

A large flock of birds, likely starlings, is captured in flight against a vibrant sunset sky. The birds are arranged in a dense, V-shaped formation, their silhouettes glowing with the orange and yellow light of the setting sun. The sky transitions from a deep blue at the top to a bright yellow near the horizon. The overall scene conveys a sense of natural optimization and collective movement.

PARÇACIK SÜRÜ OPTİMİZASYONU

*Russell Eberhart
James Kennedy*

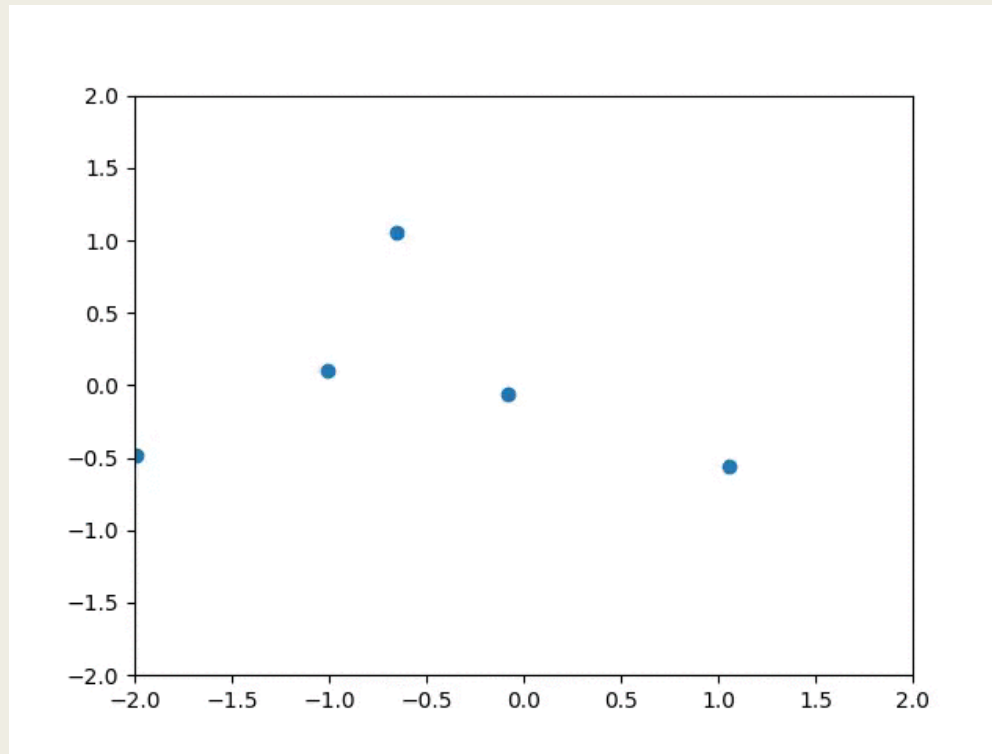
PARÇACIK SÜRÜ OPTİMİZASYONU (PARTICLE SWARM OPTIMIZATION)

- Parçacık Sürü Optimizasyonu (PSO), sürü halinde hareket eden bazı hayvanların yiyecek bulmak gibi temel ihtiyaçlarını giderirken sergiledikleri hareketlerin, sürüdeki diğer bireyleri etkilediğinin ve sürünün amacına daha kolay ulaştığının gözlemlenmesinden esinlenilerek Dr. Kennedy ve Dr. Eberhart tarafından 1995 yılında geliştirilmiş bir optimizasyon algoritmasıdır.
- Kuşların, yerini bilmedikleri yiyeceği aramaları, bir probleme çözüm aramaya benzetilir. Kuşlar yiyecek ararken yiyeceğe en yakın olan kuşu takip ederler. Parçacık olarak adlandırılan her tekil çözüm, arama uzayındaki bir kuştur. Çözüm uzayında her boyuttaki hızın ve yönün her seferinde nasıl değişeceği, komşularının en iyi koordinatları ve kendi kişisel en iyi koordinatlarının bir birleşimi olacağı kabul edilerek optimum noktaya ulaşmaya çalışılır.

