y. Kaplan, A.T. Güner, “Sonsuz Vida Mekanizması ile Yeni Tasarlanan Bilyeli Sonsuz Vida Mekanizmasının Verimlerinin Deneysel Olarak Karşılaştırılması,” *Gazi University Journal of Science Part C: Design and Technology*, vol. 8, no. 1, pp. 160-168, 2020.
- [40] H. Saruhan, İ. Uygur, “Design optimization of mechanical system using genetic algorithms,” *SAUJS*, vol. 7, no. 2, pp. 77-84, 2003
- [41] H. Eskandar, A. Sadollah, A. Bahreininejad, and M. Hamdi, “Water cycle algorithm – A novel metaheuristic optimization method for solving constrained engineering optimization problems,” *Computers & Structures*, vol. 110-111, pp. 151-166, Nov. 2012.
- [42] A. Baykasoglu, F.B. Ozsoydan, “Adaptive firefly algorithm with chaos for mechanical design optimization problems,” *Appl soft Comput*, vol. 36, pp. 152-164, 2015.
- [43] N.B. Guedria, “Improved accelerated pso algorithm for mechanical engineering optimization problems,” *Appl Soft Comput*, vol. 40, pp. 455-467, 2016.
- [44] S. Akhtar, K. Tai, T. Ray, “A socio-behavioural simulation model of engineering design optimization,” *Eng. Optimiz.*, vol. 34, pp. 341-354, 2002.
- [45] E. Mezura-Montes, C.A.C. Coello, R. Landa-Becerra, “Engineering optimization using a simple evolutionary algorithm,” in: *Proceedings of the 15th IEEE International Conference on Tools with Artificial Intelligence*, 2003.
- [46] H. Aguirre, A.M. Zavala, E.V. Diharce, S.B. Rionda, “COPSO: constrained optimization via PSO algorithm,” Technical report No. I-07-04/22-02-2007, Center for Research in Mathematics (CIMAT), 2007.
- [47] G. Tomassetti, “A cost-effective algorithm for the solution of engineering problems with particle swarm optimization,” *Eng. Optimiz.*, vol. 42, pp. 471-495, 2010.
- [48] B. Akay, D. Karaboga, “Artificial bee colony algorithm for large-scale problems and engineering design optimization,” *J. Intell. Manuf.*, vol. 23, no. 4, pp. 1001-1014, 2012.
- [49] A.H. Gandomi, X.-S. Yang, A.H. Alavi, S. Talatahari, “Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems,” *Eng. Appl. Artif. Intell.*, vol. 29, pp. 17-35, 2013.
- [50] I. Brajevic, M. Tuba, An upgraded artificial bee colony (ABC) algorithm for con-strained optimization problems, *J. Intell. Manuf.*, vol. 24, pp. 729-740, 2013.
- [51] X.-S. Yang, A.H. Gandomi, “Bat algorithm: a novel approach for global engineering optimization,” *Eng. Comput.*, vol. 29, no. 5, pp. 464-483, 2012.
- [52] E. Mezura-Montes and C. a. C. Coello, “Useful Infeasible Solutions in Engineering Optimization with Evolutionary Algorithms,” in *Lecture Notes in Computer Science*, pp. 652-662, 2005.
- [53] J.N. Siddall, “Analytical Decision-Making in Engineering Design,” Prentice-Hall, Englewood Cliffs, 1972.
- [54] J. Golinski, “An adaptive optimization system applied to machine synthesis,” *Mech. Mach. Synth.*, vol. 8, pp. 419-436, 1973.
- [55] M.A. Elaziz,, L. Abualigah,, A.A. Ewees, et al., “Triangular mutation-based manta-ray foraging optimization and orthogonal learning for global optimization and engineering problems”, *Appl Intell*, vol. 53, pp. 7788-7817, 2023.
- [56] E.O. Wilson, B. Hölldobler, “Dense hierarchies’ and mass communication as the basis of organization in ant colonies,” *Trends in Ecology and Evolution*, vol. 3, no. 3, pp. 65-8, 1988
- [57] B. Benhala ve ark. , “Sizing of current conveyors by means of an Ant Colony Optimization technique,” *International Conference on Multimedia Computing and Systems*, Ouarzazate, Morocco, 2011, pp. 1-6, 2011.
- [58] M. Dorigo, V. Maniezzo, A. Colomi, “The Ant System: Optimization by a colony of cooperating agents”, *IEEE Transactions on Systems, Man, and Cybernetics-Part B*, vol. 26, no. 1, pp. 1-13, 1996.
- [59] Q. Shen, J. Jiang, J. Tao, G. Shen, and R. Yu, “Modified Ant colony Optimization Algorithm for variable selection in QSAR modeling: QSAR studies of cyclooxygenase inhibitors,” *Journal of Chemical Information and Modeling*, vol. 45, no. 4, pp. 1024-1029, Apr. 2005.
- [60] S. Gorbatyuk, A. Khan, M. Doddamani, V. Fiore, M.M. Moure Cuadrado, “Drilling characteristics and properties analysis of fiber reinforced polymer composites: A comprehensive review,” *Heliyon*, vol. 9, no. 3.
- [61] C. Li, G. Wang, Y. Jiang, H. Song, “A novel traction reducer: analysis and verification”, *Journal of Advanced Mechanical Design, Systems, and Manufacturing*, vol. 18, no. 2, pp. 1-18, 2024.
- [62] E. Kalaman, “İki Kademeli Redüktör Tasarımı Ve Optimizasyonu,” Lisans tezi, Makina Mühendisliği Bölümü, T.C Beykent Üniversitesi, İstanbul, Türkiye, 2023.