



*T.C*

***SAKARYA ÜNİVERSİTESİ***

**BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

# **BSM 310 – YAPAY ZEKA**

**Grup üyeleri:**

**G1612.10058 - MUSTAFA KUŞOĞLU**

**G1712.10102 - İBRAHİM YILDIRIM**

**G1612.10068 - ENES YAVUZ**

**Sakarya**

**2020**

# YERÇEKİMSSEL ARAMA ALGORİTMASI

Yerçekimsel Arama Algoritması 2009 yılında Rashedi, Saryazdi, Nezamabadi-pour tarafından Newton' un yerçekimi ve hareket kanunlarından esinlenerek geliştirdikleri bir algoritmadır. Bir sezgisel optimizasyon algoritmasıdır. Bu algoritmada nesneler bir kütleler topluluğudur. Nesnelerin birbirleri ile etkileşimi Newton' un yerçekimi ve hareket kanunlarına göre gerçekleşir. Yapılan çalışmalarda Karınca Kolonisi ve Genetik algoritmalar gibi bazı algoritmalara göre daha iyi başarımlar gösterdiği görülmüştür.

## Yerçekimi Yasası

Yerçekimi kütlelerin birbirine doğru hızlanma eğilimidir. Evrendeki her parçacık diğer her parçacığı çeker. İki parçacık arasındaki yerçekimi kuvveti, kütlelerin ağırlığı doğru orantılı ve aralarındaki mesafe ile de ters orantılıdır.



$$F = G \times \frac{M_1 \times M_2}{R^2}$$

Newton'un ikinci yasasına göre ise bir parçacığa kuvvet uygulandığında ise parçacığın ivmesi sadece kuvvete ve kütesine bağlıdır.

$$a = F/M$$

Ayrıca teorik fizikte 3 farklı tür kütle tanımlanmıştır.

**Aktif Yerçekimi Kütle:** Bir cismin uyguladığı çekim kuvvetini ölçer.

**Pasif Yerçekimi Kütle:** Bilinen bir çekim alanındaki bir cisme uygulanan çekim kuvvetidir.

**Eylemsizlik Kuvveti:** Bir nesnenin bir kuvvet tarafından ivmelenmesine karşı gösterdiği dirençtir.

## Yerçekimsel Arama Algoritması

Yerçekimi yasasından etkilenecek şekilde yapılan bu algoritmada nesnelere ajan denir. Her ajanın kütlesi performanslarıyla doğru orantılıdır. Her nesne birbirini yerçekimi kuvveti birbirine çeker. Bu yerçekimi kuvveti ile nesneler birbirleri ile iletişim kurarak işbirliği yapmaları sağlanır. Daha iyi performansa sahip nesneler daha ağır ve yavaş olacağı için önemsiz nesnelerin algoritmayı istismar etmesi engellenir.

Her ajanın 4 özelliği vardır. Bunlar kütle, eylemsizlik kuvveti, aktif yerçekim ve pasif yerçekim kütesidir. Kütlelerin konumu problemin çözümüne karşılık gelir. Yerçekimi ve eylemsizlik kuvvetleri ise işlevsel fonksiyonlar olarak kullanılır.

Bu algoritmada çözümler ağırlıkları ile doğru orantılıdır. Algoritma ile diğer nesneleri kendine çeken en ağır nesneyi bulmaktır. En ağır nesne en iyi çözümdür.

- İlk olarak algoritmada kullanılacak olan yerçekimi sabiti(G), maksimum jenerasyon sayısı ve  $\epsilon$  sabiti belirlenir
- Şimdi i. boyutta N kadar ajan olan bir sistem düşünelim.

$$X_i = (x_{i,1}^1, x_{i,1}^d, \dots, x_{i,1}^n) \text{ ve } i = 1, 2, \dots, N$$

$x_{i,1}^d$  d. boyuttaki i ajanın konumu demektir.

- Rastgele olarak dağıtılmış nesnelerin uygunluk değerleri hesaplanır. En iyi ve en kötü değerler seçilir.

$$\text{best}(k) = \min (\text{fit}_j(k))$$

$$\text{worst}(k) = \max (\text{fit}_j(k))$$

$\text{fit}_j$  j. kütlenin k. jenerasyondaki uygunluk değeridir.

- Bir nesnenin aktif, pasif yerçekimsel kuvvetleri ile eylemsizlik kuvvetleri birbirine eşit olarak alınır. Buna göre de her nesnenin kütlesi hesaplanır.

$$M_{ai} = M_{pi} = M_{ii} = M_i \quad i = 1, 2, 3, \dots, N$$

$M_{ai}$  kütlenin aktif yerçekimsel kuvveti,  $M_{pi}$  kütlenin pasif yerçekimsel kuvveti,  $M_{ii}$  eylemsizlik kütlesi

$$m_i(k) = \frac{\text{fit}_i(k) - \text{worst}(k)}{\text{best}(k) - \text{worst}(k)}$$

$m_i(k)$  i. kütlenin k. jenerasyondaki kütlesinin değeridir.

$$M_i(k) = \frac{m_i(k)}{\sum_{j=1}^N m_j(k)}$$

$M_i$  eylemsizlik kütlesidir.