PARÇACIK SÜRÜ OPTİMİZASYONU

- Algoritma basitleştirilmiş bir ortamın simülasyonu olarak başlamıştır.
- Paradigmalar birkaç bilgisayar kodu satırında uygulanır.
- Sadece ilkel matematiksel operatörler gerektirir ve hem bellek gereksinimi hem de hız açısından hesaplama açısından ucuzdur.
- Sistem belirli bir araştırma alanında rastgele çözümlerden oluşan bir popülasyonla başlatılır ve en iyi çözüm için jenerasyonları güncelleyerek arama yapar
- PSO'da parçacık (Particle) denilen potansiyel çözümler, mevcut en iyi çözümleri takip ederek problem uzayında gezinirler.(Uçuş yaparlar)

PARÇACIK SÜRÜ OPTİMİZASYONU



PARÇACIK SÜRÜ OPTİMİZASYONU

- Arama işlemi genetik algoritmalarda olduğu jenerasyon sayısınca yapılır. Her bireye parçacık denir ve parçacıklardan oluşan popülasyona da sürü (swarm) denir.
- Her bir parçacık kendi pozisyonunu, bir önceki tecrübesinden yararlanarak sürüdeki en iyi pozisyona doğru ayarlar. PSO, temel olarak sürüde bulunan birey-lerin pozisyonunun, sürünün en iyi pozisyona sahip olan bireyine yaklaştırılmasına dayanır.
- Çoğu zaman sürü içinde bulunan bireyler yeni hareketlerinde bir önceki konumdan daha iyi konuma gelirler ve bu süreç hedefe ulaşınca kadar devam eder.
- PSO, Sipariş miktarı belirleme, çizelgeleme problemleri, güç ve voltaj kontrolü, motor parametrelerini belirleme, tedarik seçimi ve sıralama problemleri gibi bir çok optimizasyon problemlerinde başarı ile kullanılmıştır.

PARÇACIK SÜRÜ OPTİMİZASYONU

Algoritma temel olarak aşağıdaki basamaklardan oluşur;

- 1) Rasgele üretilen başlangıç pozisyonları ve hızları ile başlangıç sürüsü oluşturulur.
- 2) Sürü içerisindeki tüm parçacıkların uygunluk değerleri hesaplanır.
- 3) Her bir parçacık için mevcut jenerasyondan yerel en iyi (pbest) bulunur. Sürü içerisinde en iyilerin sayısı parçacık sayısı kadardır.

4) Mevcut jenerasyondaki yerel en iyiler içerisinde küresel en iyi (gbest) seçilir.

5) Pozisyon ve hızlar aşağıdaki gibi yenilenir.

- Burada X_{id} pozisyon ve V_{id} hız değerlerini verirken, $rand_1$ ve $rand_2$ değerleri rasgele üretilmiş sayılardır. W atalet ağırlık değeri ve C_1 (exploration), C_2 (exploitation) ölçeklendirme faktörleridir.

$$V_{id} = W * V_{id} + c_1 * rand_1 * (P_{id} - X_{id}) + c_2 * rand_2 * (P_{gd} - X_{id})$$

$$X_{id} = X_{id} + V_{id}$$

- Durdurma kriteri sağlanıncaya kadar 2,3,4,5 adımları tekrar edilir.
- C_1 değeri arttıkça global arama, C_2 değeri arttıkça yerel arama kapasitesi artar.
- W değeri exploration ve exploitation ayarlaması yapar. Genellikle 0.9 dan 0.4 veya 0.2 ye lineer olarak azalır.
- W değeri arttıkça global arama, azaldıkça yerel arama kapasitesi artar.

