

# particle swarm optimization

## GİRİŞ:

### What is Swarm Intelligence(Sürü Zekası nedir)?

- Sürü terimi, günlük görevlerini akıllı ve verimli bir şekilde yerine getirmek için kolektif olarak çalışan hayvanların veya böceklerin toplanmasını temsil eder.
- Sistemler SI sistemleri tipik olarak birbirleriyle ve çevreleriyle yerel olarak etkileşime giren basit bir ajan popülasyonundan oluşur.
- SI'nın doğal örnekleri arasında karınca kolonileri, kuş sürüsü, hayvan sürüsü, bakteri üremesi ve balık eğitimi bulunur.

### Parçacık Sürüsü Optimizasyonunun Kökeni

Partikül Sürüsü Optimizasyonu (PSO), 1995 yılında Dr. Russell C. Eberhart ve Dr. James Kennedy tarafından kuş sürüsü veya balık eğitiminin sosyal davranışlarından esinlenerek popülasyon tabanlı bir stokastik optimizasyon tekniğidir.

Dr.Eberhart ve Dr.Kennedy, kuşların ve balıkların akın ve okullaşma modellerinden ilham aldı. Başlangıçta bu ikisi, gıda kaynakları etrafında kuş akınının bilgisayar yazılım simülasyonlarını geliştirmeye başladı, daha sonra algoritmalarının optimizasyon problemleri üzerinde ne kadar iyi çalıştığını fark etti.

### Parçacık Sürüsü Optimizasyonu Kavramı

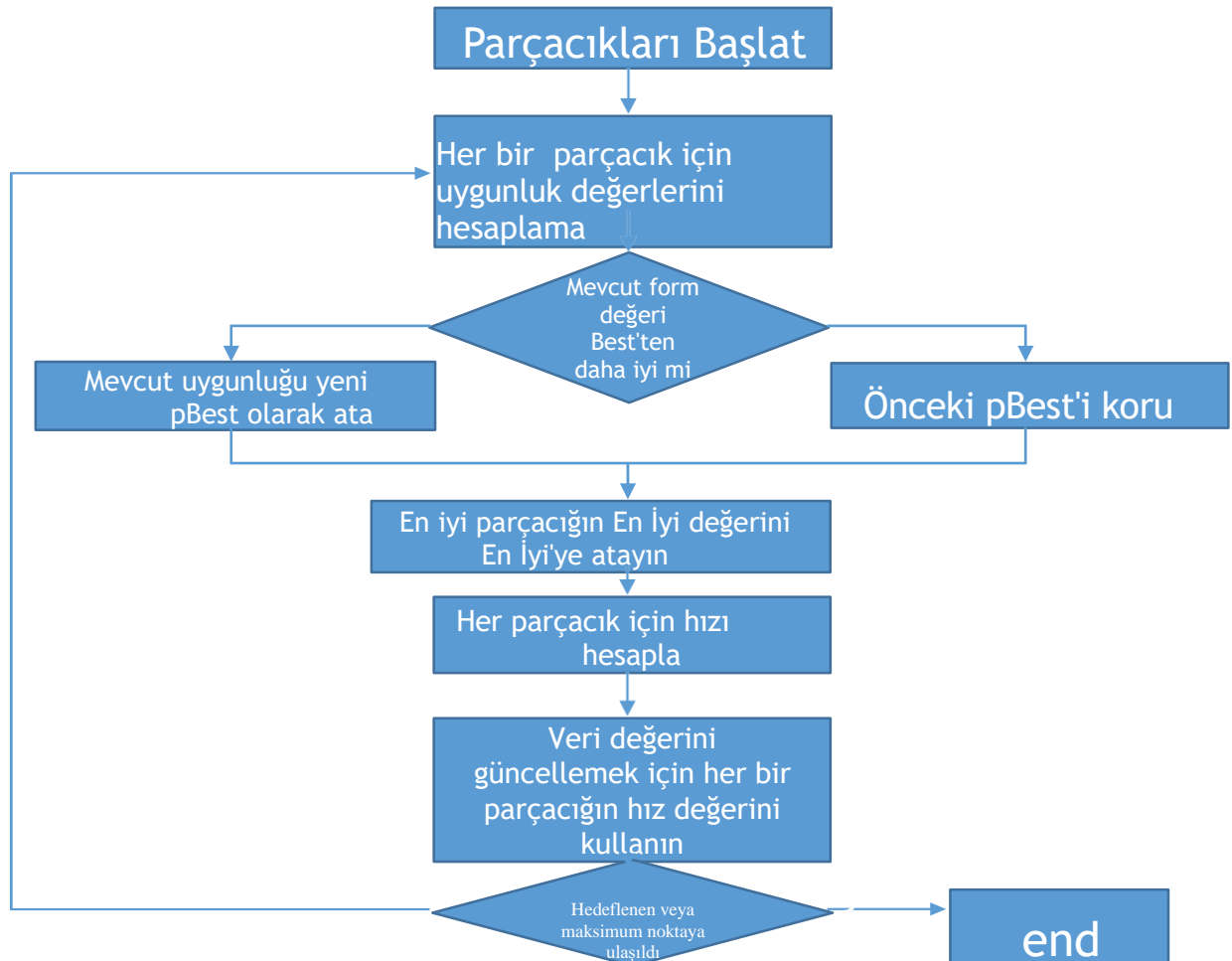
- PSO, son derece zor veya imkansız sayısal maksimizasyon ve minimizasyon problemlerine yaklaşık çözümler bulmak için kullanılabilecek yapay bir zeka (AI) tekniğidir.
- O PSO'da n kişiden oluşan bir grup, doğrudan veya dolaylı olarak birbirleriyle arama yönleri (gradyanlar) ile iletişim kurar.
- Algorithm Basit algoritma, uygulanması kolay ve esas olarak hızı ayarlamak için birkaç parametre.

