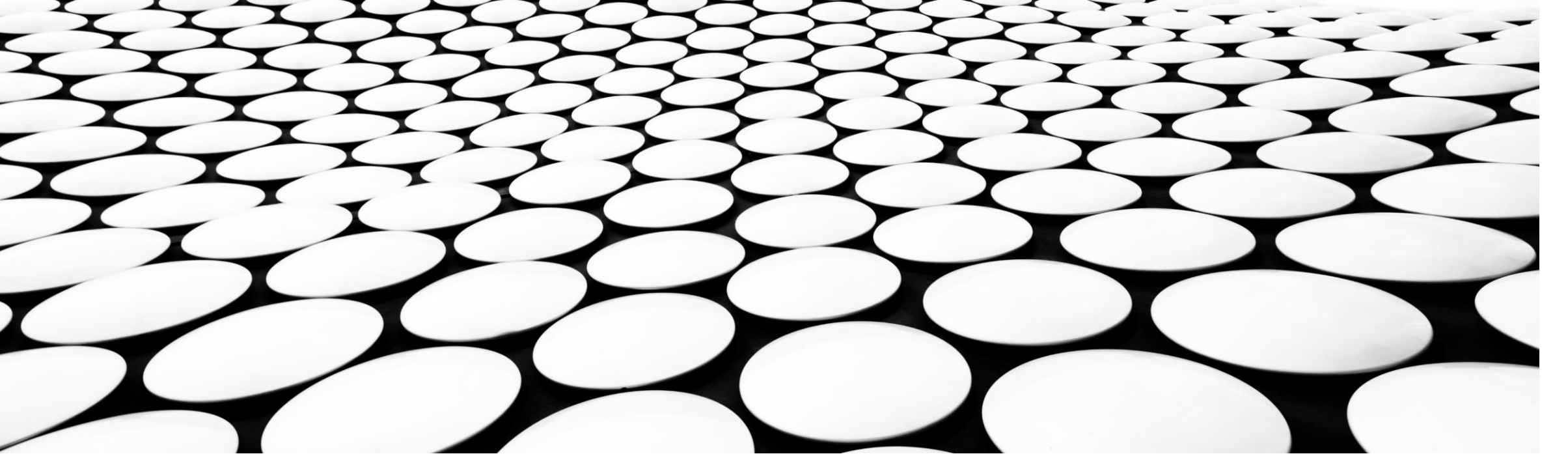


# **YAPAY ZEKA**

## **PARÇACIK SÜRÜ OPTİMİZASYONU**



# OPTİMİZASYON NE DEMEKTİR?

- Optimizasyon karşılaşılan problemlere kısıtlamalar dahilinde mümkün olan en iyi sonucu sunabilme olarak ifade edilebilir.
- Örnek vererek daha anlaşılır olmasını sağlayalım:

Bir yatırımcı olduğunuzu ve paranızı yatırabileceğiniz 25 farklı seçeneğinizin olduğunu varsayalım. İçlerinden üçünü seçmek istiyorsunuz. Böyle bir durum için toplam 2300 seçeneğiniz olur. Bir yatırımcı olarak en fazla kar elde etmek istediğinizi de hesaba katarsak tüm bu seçenekleri tek tek denemenin kardan çok uzak olduğunu söylemek hiç de zor olmayacaktır. Kaldı ki piyasa durumu farklı hesaplamaları da beraberinde getirebilir. Bu noktada ise optimizasyon devreye girerek mümkün olan en karlı yatırımı yapmanıza yardımcı olur.



# PARÇACIK SÜRÜ OPTİMİZASYONU NEDİR?

- Doğa her zaman bilim için ilham kaynağı olmuştur. Aktarmaya çalıştığımız parçacık sürü optimizasyon konusu da bu duruma örnektir.
- Dr.Kennedy ve Dr.Eberhart, bazı hayvanların; yiyecek bulmak, korunmak gibi temel ihtiyaçlarını karşılamak için beraber hareket ederek her bireyin ortak fayda sağlamasına katkıda bulunan sürü hareketlerini gözlemleyip bu davranış biçimini bilgisayar bilimine uygulamaları sonucu parçacık sürü optimizasyonu adı verilen algoritmayı oluşturulmuşlardır.
- Peki kuş veya balık gibi hayvanların sürü hareketleri bize optimizasyon konusunda ne anlatıyor olabilir?
- Optimizasyonun tanımından da hatırlanacağı gibi kısıtlamalar dahilinde mümkün olan en iyi çözüme ulaşabilmeyi hedef olarak gösterebiliriz. Bazı balık ve kuş türlerinde gözlemlenen sürü hareketlerinde sürü içerisindeki hayvanların problem karşısında iletişim kurarak ve de birbirlerinin konumlarını takip ederek çözüme ulaşmaları güzel bir optimizasyon örneğidir. Buradaki problemi bir balık sürüsünün avcıdan kaçması olarak kabul edersek; kısıtlamayı süre ve çözümü sürüde avcıdan en uzak konumdaki balığın konumuna ulaşmak olarak örnekleyebiliriz.