- s sürünün boyutu olsun. Her i parçacığı, birkaç karakteristiğe sahip bir nesne olarak temsil edilebilir.
- Bu karakteristikler aşağıdaki sembollere gösterilir:
- X_i : Parçacığın mevcut pozisyonu;
- V_i : Parçacığın mevcut hızı;
- Y_i : Parçacığın kişisel en iyi pozisyonu.
- i parçacığıyla ilişkilendirilmiş kişisel en iyi pozisyon, parçacığın ziyaret ettiği (bir önceki x_i değeri) ve bu parçacık için en yüksek uygunluk değerini veren en iyi pozisyonudur.
- Bir minimizasyon işi için daha düşük bir fonksiyon değeri sağlayan bir pozisyon daha yüksek uygunluğa sahip kabul edilir.
- f sembolü minimize edilen amaç fonksiyonunu göstermek üzere kişisel en iyi pozisyon için güncelleme denklemi t zaman aralığına bağlı olarak aşağıda gösterilmiştir.

$$y_i(t+1) = \begin{cases} y_i(t) & , f(x_i(t+1)) \geq f(y_i(t)) \\ x_i(t+1) & , f(x_i(t+1)) < f(y_i(t)) \end{cases}$$

- c_1 ve c_2 'nin değerleri $0 < c_1, c_2 \leq 2$ sabitleriyle sınırlanır.
- Ancak bilişsel katsayının olarak da isimlendirilen c_1 'in biraz daha büyük seçilmesi ve $c_1 + c_2 = 4$ durumunun sağlanması halinde daha iyi sonuçların alınabileceği gösterilmiştir.
- Buradan c_1 'in bu parçacığın kişisel en iyi pozisyonunun yönünde adım büyüklüğünü ayarladığı, c_2 'nin de global en iyi parçacığın yönünde maksimum adım büyüklüğünü ayarladığı açıktır.
- Ayrıca V_{maks} arama uzayı büyüklüğünün yarısına eşitlenebilir

$$v_{i,j}(t+1) = v_{i,j}(t) + c_1 r_{1,j}(t) [y_{i,j}(t) - x_{i,j}(t)] + c_2 r_{2,j}(t) [\hat{y}_j(t) - x_{i,j}(t)]$$

- Her parçacığın pozisyonu, bu parçacık için yeni hız vektörü kullanılarak $x(t+1) = x(t) + v(t+1)$ şeklinde güncellenir.

Başlangıç Parametreleri

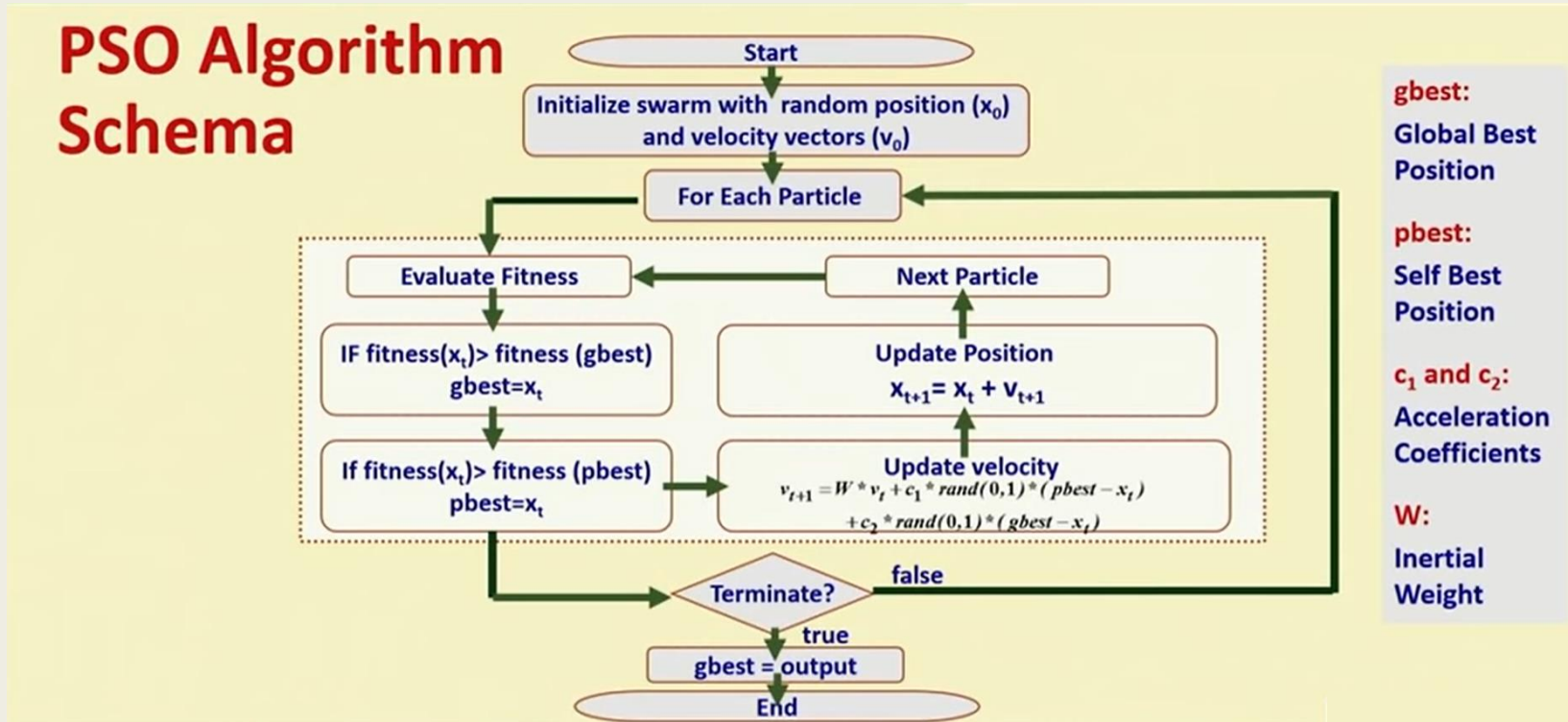
- Sürü sayısı
- Parçacıkların pozisyonu
- Parçacıkların hızları
- İterasyon sayısı

