

# Parçacık Sürü Optimizasyonunda Yeni Bir Birey Davranış Biçimi Önerisi

Ö. Tolga ALTINÖZ<sup>1</sup>

A. Egemen YILMAZ<sup>2</sup>

<sup>1</sup>Endüstriyel Elektronik Bölümü, Bala Meslek Yüksekokulu, Hacettepe Üniversitesi, Ankara

<sup>2</sup>Elektronik Mühendisliği Bölümü, Mühendislik Fakültesi, Ankara Üniversitesi, Ankara

<sup>1</sup>e-posta: taltinoz@hacettepe.edu.tr

<sup>2</sup>e-posta: aeyilmaz@eng.ankara.edu.tr

## Özetçe

Bu çalışmada, Parçacık Sürü Optimizasyonu (PSO)'nda sürüdeki her bir bireyin belirli bir fonksiyonun minimum/maksimum değerini (optimum) bulurken ortaya koyduğu davranış incelenmiştir. Klasik PSO formülasyonunda bireylerin optimuma ulaşmak için ortaya koydukları davranış biçimlerine ek olarak, “kötüden kaçma” davranış biçimi modellenmiş ve formülasyona eklenmiştir. Bu çalışmada elde edilen bu yeni formülasyonun performansı, farklı boyutlardaki tek/çok modlu (*uni/multimodal*) karşılaştırma (*benchmark*) fonksiyonları kullanılarak incelenmiştir. Söz konusu incelemelerde, optimuma yakınsama eğilimi ve elde edilen sonuç değerleri, performans ölçütü olarak kullanılmıştır. İnceleme sonuçları karşılaştırmalı tablo ve grafikler aracılığıyla sunulmuş ve yorumlanmıştır.

## 1. Giriş

Parçacık Sürü Optimizasyonu (PSO) yöntemi, ilk olarak Eberhart ve Kennedy tarafından 1995 yılında ortaya atılmıştır [1]. PSO, bir hayvan (balık, kuş, böcek, vs.) sürüsünün davranışı üzerine geliştirilmiştir. Yöntem, genel olarak söz konusu sürünün, besin kaynağı ararken ortaya koyduğu davranış üzerine kuruludur. Sürüde bulunan hiçbir bireyin optimum kaynağın yerini bilmediği durumda bile sürünün tüm bireylerinin başarılı bir şekilde kaynağa ulaşabilmesinden esinlenilerek geliştirilmiş olan bu yöntem; 1) her bir bireyin kendi hatıralarında yer etmiş olan iyi konuma gitme eğilimi olarak tanımlanabilecek bilişsel (*cognitive*) davranış biçimi, 2) her bir bireyin iyi konumlarda bulunan diğer bireyleri takip etme eğilimi olarak tanımlanabilecek sosyal (*social*) davranış biçimi, 3) her bir bireyin rastgele olarak arama yapma eğilimi olarak tanımlanabilecek keşifsel (*exploratory*) davranış biçimleri arasında bir denge olduğu varsayımına dayanmaktadır [2]. Sürünün bu davranışı formüle edilerek optimizasyon problemlerinin çözümünde kullanılmaktadır.

Algoritma, en iyi olarak kendisine ilham kaynağı olan kuş sürüsü örneği ile betimlenebilir. Sürüde bulunan her bir kuşun hız ve konum değiştirmesi ile sürü, bir tarla üzerinde hareket etmektedir. Genel kabul gören ifadeye göre her bir kuş bir parçacık olarak varsayılmaktadır. Buna göre  $d$ -boyutlu bir fonksiyonun optimumu bulunurken, her bir parçacık  $d$ -boyutlu birer vektör olan başlangıç konumu ve hızı ile arama işlemine başlamaktadır [1-3].

PSO yönteminde parçacıklar, sürü içinde optimuma en yakın olan parçacığa doğru hareketlenmektedir. Bu nedenle parçacık konum bilgilerinin sürü içinde paylaşılması gerekmektedir. Bu çalışmada, her bir parçacığın diğer tüm parçacıkların konumunu bildiği ve böylelikle konum bilgisinin parçacıklar arasında değiş tokuş yapıldığı

varsayılmaktadır. En iyi konumda bulunan parçacık konum bilgisinin tüm parçacıklar arasında paylaşılması esnasında, söz konusu parçacık “*gbest*” olarak adlandırılmaktadır. Diğer konum bilgisi paylaşım yönteminde ise paylaşım yalnızca parçacıkların komşuları ile yerel (*local*) bazda gerçekleşmektedir. Bu şekilde konum bilgisini paylaşma durumunda ise yerel olarak en iyi durumda bulunan parçacık “*lbest*” olarak adlandırılır. Bu iki paylaşım biçimi arasındaki en büyük fark, yerel optimumun bulunmasında yaşanmaktadır. “*lbest*”li formülasyonda, yerel optimuma düşme durumu “*gbest*”li formülasyona göre daha az gözlenirken, “*gbest*” formülasyonu ise daha hızlı yakınsamaktadır [2-3]. Bu çalışmada “*lbest*”li formülasyona kıyasla daha hızlı yakınsadığı için “*gbest*” yaklaşımı kullanılmıştır.

