CSA için uygulanan yöntemlerin incelenmesi

* Algoritma konum güncellerken 2 aşama yapıyor. Rast gele bir ajan seçiliyor. Bu ajan takip edildiğini anlar ise onu takip edeni rast gele bir noktaya götürüyor. Anlamaz ise PSO mantığı ile takip eden kuşun konumu, o kuşa göre güncelleniyor. Hafızada her zaman en iyi fit değerleri saklanıyor. Yeni konum eğer hafızadan daha iyi ise hafıza guncelleniyor. Konum güncellemede hafızaya göre yapılıyor. xnew(i,:)= x(i,:)+fl\*rand\*(mem(num(i),:)-x(i,:)); Mem hafızadaki konum. x(i,:) ise şuanda olduğu konum.
  + Mem(num(i)) ifadesi random alınıyor. Bunun yerine fdb ile seçilen alınıyor.
    - Bu seçim iterasyon boyunca hep uygulandı ve %50 olarak uygulandı(tekse yap). %50 olan daha iyi sonuç verdi. %50 içinde 3 farklı senaryo var. İlk son ve direk %50 şeklinde. Direk %50 daha iyi sonuç verdi. Sonra oranı %70 50 25 30 10 gibi sayılarla denediler ve farklı oranlar seçildi problemler için. Bu sonucu iyileştirmek için takip etmeyi anlayan parametreyi 0.1’den 0.5’e çektiler. En son burada bitirildi ve oran stabil olarak %10’da kaldı.
    - Aslında mantık güzel. Komşuluk ve çeşitlilikte mutasyon işlemi yapılmış; FDB ile seçilen ajanın değişkenlerinin %30’u rastgele sayılar ile değiştirilmiş. Başarı elde edilememiş ama bu seçenek bana iyi geldi.

EFO için uygulanan yöntemlerin incelenmesi

1. Yeni\_Çözüm\_Adayı\_Noktası => nört\_Alan\_Çözüm\_Aday\_Noktası + (altın\_Oran => 1.6180\*R\* (pozitif\_Alan\_Çözüm\_Aday\_Noktası -nört\_Alan\_Çözüm\_Aday\_Noktası)) + R\*(nötr\_Alan\_Çözüm\_Aday\_Noktası - negatif\_Alan\_Çözüm\_Aday\_Noktası) formülü kullanır. Burada nört pozitif negatif çözüm adaylarını alınması rast gele olur. Buna müdahale yapıldı
   1. Pozitif ve negatif çözüm adaylarının seçimleri ayrı ayrı fdb ile yapıldı. Yapılış şekli ise, tüm popülasyonu verip almak ve sadece negatif veya pozitifi vererek almak. Negatifte tüm popülasyonu verip gelen çözüm adayının kullanılması 100 boyutta çok iyi şeylere yol açtı.