Kontrol Parametreleri

- Sürü sayısı
- Ataletsel ağırlık (w)
- İvme katsayıları C_1 ve C_2
- İterasyon sayısı

- Her parçacık en iyi (optimum) değeri arar.
- Her parçacık hareket etmektedir ve bi hızı vardır.
- Her parçacık önceki en iyi pozisyonlarını hafızada tutarlar.
- Sürüdeki parçacıklar birlikte çalışır. Ziyaret edip keşfettikleri yerlerin bilgilerini birbirleri ile paylaşırlar.
- Basitçe paylaşım ;
 - *Her parçacık bağlı (ilişkili) olduğu komşu parçacığa sahiptir.*
 - *Parçacık komşu çözümlerin fitness (Uygunluk) değerlerini bilir ve en iyi fitness değerine sahip komşusunun pozisyonunu kullanır.*
 - *Bu pozisyon parçacığın hızını ayarlamak için kullanılır.*

AKIŞ DİYAGRAMI



Pseudo code of PSO

Initialize the controlling parameters (N , $c1$, $c2$, $Wmin$, $Wmax$, $Vmax$, and $MaxIter$)

Initialize the population of N particles

do

for each particle

calculate the objective of the particle

Update PBEST if required

Update GBEST if required

end for

Update the inertia weight

for each particle

 Update the velocity (V)

 Update the position (X)

end for

while the end condition is not satisfied

Return GBEST as the best estimation of the global optimum

Sayısal Örnek

Maksimizasyon

$$F(x)=1+2x-x^2$$

$W=0.70$ $C_1 = 0.20$ $C_2 = 0.60$ $n=5$ (Beş sürü parçacığı) $E=4$ İterasyon

Rasgele hız ve konum değerleri oluşturulur.

$$\mathbf{r}_1 = [0.4657, 0.8956, 0.3877, 0.4902, 0.5039]$$

$$\mathbf{r}_2 = [0.5319, 0.8185, 0.8331, 0.7677, 0.1708]$$

Sürü Parçacıklarının Başlatılması

Parçacıkların şu anki konumu

- $\mathbf{CP}(\mathbf{0}) = 10x[r_1 - 0.5]$
- $\mathbf{CP}(\mathbf{0}) = 10x[\{0.4657, 0.8956, 0.3877, 0.4902, 0.5039\} - 0.5]$
- $\mathbf{CP}(\mathbf{0}) = [-0.3425, 3.9558, -1.1228, -0.0981, 0.0385]$

Bazı parçacıkların 1 den büyük olabilmesi için 10 ile çarpıldı.