## ALGORİTMA :

### PSO'daki parametreler

- Rastgele konular ( $X_i$ ) ve hızlar ( $V_i$ ) atanarak başlatılan nüfus; potansiyel çözümler daha sonra hiperuzaydan geçer
- Her bir partikül hiperuzaydaki “en iyi” (en yüksek uygunluk) pozisyonunu takip eder. Buna denir
  - ❑ “bireysel bir parçacık için en iyisi.
  - ❑ en iyi grup için "gbest".
  - ❑ mahallede en iyisi için "en iyi".
- Her bir zaman adımında, her bir parçacık en iyi ve en iyi (ya da en iyi) yönünde stokastik olarak hızlanır.

### AKIŞ SEMASI:



## Matematiksel Yaklaşım

Denklem:

$$V_i[k+1] = V_i[k] + C_1 \times \text{Rand}(\cdot) \times (pbest_i[k] - X_i[k]) + C_2 \times \text{rand}(\cdot) \times (gbest[k] - X_i[k]) \quad (1)$$

$$X_i[k+1] = X_i[k] + V_i[k+1] \quad (2)$$

$V_i = [vi1, vi2, \dots, vin]$  partikül  $i$  için hız denir.

$X_i = [xi1, xi2, \dots, xin]$ , partikül  $i$ 'nin pozisyonunu temsil eder.

$Pbest$ : partikül  $i$ 'nin önceki en iyi pozisyonunu temsil eder (yani, yerel-en iyi pozisyon veya deneyimi)

$Gbest$ :  $X = [X1, X2, \dots, XN]$  popülasyondaki tüm parçacıklar arasındaki en iyi pozisyonu temsil eder. . . ,  $XN$ ] (yani küresel en iyisi durum)

$\text{Rand}(\cdot)$  Ve  $\text{rand}(\cdot)$ :  $[0,1]$  arasında iki rastgele değişkendir.

$C1$  ve  $C2$ : her bir parçacığı sırasıyla en iyi ve sürünün en iyi konumlarına doğru yönlendiren ivme katsayıları adı verilen pozitif sayılardır.

$C1$ : 1.5 ile 2 arasında değer alır.

$C2$ : 2.0 ile 2.5 arasında değer alır.

Optimizasyon nedir:

Karşılaşılan problemlere kısıtlamalar dahilinde mümkün olan en iyi sonuç verebilmesidir.

Örnek vererek daha iyi anlayalım:

Buradaki problemi bir balık sürüsünün avcıdan kaçması olarak kabul edersek; kısıtlamayı süre ve çözümü sürüde avcıdan en uzak konumdaki balığın konumuna ulaşmak olarak örnekleyebiliriz.