# HAYVAN SÜRÜLERİNİN GÖRÜNÜŞÜ



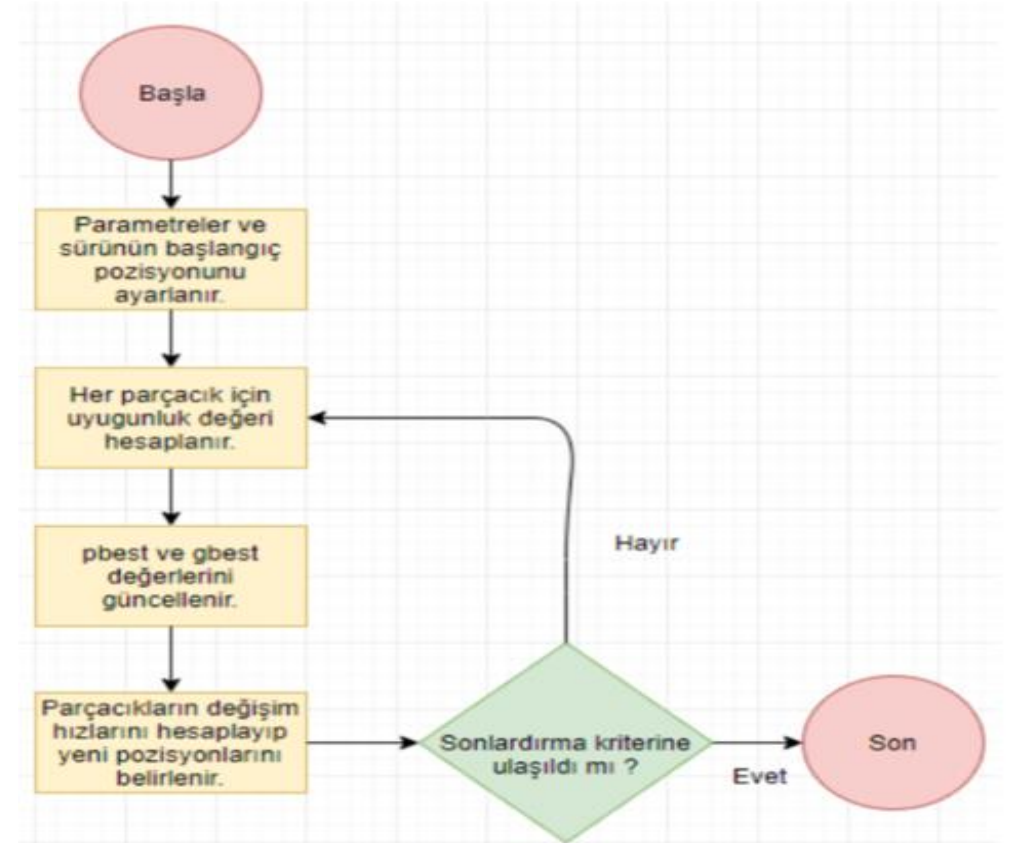
Resim 1: Sığircık kuşları



Resim 2: Köpekbalığından kaçan balık sürüsü

# PSO ALGORİTMASININ İŞLEYİŞİ

1. Sürünün başlangıç konumu ve parametrelerine rastgele değerler atanır.
2. Bütün parçacıkların uygunluk fonksiyonuna göre uygunluk değerleri hesaplanır.
3. Her bir parçacığın pbest değeri bulunur.
4. Sürünün geneline bakılarak gbest değeri bulunur.
5. Sürünün konumu ve parametreleri tekrar hesaplanır.
6. İstenilen değer bulunduysa durulur bulunmadı ise 2.adımdan devam edilir.