Negatif ve pozitif pozisyonlar elde edebilmek için 0.5 çıkarıldı.

Hızların Başlatılması

- $\mathbf{V}(\mathbf{0}) = [r_2 - 0.5]$
- $\mathbf{V}(\mathbf{0}) = \{[0.5319, 0.8185, 0.8331, 0.7677, 0.1708] - 0.5\}$
- $\mathbf{V}(\mathbf{0}) = [0.0319, 0.3185, 0.3331, 0.2677, -0.3292]$

1. Iterasyon

- $\mathbf{CP(0)} = \mathbf{CP(1)} = [-0.3425, 3.9558, -1.1228, -0.0981, 0.0385]$
- Anlık hız $\mathbf{V(0)} = \mathbf{V(1)} = [0.0319, 0.3185, 0.3331, 0.2677, -0.3292]$
- Uygunluk değeri $\mathbf{F(CP(1))} = 1 + 2 * CP(1) - CP(1)^2$
 $= [0.1976, -6.7378, -2.5061, 0.7942, 1.0755]$

Yerel ve Global En İyi Değerler

- Her parçacığın yerel en iyi pozisyonu ilk iterasyon olduğu için güncel pozisyonlarıdır.
- $\mathbf{LBF} = \mathbf{CF}(1) = [0.1976, -6.7378, -2.5061, 0.7942, 1.0755]$
- Global en iyi uygunluk değeri 1. iterasyon için $\mathbf{GBF}(1) = 1.0755$
- $\mathbf{LBP}(1) = \mathbf{CP}(1) = [-0.3425, 3.9558, -1.1228, -0.0981, 0.0385]$
- Her parçacığın yerel en iyi uygunluk değeri ilk iterasyon olduğu için güncel pozisyonlarıdır.
- En iyi global pozisyon, en iyi global uygunluk değerini elde ettiğimiz parçacığın pozisyonudur.
- 1. İterasyonun global en iyi pozisyonu $\mathbf{GBP}(1) = 0.0385$ olacaktır.

2.İterasyon Hız Değeri

$$V(i + 1) = W * V(i) + c_1 * r_1 * (LBP(i) - CP(i)) + c_2 * r_2 * (GBP(i) - CP(i))$$

İterasyon 1 den;

$$V(1) = [0.0319, 0.3185, 0.3331, 0.2677, -0.3292]$$

Her parçacık için v değeri yazılır;

Örn (0.0319 için);

$$r_1=0.0457, \quad r_2 = 0.5319, \quad CP(1) = -0.3425,$$

$$LBP(1) = -0.3425 \quad GBP(1) = 0.0385$$

$$V(2) = 0.7 * V(1) + 0.2 * r_1 * (LBP(i) - CP(i)) + 0.6 * r_2 * (GBP(i) - CP(i)) = 0.1439$$

Her V(1) değeri için hesaplanırsa;

$$V(2) = [0.1439, -1.7008, 0.8136, 0.2503, -0.2304] \text{ olarak bulunur.}$$

Anlık Pozisyon ve Uygunluk Değeri

Anlık pozisyon; $CP(i + 1) = CP(i) + V(i + 1)$

$CP(1) = [-0.3425, 3.9558, -1.1228, -0.0981, 0.0385]$,

$V(2) = [0.1439, -1.7008, 0.8136, 0.2503, -0.2304]$ değerlerini biliyoruz;

Buradan $CP(2) = [-0.1986, 2.2550, -0.3092, 0.1522, -0.1919]$ olarak bulunur.

Uygunluk değeri; $CF(i) = f(CP(i)) = 1 + 2 * CP(i) - CP(i)^2$

Buradan $CF(2) = [0.5634, 0.4250, 0.2860, 1.2812, 0.5794]$ bulunur.

$LBF(1) = [0.1976, -6.7378, -2.5061, 0.7942, 1.0755]$ Biliniyor. Anlık uygunluk değerlerine bakılır ve $LBF(1)$ den yüksek olanlar güncellenir.