Literatürde yapılan yeni çalışmalar ile, PSO formülasyonu geliştirilmektedir [4-8]. Önerilen bu yeni formülasyonlar, sürünün yakınsama hızı ve hata durumlarını iyileştirmeyi hedeflemektedir. Bazı yeni formülasyonlarda, sürünün optimuma ulaşırken  $d$ -boyutundaki uzayda taradığı bölge değiştirilmektedir. Sürü arama yaparken bulunduğu uzayda her defasında ancak belirli bir alanı taramaktadır. PSO formülasyonunda yapılacak bir iyileştirme, sürünün başlangıçta mümkün olan en geniş alanı taraması ve optimuma yaklaştıkça taranacak alanın daraltılması garanti edilerek sağlanır [2,6]. Sürü, başlangıçta mümkün olan en büyük alanı tarar; sürü içindeki parçacıklardan herhangi biri diğerlerine göre optimuma daha yakın olduğundan, sürü o parçacığa doğru hareketlenir. Optimum değere yaklaştıkça taranan alanın daraltılması performansı arttırmaktadır [4,5].

Yapılan çalışmaların büyük bir bölümü sürünün davranışı üzerine durmaktadır [6,7]. Bu çalışmada ise sürüyü oluşturan bireylerin yeni bir davranış biçimi formüle edilme ve karşılaştırmalı olarak incelenmektedir. Karşılaştırma işlemlerinde, en temel ve en yaygın olarak kullanılan PSO formülasyonlarından biri olan “atalet değerli” PSO kullanılmıştır.

Bu bildirinin içeriği şu şekilde özetlenebilir: ikinci bölümde PSO algoritmaları incelenmiştir, üçüncü bölümde ise önerilen birey davranışı formüle edilmiştir. Dördüncü ve beşinci bölümlerde ise sırası ile deneysel çalışmalar sunulmuş, karşılaştırılmış ve yorumlanmıştır.

## 2. Parçacık Sürü Optimizasyonu (PSO)

PSO algoritması, hayvan sürülerinin davranışları izlenerek geliştirilmiştir. Bu davranış bir fonksiyonun veya fonksiyonlardan meydana gelen sistemlerin belirli ölçütleri istenilen değerlere getirilmesini sağlar. Bu ölçütler PSO algoritmasında “gelişim fonksiyonu” ile ifade edilmektedir.

Tablo 1: PSO Algoritması

|  |          |
|--|----------|
| Sürüye Rastgele başlangıç hızı ve konumu ata   |          |
| <b>Do</b>  |          |
| <b>For</b> $i = 1$ to sürü boyutu<br>$F = \text{Gelişim Fonksiyonu Hesapla}$<br><b>If</b> $F < \text{en iyi değer}$<br>$i$ ninci parçacığın en iyi değeri $\leftarrow i$ ninci parçacığın konumu<br>en iyi değer $\leftarrow F$<br><b>end if</b><br><b>end for</b> | <b>1</b> |
| tüm sürüdeki en iyi değeri ve o değere sahip parçacığın konumunu bul   |          |
| <b>For</b> $i = 1$ to sürü boyutu<br>Konum ve hızı güncelle.<br><b>end for</b>   | <b>3</b> |
| <b>While</b> sonlandırma koşulları   |          |

Gelişim fonksiyonunun boyutu  $d$ , her bir parçacığın boyutuna eşittir. PSO algoritmasının, Tablo 1'de verilen işleyişi şu şekilde özetlenebilir: 1) Sürüyü oluşturan bireylere başlangıçta rastgele hız ve konum bilgisi atanır ve döngü çalıştırılır. 2) Tabloda birinci alana karşılık gelen bölümde her bir parçacığın kendi hatıralarındaki en iyi yer tespit edilir. Sürüdeki her bir birey için Gelişim fonksiyonu hesaplatılıp " $F$ " olarak tanımlanan değışkene atanır. Bu değışken bireyin geçmişteki optimum değeri ile karşılaştırılır. Bu yeni değeri eğer eskisinden daha optimal ise bu değışken içindeki değeri 'en iyi değeri' olarak atanır. 3) Tabloda ikinci alana karşılık gelir. Burada tüm sürüdeki en optimal değeri ve bu değerin konumu tespit edilir. 4) Tabloda üçüncü alana karşılık gelir. Bu alanda farklı PSO formülasyonlarına göre konum ve hız değeri güncellenir.

Yukarıda açıklandığı gibi, Tablo 1 ile verilen algoritma üç farklı alana bölünerek incelenmektedir. Bu alanlarda yapılan değışiklikler, PSO yönteminin farklı adlarda anılmasına neden olmakta ve performansta iyileştirmeler meydana getirmektedir. Algoritmada, her bir bireyin kendi hatıralarında yer etmiş olan iyi yerlere gitme eğilimi olarak tanımlanabilecek bilişsel (*cognitive*) davranış biçiminin bulunduğu kısım, birinci bölümde verilmektedir. Literatürde yapılan çalışmalar, genellikle bu bölümde herhangi bir değışiklik yapmayı önermemektedir. Bu çalışmada bu bölüm diğer bölümler ile birlikte değıştirilerek, yenilikler getirilmesi amaçlanmaktadır.