Şekil 1:PSO Algoritması



## FONKSİYON TANIMI

- Sürüdeki herhangi bir parçacığın yeni hız değerini hesaplamak için şu fonksiyon kullanılır.
- $v_{i+1} = wv_i + c_1(rand_1)(pbest - x) + c_2(rand_2)(gbest - x)$
- Bu fonksiyondaki değişkenler şöyle ifade edilebilir.
- $v$ : Parçacığın hızı.
- $w$ : Parçacığın şu an ki hızının yeni hıza etkisini belirten değer 0.4 ile 1.4 arasında değer alabilir.
- $x$ : Parçacık değeri.
- $rand_1$  ve  $rand_2$ : Rastgele üretilen değerler.
- $c_1$ : Parçacığın pbest değerinin yeni hızı ne kadar etkileyeceğini belirten katsayı 1.5 ile 2 arasında değer alabilir.
- $c_2$ : Parçacığın gbest değerinin yeni hıza ne kadar etki edeceğini belirten katsayı 2 ile 2.5 arasında değer alabilir.
- $pbest$ : Parçacığın çözüme en çok yaklaştığı durum.
- $gbest$ : Tüm parçacıklar arasında çözüme en çok yaklaşılan durum.

## ÖRNEK UYGULAMA

Örnek problemimiz verilen denklemin sonucunu 0 yapacak  $x$  değerini bulmak olsun. Burada denklemi bir işin maliyet fonksiyonu olarak düşünebilirsiniz.

$$f(x) = x^2 + 2x - 3$$

- *Öncelikle kaç adet parçacık ile çözümü arayacağımızı belirliyoruz.*

Bu sayı arama uzayının genişliğine ve sizin isteğinize bağlı olarak belirlenir. Biz bu örneğimizde 3 belirleyelim.

Parçacıklar rastgele belirlenir. Parçacıkları aşağıda verildiği gibi sırasıyla **3, 7, 5** olarak belirledik.

$$P_1 = 3$$

$$P_2 = 7$$

$$P_3 = 5$$

- *Her parçacığın uygunluk değeri hesaplanır.*

Bu örnekte amacımız problemin denklemini 0'a yaklaştırmak olduğundan uygunluk fonksiyonumuz problemin denkleminin ta kendisidir.

## ÖRNEK UYGULAMA

$$f(3) = 12$$

$$f(7) = 60$$

$$f(5) = 32$$

Görüldüğü üzere belirlediğimiz  $x$  değerlerini denklemde yerine koyarak uygunluk değerlerini kolayca elde ettik. Tekrar hatırlatmak gerekirse; amacımız uygunluk değeri 0 olan  $x$  değerini bulmaktır.

- ***$pbest$  ve  $gbest$  değerleri hesaplanır.***

Şuan ilk iterasyonda olduğumuzdan parçacıkların kendileri zaten  ***$pbest$*** 'leridir.  ***$gbest$***  ise 0'a en yakın olan  ***$P1$***  parçacığıdır.

- ***$c1$ ,  $c2$  değerleri ve rastgele olarak  $rand1$ ,  $rand2$  değerleri belirlenir.***

$c1$  ve  $c2$  değerleri genellikle 2 belirlenir ben de bu parametreleri 2 olarak belirliyorum.

***$rand1$***  ve  ***$rand2$***  değerini de hesaplama kolaylığı açısından 2 belirliyorum.

Ayrıca daha ilk iterasyonda olduğumuzdan parçacıklar herhangi bir hıza sahip değildir. Bu nedenle ilk değişim hızı  ***$V0$*** 'ı 0 kabul ediyoruz.



## ÖRNEK UYGULAMA

Nihayet sıra değişim fonksiyonumuzu kullanarak parçacıkların değişimini hesaplamaya geldi

- *Parçacıkların değişim hızları hesaplanır.*

$$P_1 \rightarrow 0 + 2 * 2 * (3 - 3) + 2 * 2 * (3 - 3) = 0$$

$$P_2 \rightarrow 0 + 2 * 2 * (7 - 7) + 2 * 2 * (3 - 7) = -16$$

$$P_3 \rightarrow 0 + 2 * 2 * (5 - 5) + 2 * 2 * (3 - 5) = -8$$

Böylece formülümüzü kullanarak parçacıkların değişimi hesaplanmış oldu.

- *Parçacıkların yeni değerleri belirlenir.*

Bu aşamada parçacıklar, değişim değerleri ile toplanarak yeni parçacıklar belirlenir.