## PSO Sözde Kodu

For her parçacık:

Parçacığı başlat

Do :

For her parçacık:

Uygunluk değerini hesapla

Kondisyon değeri tarihteki en iyi kondisyon değerinden (pBest) daha iyi ise

Mevcut değeri yeni pBest olarak ayarla

End

For each particle :

Parçacık mahallesinde bulun, en iyi zindeliğe sahip parçacık Hız denkleminde göre parçacık hızını hesaplayın (1) Hız daralmasını uygulayın

Partikül pozisyonunu pozisyon denkleminde göre güncelleyin (2)

Konum daralmasını uygulayın

End

While maksimum tekrarlar veya minimum hata ölçütlerine ulaşamıyor

## PSO yapısındaki değişiklikler

maksimum hız seçimi:

Parçacıklar kontrolsüz hale gelecek ve arama alanını aşacak şekilde hızlar çok yüksek olabilir. Bu nedenle, hızlar maksimum  $V_{max}$  değerine, yani

durağanlık ağırlığı ekleme:

Aramanın kapsamını daha iyi kontrol etmek için PSO için atalet ağırlığı adlı yeni bir  $w$  parametresi eklenir. Yani, Denk. (1)

Constriction factor(daralma faktörü):

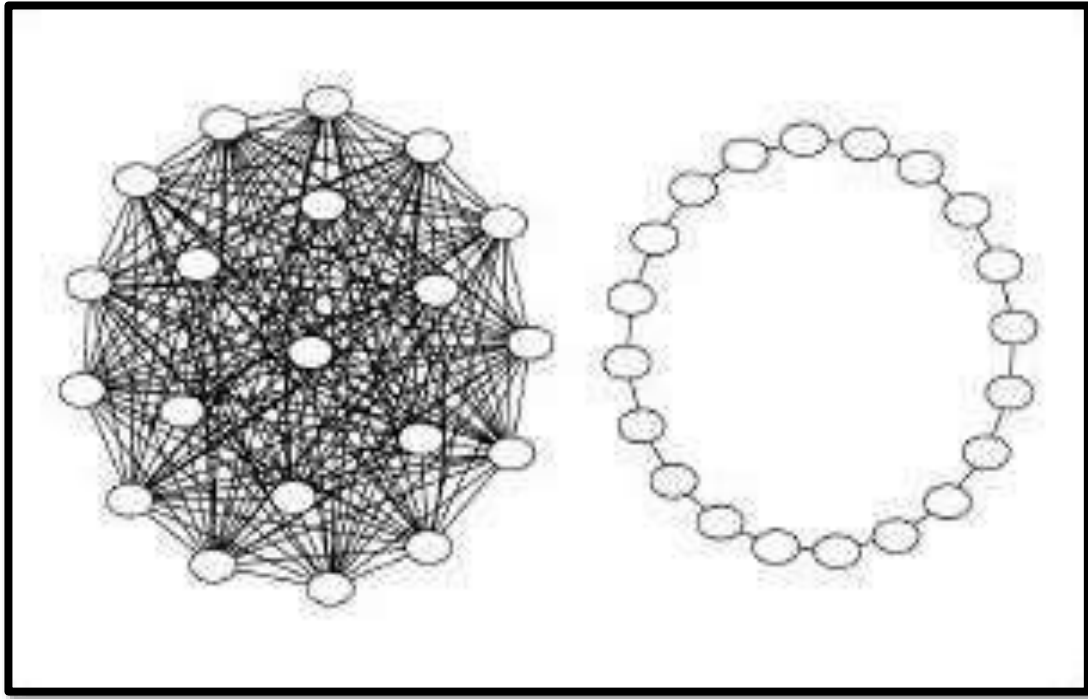
Algoritma hızı kısıtlamadan çalıştırılıyorsa, sistem birkaç yinelemeler. Yani, daralma faktörü özellikleri indükler. Daralma faktörü ile, hızı hesaplamak için PSO denklemi:

Deneme sonucu :

$C = 5$  ise, denklemden  $\chi = 0,38$ . (4), sönümleme etkisi çok net neden olacaktır. Ancak,  $C$  4 olarak ayarlanırsa,  $x = 0.729$ 'dur ve iyi çalışır.