İkinci bölümde ise sürüde bulunan parçacıklardan optimuma en yakın olan parçacığın konumu bulunmaktadır; bu değeri "*lbest*" veya "*gbest*" yöntemlerinden herhangi birisi ile hesaplanabilir.

Üçüncü ve son bölüm, PSO formülasyonunu vermektedir. Literatürde bulunan birçok çalışmanın hedefi, bu alanda tanımlanan eşitlikleri geliştirmektir. Burada her bir parçacığın konumu ve hızı belirlenen bir denkleme göre değıştirilmektedir.

En basit halinde geleneksel PSO formülasyonuna göre hızın ve konumu değışimi, sırası ile (1) ve (2) numaralı denklemler ile verilmektedir.

$$v_{id} = v_{id} + c_1 * rand() * (p_{id} - x_{id}) + c_2 * rand() * (p_{gd} - x_{id}) \quad (1)$$

$$x_{id} = x_{id} + v_{id} * \Delta t \quad (2)$$

Bu denklemlerde  $i$ , sürüdeki parçacığın indeksini;  $d$ , denklem boyutunu;  $rand()$ ,  $[0, 1]$  arasında rastgele bir sayıyı;  $v$ , parçacık hızını;  $x$ , parçacık konumunu ifade eder (Simülasyon zamanındaki adım miktarı olan  $\Delta t$  değışkeninin değeri, işlem kolaylığı olması açısından tüm formülasyonlarda 1 olarak alınmaktadır). Ayrıca,  $p_{id}$ , sürüdeki  $i$ 'ninci parçacığın,  $d$ 'ninci boyutundaki hatıralarında bulunan en optimal değeri;  $p_{gd}$ ,  $d$ 'ninci boyutta, tüm sürüdeki en optimal değeri,  $x_{id}$ , sürüdeki  $i$ 'ninci parçacığın,  $d$ 'ninci boyutundaki değerini vermektedir. Burada tanımlanan  $c_1$  ve  $c_2$  katsayıları ise sırasıyla parçacıkların bilişsel ve sosyal davranışlarının ağırlıklarını belirten ölçütler olup; değeri bir çok çalışmada 2 olarak alınmaktadır [2,6,7].

(1) numaralı denklem ilk kısmı eski (mevcut) hız değerini ifade etmekte; ikinci kısım ise her bir parçacığın bilişsel davranış ve sosyal davranışlarını temsil etmektedir. Sadece ilk kısmın hız güncellenmesi için kullanılması durumunda, parçacığın başlangıçta tanımlanan sabit hız ile ilk konumundan, yönünü ve hızını değıştirmeden hareket edeceği görülebilir. Sadece ikinci kısmın uygulanması durumunda ise, parçacıkların anlık olarak optimuma en yakın parçacığa doğru hareket ettiği, şans eseri yeni bir parçacık daha optimal olur ise bir ilerleme sağlanacağı, aksi takdirde giderek küçülen bir alanda arama yapılacağı görülmektedir. Bu davranışlar, farklı PSO formülasyonlarının ortaya çıkmasına ilham vermişlerdir. Bu formülasyonlardan biri olan "atalet değeri" PSO, bu çalışmada tercih edilmiştir.

## 2.1. Atalet Değerli PSO (*Inertial Weight PSO - IWPSO*)

Bu PSO algoritması hakkında, sürünün taradığı alan genişliği hakkında aşağıdaki yorumlara ulaşılmaktadır [4-7]:

- Eski hız değeri, sürünün geniş bir alanı taramasını sağlar
- Eşitliğin ikinci kısmı ise sürünün daha dar alanı taramasını sağlar.

Elde edilen bu yorumlara ek olarak algoritmanın davranışı ile ilgili aşağıdaki isteklerde bulunulmalıdır.

Tablo 2: Yeni PSO Algoritması

|   |
|---|
| <p>Sürüyü Rastgele başlangıç hızı ve konumu ata</p> <p><b>Do</b></p> <p>    <b>For</b> <math>i = 1</math> to sürü boyutu</p> <p>        <math>F = \text{Gelişim Fonksiyonu Hesapla}</math></p> <p>        <b>If</b> <math>F &lt; \text{en iyi değer}</math></p> <p>            <math>i</math> ninci parçacığın en iyi değeri <math>\leftarrow i</math> ninci parçacığın konumu</p> <p>            en iyi değer <math>\leftarrow F</math></p> <p>        <b>end if</b></p> <p>        <b>If</b> <math>F &gt; \text{en kötü değer}</math></p> <p>            <math>i</math> ninci parçacığın en kötü değeri <math>\leftarrow i</math> ninci parçacığın konumu</p> <p>            en kötü değer <math>\leftarrow F</math></p> <p>        <b>end if</b></p> <p>    <b>end for</b></p> <p>    tüm sürüdeki en iyi değeri ve o değere sahip parçacığın konumunu bul</p> <p>    tüm sürüdeki en kötü değeri ve o değere sahip parçacığın konumunu bul</p> <p>    <b>For</b> <math>i = 1</math> to sürü boyutu</p> <p>        Konum ve hız güncellenir.</p> <p>    <b>end for</b></p> <p><b>While</b> sonlandırma koşulları</p> |
|---|