$$P_1 \rightarrow 3 + 0 = 3$$

$$P_2 \rightarrow 7 - 16 = -9$$

$$P_3 \rightarrow 5 - 8 = -3$$

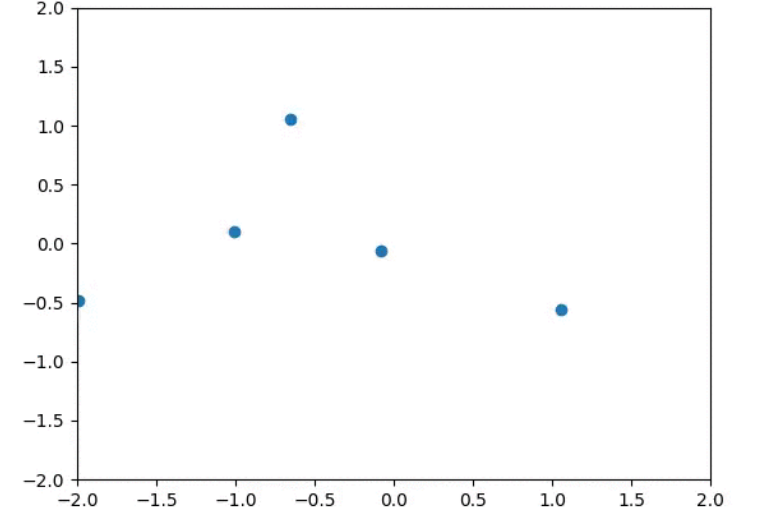
## ÖRNEK UYGULAMA

- *Yeni parçacıkların uygunluk değerleri bulunur.*

$$f(3) = 12$$

$$f(-9) = 34$$

$$f(-3) = 0$$



Tebrikler. Daha ilk iterasyonda bırakın problemin çözümüne yaklaşmayı çözdünüz bile

Denklemimizi 0 yapan değer **-3** olarak bulunmuş oldu.

Eğer çözüme ulaşamamış olsaydık; yeni parçacıklar da göz önüne alınarak yeniden ***pbest*** değeri ve bu zamana kadar gelmiş tüm ***pbest***'lerin en iyisi olan ***gbest*** değeri belirlenecekti. Ek olarak **rand1** ve **rand2** değerleri tekrar belirlenip parçacıkların değişimi hesaplanacak ve yeni parçacıklar bulunacaktı. Ve bir kez daha uygunluk değerini bulup çözüme ne kadar yaklaştığımızı değerlendirecektik.

## PSO ALGORİTMASININ ZAYIF YÖNÜ

PSO Algoritmaları her ne kadar problemlerin sonucunu bulabilseler de birden fazla çözümü olan problemlerin çözüm uzayını bulmakta yetersiz kalırlar.

Herhangi bir parçacık çözümlerden birisine çok yaklaştığında diğer parçacıklarda onu takip edecek ve bulunan o çözüm etrafında toplanmaya başlayacaktır.

Bu nedenle diğer çözümler bulunamayacak ya da diğer çözümün gölgesinde kalacaklardır.



Resim 3: Parçacıkların hareketi

Resimdeki mavi noktalar parçacıkları, diğer noktaların ise problemin çözümünü ifade ettiğini düşünürsek bütün parçacıkların bir çözüm etrafına kümelendiği açıkça görülebilir.

## PSO ALGORİTMASININ GÜÇLÜ YÖNÜ

- PSO Algoritmaları her ne kadar çözüm uzayı geniş problemleri çözmekte yetersiz kalsa da bu eksikliğini kolay uygulanabilirliği ile telafi eder.
- PSO Algoritmalarında parçacıkların yeni değerleri belirlenirken yapılan hesaplamalar; herhangi bir diferansiyel denklem ya da gradyan hesap gibi karmaşık hesaplar gerektirmediği için hızlı çalışır.
- PSO Algoritmalarında kullanılan uygunluk fonksiyonlarında ayarlanması gereken az sayıda parametre olduğundan algoritmanın çalışması hızlıdır.
- Yapay sinir ağı eğitimlerinde geri yayılım(back propagation) algoritmalarından daha hızlı olduğu için tercih edilir.

## PSO ALGORİTMASININ KULLANILDIĞI YERLER

- Fonksiyon optimizasyonu
- Çizelgeleme
- Yapay Sinir Ağlarının Eğitimi
- Bulanık Mantık Sistemleri
- Görüntü İşleme
- Anten Dizayn

Gibi non-lineer denklem içeren bir çok alanda kullanılan bir algoritmadır.



## KAYNAKÇA

- Resim 1 : <https://i.ytimg.com/vi/08v446aEj-0/maxresdefault.jpg>
- Resim 2 : [https://img1.aksam.com.tr/fotogaleri/haber\\_icerik/images/130820170027505683748.jpg](https://img1.aksam.com.tr/fotogaleri/haber_icerik/images/130820170027505683748.jpg)
- Resim 3 : <https://www.muhendisbeyinler.net/parcacik-suru-optimizasyonu/>
- <https://medium.com/deep-learning-turkiye/par%C3%A7ac%C4%B1k-s%C3%BCr%C3%BC-optimizasyonu-pso-1402d4fe924a>
- <https://medium.com/@hamzaerguder/par%C3%A7ac%C4%B1k-s%C3%BCr%C3%BC-optimizasyonu-24e01beec438>