**GSA**

Yerçekimsel arama algoritması (YAA), Rashedi ve arkadaşları (2009) tarafından bulunan Newton’un hareket ve yerçekimi kanunlarına dayalı bir optimizasyon algoritmasıdır. Bu nedenle sezgisel optimizasyon yöntemlerinden biri olan YAA’yı ele almadan önce Newton’un hareket ve yerçekimi kanunlarına yer verilmelidir.

1. **Newton’un Hareket Kanunları**
   1. **Eylemsizlik Yasası** **=>** Evrendeki herhangi bir cisim üzerine hiçbir kuvvet etki etmiyorsa ya da cisme etki eden kuvvetlerin vektörel olarak toplamı sıfıra eşitse, eğer cisim belli bir hızda hareket ediyorsa düzgün doğrusal hareketine devam eder, duruyorsa durağanlığını korur. Dolayısıyla bu yasada cismin ivmesinin sıfır olması söz konusudur
   2. **İvme Yasası =>**  Bir cisim üzerine etki eden kuvvetlerin vektörel olarak toplamı sıfıra eşit değilse yani cisim herhangi bir kuvvet yönünde çekiliyorsa, cisimde bir hız değişimi olur. Bu hız değişimine ivme adı verilir. Bir cismin ivmesi, üzerine uygulanan kuvvet ile doğru, kütlesi ile ters orantılıdır. F = m\*a
   3. **Etki – Tepki Yasası =>** İki cisim arasında herhangi bir kuvvetle oluşan etki her zaman kendisine eşit büyüklükte, aynı doğrultuda fakat ters yönde bir tepkiye neden olur
2. **Newton’un Evrensel Çekim Kanunu**
   1. Evrendeki cisimler birbirlerini yerçekimine bağlı olarak çekerler. Bu kütle çekim kuvveti, cisimlerin kütlelerinin çarpımıyla doğru, cisimler arasındaki uzaklıkla ters orantılıdır. F = G\*( (M1\*M2) / (R^2) ). G yer çekim sabitidir.

YAA, Newton’un hareket kanunlarından ikincisi olan ivme kanunu ve yine Newton’un evrensel çekim kanunu temel alınarak tasarlanmış bir optimizasyon yöntemidir. YAA’da arama uzayındaki her bir parçacık bir kütle olarak kabul edilir. Bu nedenle YAA’yı bir yapay kütle sistemi olarak ifade etmek mümkündür. Arama uzayındaki tüm kütleler Newton’un evrensel çekim kanununa göre birbirlerini çekerler ve yerçekimi kuvveti ile birbirlerine kuvvet uygularlar. Bu kuvvete maruz kalan kütleler arama uzayı içerisinde hareket ederek optimum çözüme ulaşırlar.

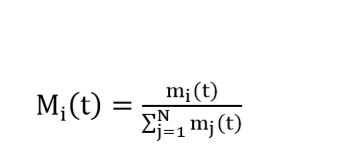
Kütleler, üzerine herhangi bir kuvvet etki ettiğinde, bu kuvvete karşı koymak isteyerek bir direnç gösterir. Bu dirence eylemsizlik kütlesi denir. Eylemsizlik kütlesi fazla olan kütle daha yavaş hareket etmek isteyecek, az olanı ise daha hızlı hareket edecektir.

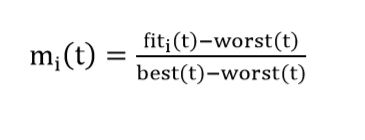
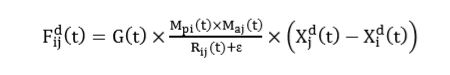
Algoritma boyunca en ağır olan kütle diğer kütlelere nazaran daha yavaş hareket edecek ve diğerlerini kendine çekecektir. İterasyon sayısı bitiminde ya da herhangi bir sonlandırma eylemi olduğunda kütlesi en fazla olan nesne, problemin optimum çözümünü oluşturmuş olacaktır.

