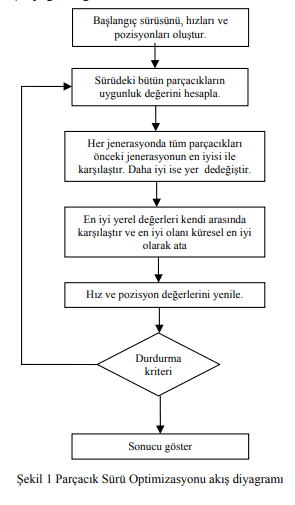
Parçacık Sürü Optimazasyonu ile Araç Rotalama Problemi

# Giriş

**Araç Rotalama Problemi (ARP),**bir veya birkaç depodan müşterilere hizmet götürecek araçlar için en uygun rotaları belirlemeyi amaçlayan bir kombinatoryal eniyileme (optimizasyon) problemidir. NP-Zor sınıfında yer alır.

Parçacık Sürü Optimizasyonu (PSO), sürü halinde hareket eden balıklar ve böceklerden esinlenerek Kenedy ve Eberhart (1995) tarafından geliştirilmiş bir optimizasyon yöntemidir.Temel olarak sürü zekâsına dayanan bir algoritmadır. Sürü halinde hareket eden hayvanların yiyecek ve güvenlik gibi durumlarda, çoğu zaman rasgele sergiledikleri hareketlerin, amaçlarına daha kolay ulaşmalarını sağladığı görülmüştür. PSO bireyler arasındaki sosyal bilgi paylaşımını esas alır. 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 bireylerin pozisyonunun, sürünün en iyi pozisyona sahip olan bireyine yaklaştırılmasına dayanır. Bu yaklaşma hızı rasgele gelişen durumdur ve çoğu zaman sürü içinde bulunan bireyler yeni hareketlerinde bir önceki konumdan daha iyi konuma gelirler ve bu süreç hedefe ulaşıncaya kadar devam eder. (Kennedy & Eberhart, 2007)

# Genel Olarak PSO



Algoritma temel olarak aşağıdaki basamaklardan

oluşur;

* Rasgele üretilen başlangıç pozisyonları ve hızları ile başlangıç sürüsü oluşturulur.
* Sürü içerisindeki tüm parçacıkların uygunluk değerleri hesaplanır.
* Her bir parçacık için mevcut jenerasyondan yerel en iyi (pbest) bulunur. Sürü içerisinde en iyilerin sayısı parçacık sayısı kadardır.
* Mevcut jenerasyondaki yerel eniyiler içerisinden küresel en iyi (gbest) seçilir.
* Pozisyon ve hızlar aşağıdaki gibi yenilenir.

# Projede PSO – ARP-Süpürme Algoritması

Proje ilk olarak müşterileri ve araçların gerekli bilgilerini kullanıcıdan alır

**Burada dikkat edilmesi gereken nokta Müşteriler kısmına ilk eklenecek olan müşteri depodur. Tüm parametre değerleri {0,0,0,0,0} olarak yazılır.  
(x ve y koordinatlarının sıfır olması açı hesabında kolaylık sağlaması için).  
Rasgele oluşturma kısmına Müşteri sayısı veya araç sayısı yazılırsa tüm müşteriler ve araçlar belirlenen parametreler arasında rastgele değerlere sahip olarak oluşturulur.**

Veriler MSSQL veri tabanına kaydedilir.   
  
Oluşturulan nesnelere veriler aktarılır ve listeler oluşturulur.  
  
İşlem gerçekleştirilmeden önce Müşterilerin depoya olan açısı hesaplanır ve bireyin müşterileri adlı klonlanmış listede müşteriler açılarına göre küçükten büyüğe sıralı halde listelenir.

Bütün müşterilerin birbirlerine olan uzaklığı hesaplanır.

Parçacıklar oluşturulur. Depo hariç müşteri sayısı kadar. (dilenirse manuel değer verilebilir)  
Parçacıkların her biri , açısı düşükten büyüğe olan sıralı müşterilere gider. Her parçacık birbirinden farklı müşterilerden başlayarak süpürme algoritması ile kendi çözümlerini elde eder.  
Bu çözümler sıralı olarak üretilir rastgele değil. O yüzden bazı işlemlerde farklılaştırmalar söz konusudur.

İlk parçacık çözüm (0\*1\*2\*0----0\*3\*4\*5\*0---0\*6\*) ise  
İkinci parçacık için çözüm(0\*2\*3\*4\*0----0\*5\*6\*1\*0) olabilir ve bu böyle devam eder. İlk müşteri her parçacık döngüsünde son müşteri olur. Bu sayede her parçacığın başlangıç müşterisi farklı oldu.

Çözümlerde elde edilen sonuç rota uzunluğudur.  
Pozisyon değerleri bir matris içinde tutulur burada dikkat edilmesi gereken nokta pozisyon hesabı (Hai Shen, 2009) yayımladıkları makaleden uyarlanmıştır.

Kısaca [3.1 , 5.8 , 0.1 , 0.3 ........] gibi değerin tam sayı kısmı araç ID si ve diğer reel sayı kısmı araçların gittiği müşteri sırasını gösterir.  
  
3. İD ye sahip Aracın ilk müşterisi 1.ID ye sahip müşteri  
5. İD ye sahip aracın 8. Müşterisi 2.İD ye sahip müşteri gibi  
0. İD ye sahip aracın 8. Müşterisi 1.İD ye sahip müşteri  
  
  
Hız hesabı klasik PSO hız hesabına göre yapılmıştır. Bu projede Atelet ağırlığı, Pozisyon ağırlıkları ve Minumum maxmium ağırlar sabit olarak alınmış.   
  
Öğrenme(ivme) katsayıları ise [0,1] düzgün dağılımında her iterasyonda rastgele üretilmiştir.

Mevcut pozisyonlara hızlar eklendikten sonra elde edilen pozisyon değerlerine göre Araç Listesine Swap ve İnsert Operatörleri uygulandı. Bu sayede araçların sırlaması değişti. Pozisyon büyüklüğüne göre swap ve insert büyüklüğü(azlığı çokluğu) değişir.  
  
Burada müşterilerin yerlerini değiştirmemin sebebi bütün çözümleri süpürme algoritması ile elde edildiği için olası optimal çözüme uygun rotaların oluşturulmuş olmasından dolayı.  
  
Hız ve pozisyona ekleme kısmına kadar algoritma tatmin edici fakat. Bunun sonucunda yapılan swap-insert işlemlerinin mantığı biraz zorlama ile yapıldı. Çünkü gerekli yöntemleri bu konuda araştırmalarım neticesinde bulamadım. Ben de elde edilen pozisyon değerlerine göre araçlar üzerinden swap ve insert yaptım.  
  
Bu arada en fazla 100 müşteriyi 50 iterasyona kadar çözebildim CPU yetersiz geliyor. Sanırım düzgün kodlama yapamadığım için fazla işlem yapmak zorunda kaldım.