- Algoritma mümkün olduğu kadar geniş bir alanı taramalı
- Algoritma optimuma ulaşmalı

Elde edilen bu sonuçlar doğrultusunda sürünün davranışı matematiksel eşitlik olarak tespit edilmeden önce dilsel olarak ifade edilebilir. Sürünün başlangıçta geniş bir alanı taraması istenmelidir ve sürü optimuma yaklaştıkça taranan alan küçülmeli ve bu sayede optimum elde edilebilmelidir. Bu işlemin yapılabilmesi için eski hız değerinin eşitlik içinde kontrol edilmesi gerekmektedir. Bu durumu geleneksel PSO algoritmasına eski hız değeri bir katsayı ile çarpılarak IWPSO formülasyonu olarak adlandırılan eşitliklere ulaşılmaktadır.

$$v_{id} = (w)v_{id} + c_1 * rand() * (p_{id} - x_{id}) + c_2 * rand() * (p_{gd} - x_{id}) \quad (3)$$

$$w = w_{\max} - \frac{w_{\max} - w_{\min}}{j_{\max}} * j \quad (4)$$

$w_{\max}$  ve  $w_{\min}$  değerleri sırası ile 0.9 ve 0.4 değerlerine sahiptir. Değişkenlerin aldığı bu değerler daha önce yapılan deneysel çalışmalar sonucunda elde edilmiştir [4-5].  $j$  değişkeni mevcut iterasyonunu indeksini,  $j_{\max}$  ise maksimum iterasyon sayısını ifade etmektedir.

### 3. “Kötüden Kaçma” Davranışı ve Matematiksel İfadesi

PSO formülasyonu, tüm sürünün optimumu bulunurken sergilediği davranış aracılığı ile elde edilmektedir. Şu ana kadar yapılmış ve yayınlanmış olan birçok çalışmada, her bir parçacığın davranışı yerine sürü davranışını temel alınmıştır. Bu çalışmanın amacı ise, sürü davranışlarının yanında yeni bir birey davranışının formüle edilmesidir.

Hayvanlarda olduğu gibi tüm canlı sınıflarında bireyler en iyiye ulaşmaya çalışmaktadır. Bu davranış, her bir canlının hayatını devam ettirebilmek için ihtiyaç duyduklarını elde etme çabasıdır. Bu davranış bir kuş için en iyi yuva yerini seçmek, bir balık için en korunaklı mercanı bulmak veya insanlar için en iyi evi seçmek olabilir. Bu seçimlerdeki kriterler bireyin ihtiyaç duyduklarını karşılayacak en iyi hedefin ulaşılması olarak görülmektedir. Fakat bu davranışın yanında bireylerin karşılaştıkları durumlarda yalnızca en iyiye gitmek için çaba sarf etmedikleri

gözlemlenebilir. Her bir birey aynı zamanda kötü olandan da uzaklaşmaya çalışmaktadır.

Birey davranışı iyiye gitme biçiminde olduğu gibi kötüden de uzaklaşma biçimini almaktadır. (1) numaralı denklemin ikinci kısmı, bireyin bulunduğu noktadan daha iyi bir noktaya gitme davranışını matematiksel olarak ifade etmektedir. Bu durumda bu eşitlikte kötüden kaçmak için fazladan bir çaba bulunmamakta, daha ziyade iyiye gitme davranışı gözlemlenmektedir. “Kötüden Kaçma” bilinci, denklem içine eklenmeden önce sözel olarak ifade edilmelidir. O halde her bir birey kendi hatıralarında yer etmiş olan kötü yerlerden uzaklaşacak ve kötü konumlarda bulunan diğer bireylerden uzaklaşmak isteyecektir. Bu hareket için üretilen algoritma Tablo 2’de verilmiştir.

Tablo 2 ile verilen Yeni PSO Algoritmasının işleyişi şu şekilde özetlenebilir: 1) Sürüyü oluşturan bireylere başlangıçta rastgele hız ve konum bilgisi atanır ve döngü çalıştırılır. 2) Her bir parçacığın kendi hatıralarındaki en iyi ve en kötü yer tespit edilir. Sürüdeki her bir birey için Gelişim fonksiyonu hesaplatılıp “ $F$ ” olarak tanımlanan değişkene atanır. Bu değişken bireyin geçmişteki en iyi ve en kötü değeri ile karşılaştırılır. Bu yeni değer eğer eskisinden daha optimal ise bu değişken içindeki değer ‘en iyi değer’ ve/veya ‘en kötü değer’ olarak atanır. 3) Burada tüm sürüdeki en iyi ve en kötü değer ve bu değerlerin konumu tespit edilir. 4) Bu alanda, konum ve hız değerlerinin güncellenmesi amacıyla (5) ve (6) numaralı denklemler kullanılır.

