**Kümeleme Problemi**

Iris veri setindeki değerler kullanılarak iris çiçeği 3 farklı kümeye ayrıldı. Bu işlem PSO algoritması kullanılarak yapıldı. Iris veri setinde 4 nitelik olduğundan ve bu veri setini 3 kümeye ayırma işleminden dolayı PSO algoritmasına uyarlama esnasında 12 boyut olarak belirlendi (Şekil 1). Yazılan fonksiyona her bir aşamada bu 12 değer gönderildi. Fonksiyonda alınan bu 12 değer 4 nitelik olduğundan dolayı 4 erli gruplar şeklinde işleme alındı. Bu 3 gruptaki değerler ile veri setindeki değerler arasındaki uzaklıklar hesap edilerek en kısa mesafe bulundu (Şekil 2). Daha sonra bu uzaklıklar toplanılarak şu an ki parçacık için maliyet hesabı yapıldı. Fonksiyonda üretilen bu maliyetin değeri ve pozisyon bilgileri geriye dönüldü. Bu işlem her üretilen parçacık için tekrar edilip en az maliyetin bulunması sağlandı.

metin, yazı tipi, ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldumetin, ekran görüntüsü, yazı tipi, çizgi içeren bir resim

Açıklama otomatik olarak oluşturulduŞekil 1metin, ekran görüntüsü, yazı tipi, grafik içeren bir resim

Açıklama otomatik olarak oluşturuldu

**Şekil 3**

**Şekil 2**

**Şekil 1**

**PSO Benzetme Adımları**

Yukarıda da bahsedildiği üzere veri setindeki veri sayısı bulundu ve popülasyon büyüklüğü 100 olarak belirlendi. Veri setindeki her bir değer için parçacık üretileceği için boyut belirlenmesi gerekliydi. 3 kümeye ayırma işlemi olduğundan ve her kümede 4 nitelik bulunacağından dolayı her bir parçacığın boyutu 12 olmalı. Arama uzayı Iris veri setindeki değerleri içinde barındırabileceği max, min değerler olarak 0 ve 10 aralığı seçildi.

**Algoritmanın Çözümü**

Algoritma her bir veri için üretilen parçacıkları yazılan fonksiyonda birer birer her bir veri için deneyerek minimum uzaklıklarını bulup maliyetlerini ve bu maliyetin hangi değerler ile sağlandığı bilgisini PSO’ya gönderip PSO üzerinde max iterasyon kadar denedikten sonra global min her iterasyonda güncelleyip en verimli sonucu ve bu sonucun değerlerini geri döndürerek çözümü sağlamaktadır. Sonlandırma kriteri sağlandığında da elde edilen global en iyi çözüme göre veri seti kümelenmiş olur. Hangi verinin hangi kümede olduğu bilgisi main dosyası içerisindeki clusters değişkeninde tutuldu.

* metin, ekran görüntüsü, yazılım, sayı, numara içeren bir resim

  Açıklama otomatik olarak oluşturuldu5000 iterasyonda 0 ve -10 uzay aralığında en yakın mesafe 349.9199 olarak ölçüldü. Bu sonucu en başlarda buldu yani local minimuma takıldı.

**Şekil 4**

* metin, ekran görüntüsü, sayı, numara, yazı tipi içeren bir resim

  Açıklama otomatik olarak oluşturuldumetin, ekran görüntüsü, yazı tipi, sayı, numara içeren bir resim

  Açıklama otomatik olarak oluşturulduLocal minimuma takılmayı önlemek için PSO algoritmasında bulunan hız vektörünün alt ve üst sınırını aştığı durumlarda max ve min değerlerine eşitleyen kodu kaldırarak PSO’nun daha esnek çalışması sağlanıldı(Şekil 5). Bu sayede local minimuma takılması önlenmiş oldu. 0 ve 10 uzay aralığında en yakın mesafe ilk iterasyonda 324.6064, 5000. iterasyonda ise 131.1668 olarak bulundu.

**Şekil 6**

**Şekil 5**