### Nüfus Topolojisi

- Bireyler arasındaki bağlantı modeli bir sosyal ağ gibidir.
- Bağlantı modeli, çözümlerin çözüm boşluğundan nasıl akabileceğini kontrol eder.



### Effect of Re-Initialization(Yeniden Başlatma Etkisi)

- Üç algoritma arasında PSO'nun hızla kümelenme eğilimi vardır ve sürüler hızla durgunlaşabilir. Bu dezavantajı gidermek için, küresel en iyi parçacığın baskın etkisini azaltmak için birkaç alt gruplandırma yaklaşımı önerilmiştir. Çok daha basit ve sık kullanılan bir alternatif, küresel en iyi parçacığı tutmak ve geri kalan parçacıkların tümünü veya bir kısmını yeniden oluşturmaktır. Bu, yeni bir sürü üretme etkisine sahiptir, ancak parçacıklardan biri olarak küresel en iyisi ile bu süreçte yeniden başlatma işlemi denir.
- GA'da kümelenme daha az belirgindir, ancak genellikle nüfusun üst kısmının benzer görünebileceği ve çeşitliliğin artırılması için yeniden başlatmanın popülasyona rasgele enjekte edebileceği bulunmuştur.
- DE'de kümeleme en azdır ve yeniden başlatma DE için en az etkiye sahiptir

## Effect of Local Search(Yerel Aramanın Etkisi)

- GA'da, çözelti alanının yoğunluğu daha azdır, bu nedenle GA operatörlerinin tüm potansiyel çözümleri üretemedikleri görülür. Popüler bir düzeltme, GA operatörleri tarafından üretilen çözümler çevresinde daha iyi bir çözüm bulunup bulunmadığını görmek için yerel aramanın kullanılmasıdır. Yerel arama süreci genellikle zaman alıcıdır ve bunu tüm popülasyona uygulamak uzun bir çözüm süresine yol açabilir.
- PSO için, en iyi parçacığın tüm sürü üzerinde baskın bir etkisi vardır ve zaman tasarrufu stratejisi, sadece en iyi parçacığa yerel arama uygulamaktır ve bu, daha kısa çözüm süresi ile çözüm iyileştirmesine yol açabilir. Bu stratejinin Pratchayaborirak ve Kachitvichyanukul'da (2011) atölye planlaması için oldukça etkili olduğu gösterilmiştir.
- En iyi parçacık olduğu için aynı strateji DE'de aynı etkiyi vermeyebilir çözüm popülasyonu üzerinde baskın bir etkisi yoktur.

## Effect of Subgrouping(Alt Gruplamanın Etkisi)

- Çözelti kalitesini artırmak için homojen popülasyonun alt gruplarının kullanılması GA ve PSO'da gösterilmiştir.
- Bu alt gruplama, bazı çözüm gruplarının baskın çözümlerin etkisinden kurtulmasına izin verir ve bu nedenle grup çözüm alanının farklı bir alanında araştırma yapabilir ve algoritmaların keşif yönünü geliştirebilir.
- DE için, en iyi parçacığın pertürbasyon süreci üzerinde çok az etkisi vardır, bu nedenle homojen popülasyonlu alt gruplamanın DE'nin çözelti kalitesi üzerinde sınırlı bir etkiye sahip olabileceğini varsaymak mantıklıdır.

## GA, PSO ve DE'nin kalitatif karşılaştırması

### GA, PSO ve DE'nin kalitatif karşılaştırması

	GA	PSO	DE
Çözüm Sıralaması İste	Yes	No	No
Nüfus büyüklüğünün çözüm süresine etkisi	Üstel	Linear	Linear
En iyi çözümün nüfus üzerindeki etkisi	Medium	Most	Less
Ortalama zindelik daha kötü olamaz	False	False	True
Erken yakınsama eğilimi	Medium	High	Low
Arama alanının sürekliliği (yoğunluğu)	Less	More	More
Yerel arama olmadan iyi bir çözüm elde etme yeteneği	Less	More	More
Homojen alt grupta yakınsamayı artırır	Yes	Yes	NA