$$v_{id} = (w) * v_{id} + c_1 * rand() * (p_{id} - x_{id}) + c_2 * rand() * (p_{gd} - x_{id}) + c_3 * rand() * (x_{id} - k_{id}) + c_4 * rand() * (x_{id} - k_{gd}) \quad (5)$$

$$x_{id} = x_{id} + v_{id} \quad (6)$$

(5) numaralı denklem ile verilen hız güncelleme eşitliğinde (3) numaralı denklemden farklı olarak  $k_{id}$  ve  $k_{gd}$  değerleri eklenmiştir. Bu değerler, sırası ile her bir bireyin kendi hatıralarında yer etmiş olan kötü konum ile sürünün en kötü konum bilgilerini vermektedir. (3) ve (5) numaralı denklemler arasındaki farklılıklar, “kötüden kaçma” bilincinin matematiksel ifadesini vermektedir.

Tablo 3: Karşılaştırma Fonksiyonları

| Fonksiyon Adı            | Fonksiyon  | Tanımlı olduğu aralık |
|--------------------------|--|-----------------------|
| Küre (Sphere) Fonksiyonu | $f_1(x) = \sum_{i=1}^n x_i^2$  | [-100,100]            |
| Griewank Fonksiyonu      | $f_7(x) = \frac{1}{4000} \sum_{i=1}^n (x_i - 100)^2 - \prod_{i=1}^n \cos\left(\frac{x_i - 100}{\sqrt{i}}\right) + 1$ | [0,600]               |
| Rosenbrock Fonksiyonu    | $f_9(x) = \sum_{i=1}^{n-1} \left(100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2\right)$  | [-30,30]              |
| Rastrigin Fonksiyonu     | $f_{10}(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$  | [-5.12,5.12]          |

Bu çalışmada “atalet değerli” PSO formülasyonu (bir başka deyişle (3) ve (4) numaralı denklemler) ve bu formülasyonun değiştirilmiş biçimi (bir başka deyişle (5) ve (6) numaralı denklemler) karşılaştırma işlemlerine tabi tutulmuştur. Literatürde karşılaştırma işlemleri için kullanılan bir başka formülasyon ise “daralma faktörlü” PSO olarak adlandırılmaktadır. Söz konusu formülasyon ile bu çalışmada kullanılan formülasyon, parametre değerlerindeki farklılıklar göz ardı edilirse aynı yapıya sahiptir [8]. Bu çalışmada iyileştirmenin daha iyi görülebilmesi için “daralma faktörlü” PSO yerine “atalet değerli” PSO formülasyonu seçilmiştir.

#### 4. Deneysel Çalışmalar

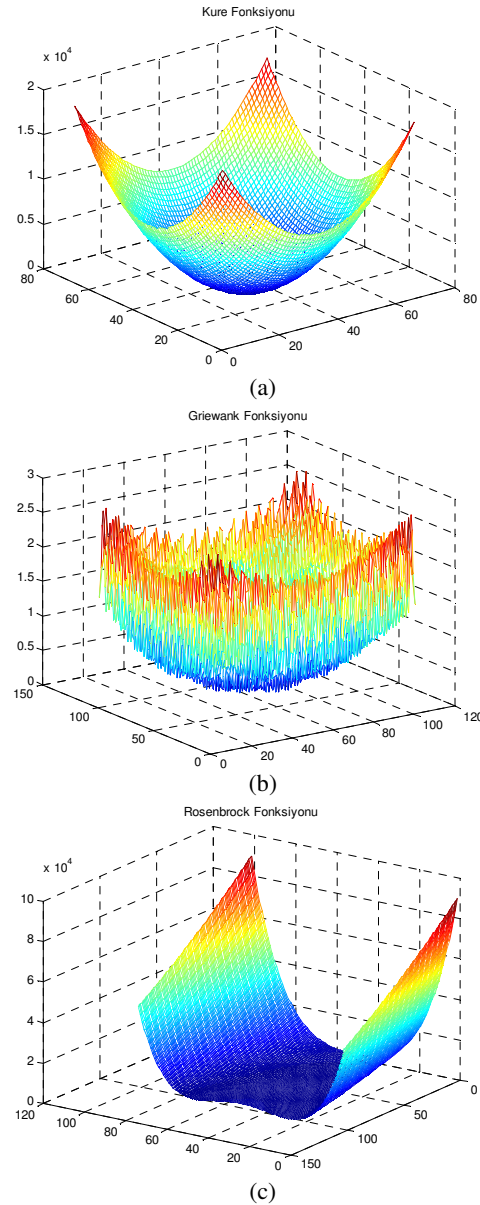
Bu çalışmada, literatürde olan diğer çalışmalarda olduğu gibi “karşılaştırma fonksiyonu” adı verilen fonksiyonlar kullanılarak yeni üretilen formülasyon incelenmektedir. “Karşılaştırma fonksiyonları” optimizasyon algoritmalarının karşılaştırılmasında kullanılmaktadır. Literatürde birçok karşılaştırma fonksiyonu tanımlanmıştır. Bu fonksiyonlar, yerel optimum sayılarına ve sıklıklarına göre sınıflandırılmaktadır. Bu çalışmada ise bu fonksiyonlar içinden en sık kullanılan dört temel fonksiyon seçilmiştir. Elde edilen sonuçlar, yakınsama zamanı ve ortalama en iyi hata değerlerine göre karşılaştırılmakta ve yorumlanmaktadır. Karşılaştırma işlemi, “atalet değerli” PSO formülasyonu ile önerilen formülasyon arasında yapılmaktadır. Sonuçlar, her bir değerlendirme fonksiyonu için incelenmektedir. Tablo 3 karşılaştırma fonksiyonlarını ve her bir fonksiyonun tanımlı olduğu aralığı vermektedir. Şekil 1 ise Tablo 3’de verilen karşılaştırma fonksiyonlarının iki boyutlu grafiklerini göstermektedir.