$LBF(2) = [0.5634, 0.4250, 0.2860, 1.2812, 1.0755]$ elde edilir.

Yerel ve Global En İyi Değerler

$$\mathbf{CP}(2) = [-0.1986, 2.2550, -0.3092, 0.1522, -0.1919]$$

$$\mathbf{LBF}(2) = [0.5634, 0.4250, 0.2860, 1.2812, 1.0755]$$

En iyi global uygunluk değeri $\mathbf{GBF}(2) = 1.2812$

En iyi global pozisyon, en iyi global uygunluk değerini elde ettiğimiz parçacığın pozisyonudur. Bu nedenle:

$\mathbf{GBP}(2) = 0.1522$ olacaktır. (4. parçacığın $\mathbf{CP}(2)$ pozisyonu)

En iyi yerel pozisyonlar için en iyi yerel uygunluk değeri ile anlık uygunluk değeri karşılaştırılır ve yüksek olanın anlık pozisyonu seçilir.

$$\mathbf{CF}(2) = [0.5634, 0.4250, 0.2860, 1.2812, 0.5794]$$

$$\mathbf{LBF}(1) = [0.1976, -6.7378, -2.5061, 0.7942, 1.0755]$$

İlk dört parçacık için CP(2) diğer 5. parçacık için CP(1) değeri kullanılır, dolayısıyla,

$$\mathbf{LBP}(1) = [-0.1986, 2.2550, -0.3092, 0.1522, 0.0385] \text{ olacaktır.}$$

1. Ve 2. İterasyon

İterasyon	V(i) & CP(i)	CF(i) & LBF(i)	GBF(i)	LBP(i) & GBP(i)
1	$V(1) = [0.0319, 0.3185, 0.3331, 0.2677, -0.3292]$ $CP(1) = [-0.3425, 3.9558, -1.1228, -0.0981, 0.0385]$	$CF(1) = [0.1976, -6.7378, -2.5061, 0.7942, 1.0755]$ $LBF(1) = [0.1976, -6.7378, -2.5061, 0.7942, 1.0755]$	$GBF(1) = 1.0755$	$LBP(1) = [-0.3425, 3.9558, -1.1228, -0.0981, 0.0385]$ $GBP(1) = [0.0385]$
2	$V(2) = [0.1439, -1.7008, 0.8136, 0.2503, -0.2304]$ $CP(2) = [-0.1986, 2.2550, 0.3092, 0.1522, -0.1919]$	$CF(2) = [0.5634, 0.4250, 0.2860, 1.2812, 0.5794]$ $LBF(2) = [0.5634, 0.4250, 2860, 1.2812, 1.0755]$	$GBF(2) = 1.2812$	$LBP(2) = [-0.1986, 2.2550, 0.3092, 0.1522, 0.0385]$ $GBP(2) = [0.1522]$

3. Ve 4. İterasyon

İterasyon	V(i) & CP(i)	CF(i) & LBF(i)	GBF(i)	LBP(i) & GBP(i)
3	$V(3) = [0.0212, -2.2232, 0.8001, 0.1752, -0.1120]$ $CP(3) = [0.0141, 0.0318, 0.4909, 0.3274, -0.2944]$	$CF(3) = [1.0279, 1.0625, 1.7410, 1.5464, 0.3246]$ $LBF(3) = [1.0279, 1.0625, 1.7410, 1.5464, 1.0755]$	$GBF(3) = 1.7410$	$LBP(1) = [0.0141, 0.0318, 0.4909, 0.3274, 0.0385]$ $GBP(1) = [0.4909]$
4	$V(4) = [0.3011, -1.3308, 0.5601, 0.1980, 0.0420]$ $CP(4) = [0.3125, -1.2990, 1.0510, 0.5254, -0.2523]$	$CF(4) = [1.5312, -3.2861, 1.9974, 1.7740, 0.4317]$ $LBF(4) = [1.5312, 1.0625, 1.9974, 1.7740, 1.0755]$	$GBF(4) = 1.9974$	$LBP(4) = [0.3152, 0.0318, 1.0510, 0.5254, 0.0385]$ $GBP(2) = [1.0510]$