## UYGULAMA :

### PSO kullanarak Sinir Ağı (NN) Eğitimi

- Bazı sayısal girişleri kabul eden ve bazı sayısal çıktılar üreten karmaşık bir işlev.
- PSO kullanarak bir nöral ağı nasıl eğitildiği hakkında bir fikir edinmenin en iyi yolu, hedefin sepal uzunluğa göre türü tahmin etmek olan bir grup Iris çiçeği için bir sinir ağı tahmincisi oluşturan bir programa bakmaktır. ve genişlik ve taçyaprağı uzunluğu ve genişliği.
- Program, ünlü bir 150- Fisher'ın "İris Verileri" adı verilen öge karşılaştırma veri kümesi.
- A 4-input, 6-hidden, 3-output neural network is instantiated.
- A 4-girişli, 6-gizli, 3-çıkışlı bir sinir ağı somutlaştırılmıştır.
- Tam bağlı bir 4-6-3 sinir ağı  $(4 * 6) + (6 * 3) + (6 + 3) = 51$  ağırlık ve bias değerleri.
- A 12 sanal parçacıktan oluşan bir sürü ve sürü, maksimum 700 yinelemede sinir ağı ağırlıkları ve sapma değerleri kümesini bulmaya çalışır.

- PSO eğitimi tamamlandıktan sonra, bulunan en iyi ağırlık ve sapmaların 51 değeri görüntülenir. Bu ağırlıkları ve önyargıları kullanarak, sinir ağı altı eğitim ögesiyle beslendiğinde, ağıdır.

## Parçacık Sürüsü Optimizasyonu ve Uyarlanabilir NN Kullanarak Mobil Robot Navigasyonu

- Geliştirilmiş parçacık sürüsü optimizasyonu (PSO), hareketli bir robotun statik engeller içeren bir ortamda yolunu optimize etmek için kullanılır.
- Düzgün olmayan yollar üreten birçok optimizasyon yöntemiyle karşılaştırıldığında, PSO yöntemi, mobil robotları kullanarak aşağıdaki yolu gerçekleştirmek üzere sürekli kontrol teknolojileri tasarlamak için daha tercih edilen düz yollar üretebilir.
- Hesaplama optimizasyon maliyetini azaltmak için, yüksek keşif kabiliyetine sahip stokastik PSO (S-PSO) geliştirilir, böylece küçük boyutlu bir sürü, yol planlamayı başarabilir.

## PSO 'nun Diğer Evrimsel Tekniklerle Hibridizasyonu

Popüler bir araştırma eğilimi, PSO'yu diğer tekniklerle, özellikle de diğer tekniklerle birleştirmek veya birleştirmektir. seçim, çapraz geçiş ve mutasyon gibi evrimsel hesaplama teknikleri. Geliştirilmiş ve genişletilmiş PSO yöntemlerinden bazıları: Geliştirilmiş PSO (IPSO): PSO'nun hem arama hem de kullanım kapasitesini artırmak için kaotik sekansların ve geleneksel lineer olarak azalan atalet ağırlıklarının ve çapraz çalışmanın bir kombinasyonunu kullanır. Modifiye PSO (MPSO): Bu yaklaşım eşitlik ve eşitsizlik kısıtlamalarıyla başa çıkmak için bir mekanizmadır. Ayrıca, optimizasyon sürecini hızlandırmak için dinamik bir arama alanı azaltma stratejisi kullanılır. Yeni PSO (NPSO): Bu yöntemde, parçacık en kötü konumunu hatırlamak için değiştirilir. Bu değişiklik, arama alanını çok etkili bir şekilde keşfetmek için geliştirilmiştir. Geliştirilmiş Koordineli Toplama tabanlı PSO (ICA-PSO): Bu algoritmada, sürüldeki her parçacık, şimdiye kadar karşılaşılan en iyi konumunun bir hafızasını korur ve sadece parçacık hariç, rastgele hareket eden en iyi başarı. Sıralı Kuadratik Programlamalı Hibrit PSO (PSO-SQP): SQP yöntemi, kısıtlı optimizasyon için en iyi doğrusal olmayan programlama yöntemi gibi görünmektedir. Çok sayıda test probleminde verimlilik, doğruluk ve başarılı çözümlerin yüzdesi açısından diğer tüm doğrusal olmayan programlama yöntemlerinden daha iyi performans gösterir.