- Nesnelerin kütleleri belirlendikten sonra ise yerçekimi kuvveti yasasındaki formül ile iki nesne(kütle) arasındaki kuvvet hesaplanır.

(t) zamanında i ve j ajanları arasındaki yerçekimsel kuvvet formülü

$$F_{ij}^d(t) = G(t) \frac{M_{pi}(t) M_{aj}(t)}{R_{ij} + \epsilon} (x_j^d(t) - x_i^d(t))$$

Bu fonksiyonda  $M_{aj}$ , j ajanın aktif yerçekimsel kuvveti,  $M_{pi}$  ise i ajanın pasif yerçekimsel kuvvetidir.  $G(t)$  yerçekimi sabitidir ve  $R_{ij}$  de ajanlar arası mesafedir. Epsilon ise küçük bir sabittir.

- Her kütleye etki eden kuvvetler bulunur.

$$F_i^d(k) = \sum_{j \in k, \text{best}, j \neq i} \text{rand}_j F_{ij}^d(k)$$

$\text{rand}_j$  0 ile 1 arası rastgele bir sayı,  $K_{\text{best}}$  ise başlangıçta  $K_0$  kadar başlayan lineer olarak azalan bir değerdir.

- Kütlelerin ivmesi Newton' un yasasına dayanarak hesaplanır.

$$a_i^d(k) = \frac{F_i^d(k)}{M_{ii}(k)} \quad M_{ii} \text{ i. kütlenin eylemsizlik kütlesidir.}$$

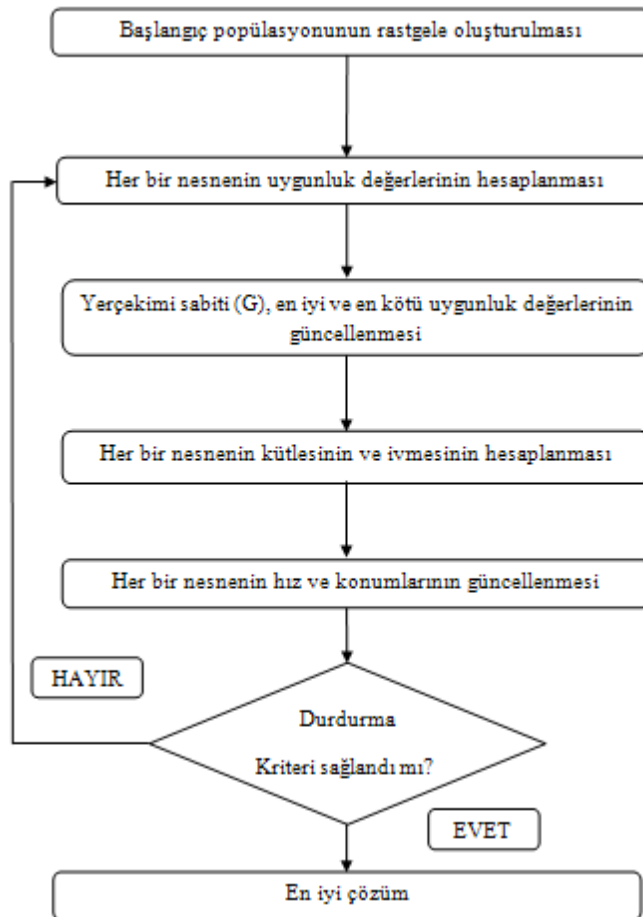
- İvmesi hesaplanan kütlelerin hızları ve konumları güncellenir.

$$V_i^d(k+1) = rand_j V_i^d(k) + a_i^d(k)$$

$$x_i^d(k+1) = x_i^d(k) + V_i^d(k+1)$$

- Max jenerasyon veya optimum sonuç oluşana kadar devam edilir. Bu işlemler ile en uygun kütle bulunmaya çalışılır.

## Yerçekimsel Arama Algoritması Akış Diyagramı



## Yerçekimsel Arama Algoritması Avantajları

- Doğrusal olmayan optimizasyon problemlerini çözmek için uygun sürü tabanlı algoritmadır.
- Daha az hesaplama süresi gerektirir
- Etkileşimli olarak çalıştırılabilir, kullanıcı tarafından önerilen çözümlere uyum sağlayabilir.
- Yüksek hassasiyetli sonuçlar verir.
- Problemlere uygulanması kolaydır.

### **Yerçekimsel Arama Algoritması Dezavantajları**

- Daha az hesaplama süresi gerektirir
- Etkileşimli olarak çalıştırılabilir, kullanıcı tarafından önerilen çözümlere uyum sağlayabilir.
- Yüksek hassasiyetli sonuçlar verir
- Problemlere uygulanması kolaydır