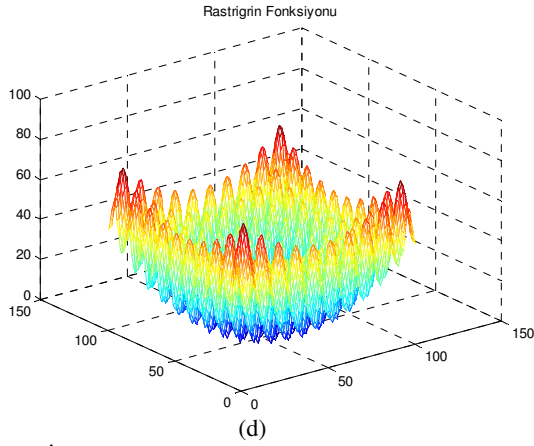
Sonuçlar iki temel test yardımı ile incelenmektedir. Bu testler sırası ile yakınsama testi ve sonuç testi olarak adlandırılmaktadır. Yakınsama testi PSO formülasyonunda, belirli sayıdaki iterasyon sonunda sonucun kalitesinin yorumlanmasıdır. Burada formülasyonun sonunda, “Optimum değerine ne kadar yaklaşmıştır?” sorusu yanıtlanmaktadır. Genel yaklaşım olarak, herhangi bir formülasyon mümkün olan en kısa iterasyon sonucunda optimuma ulaşması beklenmektedir.

Diğer bir test olan sonuç testi, farklı formülasyonların her bir fonksiyon için elde edilen sonuçlar üzerinden sonuç kalitesinin yorumlanmasıdır. Burada yapılan işlem, gerçekleştirilen çok sayıda bağımsız PSO formülasyon deneyinden bir formülasyonun ürettiği optimum değer ortalamalarını karşılaştırmaktır. Tablo 4 ile her iki formülasyon için geçerli olan katsayı değerleri verilmektedir.

Tablo 4: PSO Formülasyon Parametreleri

| Parametreler                      | Değerleri |
|-----------------------------------|-----------|
| Sürü Sayısı                       | 500       |
| $c_1=c_2$                         | 2         |
| $c_3=c_4$                         | 1         |
| $V_{max}$                         | 1         |
| İterasyon                         | 1000      |
| Deneme sayısı                     | 100       |
| Karşılaştırma fonksiyon boyutları | 10,20,30  |





Şekil 1. İki boyutlu karşılaştırma fonksiyonları a) Küre b) Griewank c) Rosenbrock d) Rastrigin Fonksiyonları