**Algoritmada Hesaplanan Parametreler**

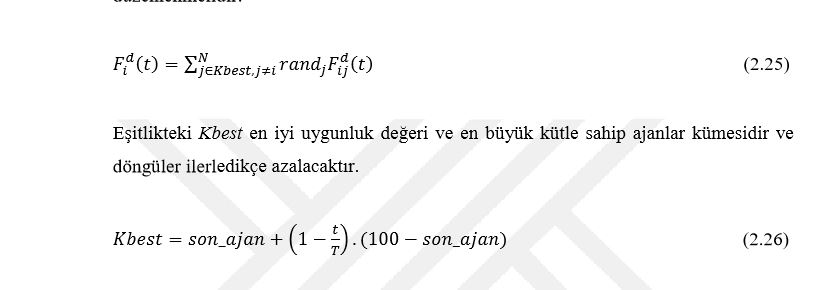
****

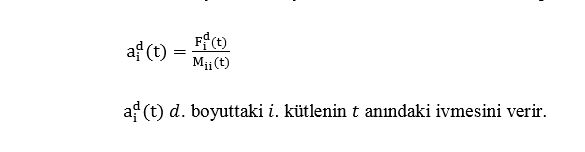
1. **Yerçekimi Sabiti =>** Algoritmanın ilk kısmında yerçekimi sabitine verilen değerin her iterasyonda azaltılması gerekmektedir. Zamanla azalan yerçekimi sabiti ile arama hızı kontrol edilmektedir. T anındaki yerçekimi sabiti şekildeki eşitlik ile hesaplanır. Burada G, yerçekimi sabitinin başlangıç değerini, α kullanıcının belirlediği sabit bir değer, t o andaki iterasyon sayısı ve T ise maksimum iterasyon sayısını gösterir.

****

1. **Kütle hesabı =>** Kütle hesabı için arama uzayında bulunan bir kütlenin aktif yerçekimsel kütlesi, pasif yerçekimsel kütlesi ve eylemsizlik kütlesi birbirlerine eşit alınarak, tüm kütleler hesaplanır. Kütlenin aktif yerçekimsel kütlesi Mai, pasif yerçekimsel kütlesi Mpi, eylemsizlik kütlesi Mi olmak üzere GSA’da bu kütleler eşit alınır.
2. **Kuvvet Hesabı =>** Kuvvet hesabı verilen şekildeki formül ile hesaplanır. T anında, £ kullanıcı tarafından karar verilen sayısal olarak ufak bir sabiti, G yerçekimi sabitini, Mpi i. Kütlenin pasif yerçekimsel kütlesini, Maj j. Kütlenin aktif yerçekimsel kütlesini temsil etmektedir. Xj – Xi ifadesi ilgili boyuttaki pozisyon vektörünü göstermektedir ve standart YAA da eklendiği görülmektedir. Bunun sebebi olarak da her boyuttaki ajanların birbirlerine yapmış oldukları kuvveti boyut içerisindeki mesafelerini de katarak boyut içerisinde kuvvetin tesirini daha iyi ortaya koymaktır. Rij ise i ve j kütlelerin oklit mesafesini temsil etmektedir.

İki kütle arasındaki kuvvet hesaplandıktan sonra bir kütleye etkiyen toplam kuvvet verilen şekildeki formül ile hesaplanır.



1. **İvme Hesabı =>** 
2. **Hız Hesabı ve Konum Güncelleme =>** İvmenin hesaplanmasıyla birlikte elde edilecek yeni hız V = rand\*V + a formülü ile hesaplanır. Rand 0-1 arasında rast gele bir sayıyı temsil etmektedir. Bulunan yeni hız değeriyle birlikte X = X + V formülü ile konumlar güncellenmektedir.

**DSK İnceleme**

1. **Dağılım –** popülasyonun oluşturulması random ile yapılmıştır
2. **Seçim :** Algoritmada oluşturulan popülasyon üzerinde işlemler yapılarak en uygun değer bulunmaya çalışılmaktadır. Algoritmanın en önemli adımlarından birisi olan kuvvet hesabında, popüşlasyonda seçim yapılırken formülü verilmiş olan Kbest formülü uygulanır. Hesaplanan Kbest değeri kadar iterasyon yapılarak, popülasyon bireyleri üzerlerinde uygulanan toplam kuvvet hesaplanmış olur. Uygunluk değeri düşük olan ajanların ağırlıkları az olduklarından dolayı ivmeleri yüksek olacaktır. İvmesi yüksek olan ajanların hareketleri daha büyük olup yerel min max tan kaçınılmayı sağlayan çeşitlilik adımı sağlanmış olur.
3. **Kontrol** Ajanlara uygulanan kuvvetin hesaplanması için kullanılan formülde yer alan G parametresi, iterasyon ilerledikçe orantısal olarak düşürülmektedir. A = F/M formülünde F kuvveti G parametresi ile doğru olantılı olduğundan dolayı, iterasyon sonlarına doğru ajanlar daha yavaş hareket ederek sömürü kabiliyeti artmaktadır. Ajanların yer güncellemesi için V parametresi kullanılmaktadır. V parametresi güncellenirken, eski v değerinin etkisini kontrol etmek için başına 0-1 arasında üretilen random sayı ile çarpılmıştır. Bunun sonuçunda ajanların sürekli çeşitlilik yapmasının önüne geçilerek (0 değeri üretilmesi durumunda ajan sadece ivme miktarı kadar hareket edecektir) komşuluk araması yapmasıda sağlanmıştır.
4. **Kontrol** G parametresinin hesaplanmasında rastgeleliliği artırmak için KAOS haritaları kullanılarak algoritmanın performansı geliştirilmiştir. Bu fark 20Value1500Iterasyon.JPG ve 230Value1500Iterasyon.JPG resimlerinde görülmektedir.