## PSO 'nun güçlü yönleri

- PSO Algoritmaları her ne kadar çözüm uzayı geniş problemleri çözmekte yetersiz olsa da bu eksikliği kolay uygulanabilirliği ile telafi eder .
- PSO Algoritmalarında yeni değerlerini 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 eğitimlerinde geri yayılım algoritmalarından daha hızlı olduğu için tercih edilir.

## PSO 'nun zayıf yönleri

- 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.

## PSO 'nun kullanım alanları

1. Çizelgeleme
2. Yapay sinir ağları eğitimi
3. Bulanık mantık sistemleri
4. Görüntü işleme
5. Anten dizayn
6. Fonksiyon optimizasyonu

### ÖRNEK UYGULAMA:

Problemimiz verilen denklemin sonucunu 0 yapacak  $x$  değerini bulmak olsun.

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

$$F(x)=x$$

- **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.

$$f(3) = 12$$

$$f(7) = 60$$

$$f(5) = 32$$

Görüldüğü üzere belirlediğimiz  $x$  değerlerini denklemden 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.**

$c_1$  ve  $c_2$  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  $V_0$ 'ı 0 kabul ediyoruz.

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.

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

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

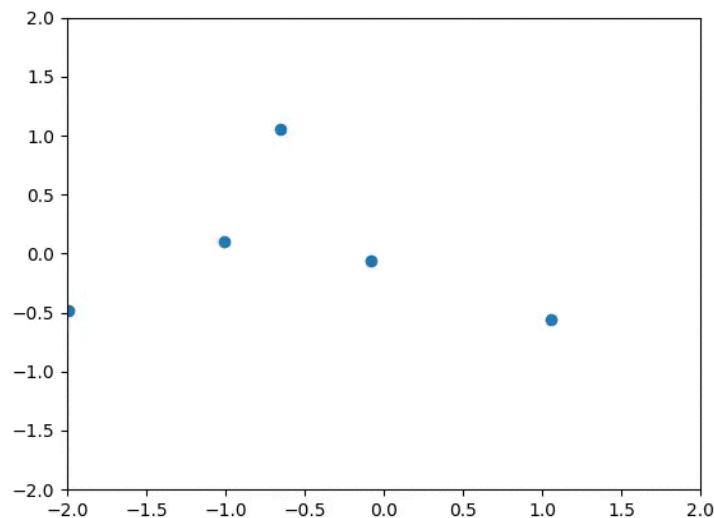
$$P_3 \rightarrow 5 - 8 = -3$$

■ ***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 ününe 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.

## SONUÇ :

- ◇ PSO algoritmasının optimal değerler bulma süreci, lideri olmayan bir hayvan toplumunun çalışmasını takip eder.
- ◇ Partikül sürüsü optimizasyonu, partikülün potansiyel bir çözümü (daha iyi durum) temsil ettiği bir partikül sürüsünden oluşur.
- ◇ Parçacık, o alandaki en iyi konumu bulmak için çok boyutlu bir arama alanında hareket edecektir (en iyi konum maksimum veya minimum değerlere mümkün olabilir).
- ◇ Akılda tutulması gereken kısıtlamalar, hızın optimum değere sahip olması gerektiğidir, çünkü çok daha az çok yavaş olacaktır ve hız çok yüksekse, yöntem become unstable.

## KULLANILAN KAYNAKLAR:

@[www.swarmintelligence.com](http://www.swarmintelligence.com)

@[www.visualstudiomagazine.com](http://www.visualstudiomagazine.com)

@[www.youtube.com](http://www.youtube.com)

@msdn.microsoft.com

@oduction to Particle Swarm Optimization by Rajib Kumar Bhattacharjee, Department of Civil Engineering,IITGuwahati

@Paper on Particle Swarm Optimization by Gerhard Venter,VanderplaatsResearchanddevelopmentInc.

@ <https://i.ytimg.com/vi/08v446aEj-0/maxresdefault.jpg>

@ [https://img1.aksam.com.tr/fotogaleri/haber\\_icerik/images/130820170027505683748.jpg](https://img1.aksam.com.tr/fotogaleri/haber_icerik/images/130820170027505683748.jpg)

@ <https://medium.com/deep-learning-turkiye/par%C3%A7ac%C4%B1k-s%C3%BCr%C3%BC-optimizasyonu-pso-1402d4fe924a>

@<https://www.muhendisbeyinler.net/parcacik-suru-optimizasyonu/>