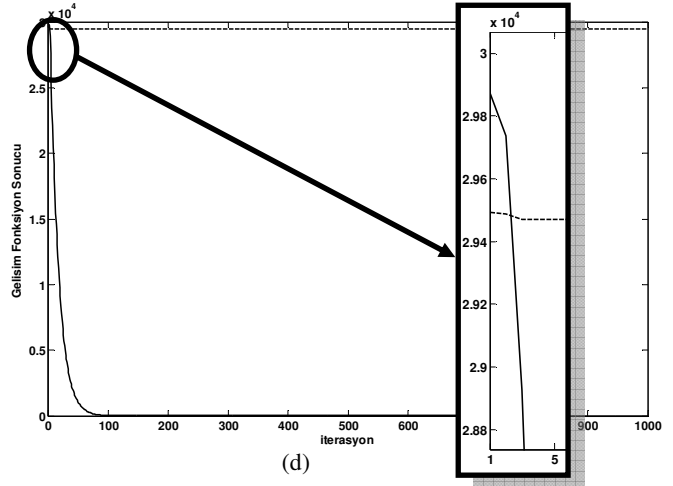
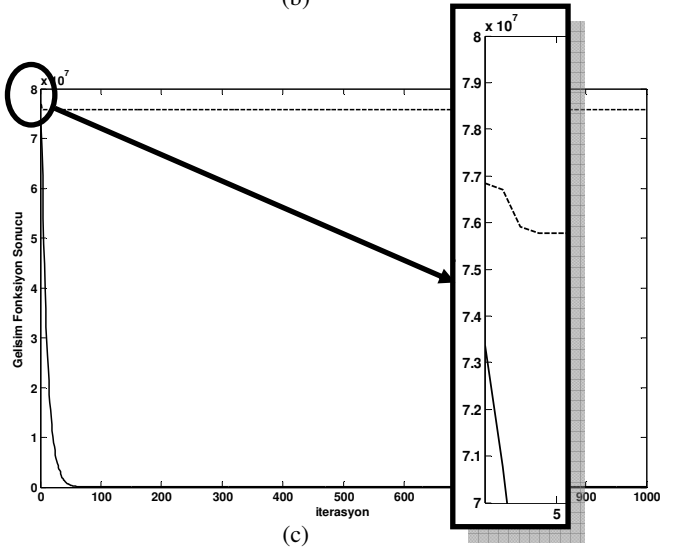
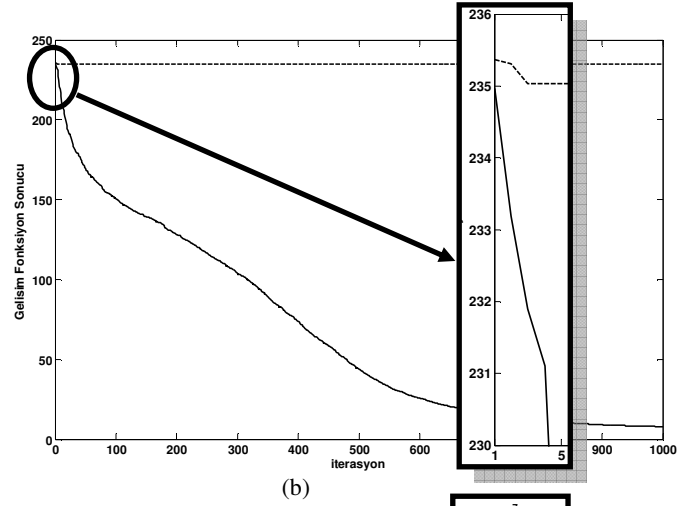
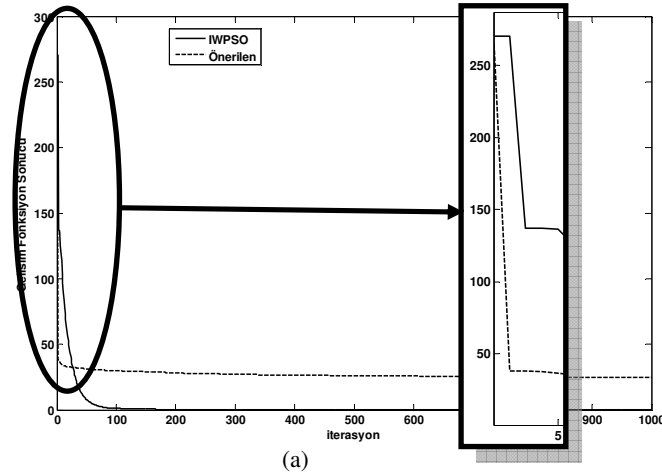
#### 4.1. Yakınsama Testi

Bu bölümde her iki formülasyon, Tablo 3’de verilen 10, 20 ve 30 boyutlarındaki karşılaştırma fonksiyonlarına uygulanmakta ve yakınsamaları karşılaştırılmaktadır (Yer kısıtı nedeniyle, sadece 20 boyutlu fonksiyonlara dair grafikler verilmektedir. Farklı boyutlarda da aynı yakınsama davranışları gözlemlenmiştir).

Şekil 2’de karşılaştırma fonksiyonlarının yakınsama grafikleri görülmektedir (Kesikli çizgiler yeni PSO formülasyon sonucunu göstermektedir). Grafikler incelendiğinde önerilen formülasyonun Griewank fonksiyonunda başlangıç iterasyonları için daha iyi sonuç verdiği görülmektedir. Diğer fonksiyonlar için ise önerilen formülasyon daha iyi sonuç vermese bile başlangıç iterasyonlarında değerler daha iyiye gitmektedir.

Önerilen PSO formülasyonu IWPSO formülasyonuna kıyasla daha kötü çalıştığı grafiklerden görülmektedir. Fakat önerilen formülasyonun Griewank fonksiyonu için ilk iterasyon sonucunda hızlı yakınsadığı görülmektedir.

Önerilen formülasyon diğer fonksiyonlar için ise ilk iterasyonlarda ani iyileşmelere neden olmaktadır. Bunun temel nedeni, bireylerin ilk anda en kötüden uzaklaşması; dolayısıyla tüm sürünün optimuma doğru anlık olarak itilmesi olarak açıklanabilir.



Şekil 2. Karşılaştırma fonksiyonları sonuçları a) Griewich b) Rastrigin c) Rosenbrock d) Küre Fonksiyonları

## 4.2. Sonuç Testi

Her iki formülasyon için gerçekleştirilen denemeler sonucunda ortalama hata değerleri Tablo 5’de verilmektedir.

Tablo 5: Sonuç testi için elde edilen veriler

| Boyut | Fonksiyonun Adı   | Ortalama |          |
|-------|-------------------|----------|----------|
|       |                   | IWPSO    | Önerilen |
| 10    | <i>Sphere</i>     | 2.05e-40 | 8.19e3   |
|       | <i>Griewank</i>   | 0.064    | 8.3718   |
|       | <i>Rosenbrock</i> | 3.3066   | 8.41e6   |
|       | <i>Rastrigin</i>  | 0.5877   | 87.0294  |
| 20    | <i>Sphere</i>     | 5.40e-20 | 2.97e4   |
|       | <i>Griewank</i>   | 0.0363   | 25.0531  |
|       | <i>Rosenbrock</i> | 67.1316  | 7.6e7    |
|       | <i>Rastrigin</i>  | 8.3975   | 236.43   |
| 30    | <i>Sphere</i>     | 4.35e-12 | 5.36e4   |
|       | <i>Griewank</i>   | 0.0122   | 46.21    |
|       | <i>Rosenbrock</i> | 575.15   | 1.7e8    |
|       | <i>Rastrigin</i>  | 34.0836  | 390.43   |

Bu değerler ışığında, IWPSO formülasyonunun önerilen algoritmaya oranla daha kaliteli sonuçlar verdiği görülmektedir. Önerilen formülasyon sonuçlarının kaliteli olmamasına rağmen farklı fonksiyonlar ve bu fonksiyonların farklı boyutları için, IWPSO formülasyonu ile aynı azalış ve artış davranışı sergilediği görülmektedir.

## 5. Sonuç

Bu çalışmada genellikle sürü karakteristiği incelenen parçacık sürü optimizasyonu algoritmasında birey davranışlarına odaklanılmış, yeni “kötüden kaçma” davranışı açıklanarak matematiksel olarak ifade edilmiştir. Elde edilen eşitlikler daha sonra farklı boyuttaki karşılaştırma fonksiyonları için çalıştırılmış; literatürde bulunan bir PSO türevi (IWPSO) ile karşılaştırmalar yapılmıştır. Karşılaştırmalarda, algoritmaların bulunduğu optimum sonuçlar ve söz konusu sonuçlara yakınsama eğilimleri incelenmiştir.

Bu karşılaştırmalar sonucunda, tarafımızdan önerilmekte olan formülasyonun, ilk iterasyonda ani bir çözüm iyileştirmesi sağladığı görülmüştür. Özellikle Griewank fonksiyonunda, bu iyileştirme etkisinin çok yüksek olduğu gözlenmiştir. Elde edilen nihai sonuçların (optimumların) ortalaması incelendiğinde ise, IWPSO formülasyonunun daha başarılı olduğu görülmektedir.

Bu sonuçlar ışığında, tarafımızdan önerilen formülasyonun tüm PSO iterasyonları boyunca değil; belirli bir katsayı ile ve sadece sınırlı bir iterasyon aralığı boyunca sisteme uygulanmasının sonuçları iyileştireceği söylenebilir. Bir sonraki çalışmada, bu katsayı ve iterasyon aralığı tespit edilecektir. Ayrıca önerilen davranış biçimi, farklı PSO türevleri için de uygulanarak algoritmanın genel performansında iyileştirme elde edilmeye çalışılacaktır. Bu çalışmada kullanılmış olan ‘*gbest*’ formülasyonu yerine ‘*lbest*’ formülasyonu kullanılarak yapılacak olan incelemeler, sonraki çalışmaların ana konusunu teşkil edecektir.

## 6. Kaynakça

- [1] Kennedy, J. and Eberhart, R., “Particle Swarm Optimization”, *IEEE International Conference on Neural Networks*, p 1942-1948, 1995.
- [2] Eberhart, R. and Kennedy, J. “A New Optimizer Using Particle Swarm Theory”, *Sixth International Symposium on Micro Machines and Human Science*, p 39-43, 1995.
- [3] Kennedy, J. and Eberhart, C. “A Discrete Binary Version of the Particle Swarm Algorithm”, *IEEE International Conference on Systems, Man, and Cybernetics*, p 4104-4108, 1997.
- [4] Shi, Y. and Eberhart, R. “A Modified Particle Swarm Optimizer”, *IEEE Congress on Evolutionary Computation*, p 69-73, 1998.
- [5] Shi, Y. and Eberhart, R. “Empirical Study of Particle Swarm Optimization”, *IEEE Congress on Evolutionary Computation*, p 1945-1950, 1999.
- [6] Clerc, M. “The Swarm and The Queen: Towards a Deterministic and Adaptive PSO”, *IEEE Congress on Evolutionary Computation*, p 1951-1957, 1999.
- [7] Clerc, M. and J. Kennedy, J. “The Particle Swarm – Explosion, Stability, and Convergence in a Multidimensional Complex Space”, *IEEE Transactions on Evolutionary Computation*, Vol 6. No. 1, p 58-73, 2002.
- [8] Eberhart, R. and Shi, Y. “Comparing Inertia Weights and Constriction Factors in Particle swarm Optimization”, *IEEE Congress on Evolutionary Computation*. p 84-88, 2000.