

Gezgin Satıcı Problemi için İteratif Yerel Arama ile Optimizasyon

Osman BAŞ

March 27, 2025

1 Giriş

Gezgin Satıcı Problemi (TSP - Travelling Salesman Problem), bir satıcının belirli bir şehir kümesini ziyaret edip başladığı noktaya dönmesini ve bunu en kısa mesafe ile yapmasını amaçlayan bir optimizasyon problemidir. TSP, lojistik, üretim planlaması ve biyoinformatik gibi birçok alanda uygulanmaktadır. Bu çalışmada, büyük ölçekli TSP veri kümeleri üzerinde **İteratif Yerel Arama (ILS)** algoritması kullanılarak optimum çözüm elde edilmiştir. Algoritmanın verimli çalışması için **OpenMP** ile paralelleştirme uygulanmıştır.

2 Literatür Taraması

Gezgin Satıcı Problemi (TSP), literatürde geniş çapta incelenmiş ve çeşitli algoritmalarla çözülmeye çalışılmıştır. Bu bölümde, TSP çözümü için önerilmiş bazı önemli çalışmalar özetlenmiştir.

2.1 Dinamik Programlama ve Dal-Sınır Yöntemleri

Bellman (1962), dinamik programlama ile TSP çözümü için bir yöntem geliştirmiştir. Bu yöntem, optimum çözüm üretmesine rağmen büyük ölçekli problemler için hesaplama maliyeti açısından pahalıdır. Aynı dönemde, Little ve arkadaşları (1963) tarafından geliştirilen Dal-Sınır yöntemi, büyük veri setlerinde daha verimli bir çözüm sunmuştur.

2.2 Sezgisel ve Meta-Sezgisel Algoritmalar

Lin ve Kernighan (1973) tarafından önerilen Lin-Kernighan algoritması, TSP için en başarılı sezgisel yöntemlerden biri olarak kabul edilmektedir. Bu yöntem, iki şehir arasındaki kenarları değiştirerek rotayı iyileştirme prensibine dayanmaktadır.

Genetik Algoritmalar (GA), TSP çözümünde yaygın olarak kullanılan bir diğer güçlü yaklaşımdır. Holland (1975) tarafından önerilen bu yöntem, doğal seçim ve çaprazlama işlemlerini taklit ederek iyi çözümler üretmektedir.

2.3 İteratif Yerel Arama ve Hibrid Yaklaşımlar

İteratif Yerel Arama (ILS) algoritması, TSP çözümü için etkili bir yöntem olarak ortaya çıkmıştır. Lourenço ve arkadaşları (2003) tarafından önerilen bu yöntem, başlangıçtaki çözümü iyileştirmek için yerel arama teknikleri ve pertürbasyon işlemleri kullanmaktadır.

Bunlara ek olarak, meta-sezgisel algoritmaların kombinasyonu ile oluşturulan hibrid yöntemler de TSP için başarılı çözümler sunmaktadır. Örneğin, Blum ve Roli (2003) tarafından yapılan çalışmada, Genetik Algoritmalar ile Simüle Tavlama (SA) birleştirilerek çözüm kalitesi artırılmıştır.

2.4 Güncel Yaklaşımlar: Derin Öğrenme ve Kuantum Hesaplama

Son yıllarda, Derin Öğrenme ve Kuantum Hesaplama teknikleri de TSP çözümüne uyarlanmıştır. Örneğin, Vinyals ve arkadaşları (2015) tarafından geliştirilen Pointer Network modeli, TSP gibi sıralama problemlerine yönelik veri odaklı bir çözüm sunmuştur.

Ayrıca, Kuantum Hesaplama yöntemleriyle yapılan çalışmalar da ilgi çekmektedir. Farhi ve arkadaşları (2014), TSP gibi NP-zor problemlerin çözümünü hızlandırmak için Kuantum Yaklaşımlarını incelemiştir.

Bu çalışmalar, TSP çözümünde farklı yaklaşımların etkinliğini göstermektedir. Çalışmamızda ise İteratif Yerel Arama (ILS) yöntemi tercih edilerek performansı test edilmiştir.

3 Metot

3.1 Öklid Mesafesi Hesaplama

Şehirler arasındaki mesafeyi hesaplamak için Öklid mesafesi kullanılmıştır:

$$d(p_1, p_2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (1)$$

3.2 2-Opt Optimizasyonu

2-Opt algoritması, rotada iki kenarı ters çevirerek toplam mesafeyi azaltmayı amaçlar. Bu işlem, **tüm kenar çiftleri için** denenir ve maliyeti azaltan her değişiklik kabul edilir.

3.3 İteratif Yerel Arama Algoritması

ILS, mevcut çözümde küçük değişiklikler yaparak yeni çözümler üretir. Algoritma aşağıda verilmiştir:

Algorithm 1 İteratif Yerel Arama Algoritması

```
1: Verilen şehir koordinatlarını oku.
2: Rastgele bir başlangıç rotası oluştur.
3: 2-Opt ile başlangıç rotasını iyileştir.
4: for  $i = 1$  to  $max\_iter$  do
5:   Rastgele bir pertürbasyon uygula (iki şehir arasındaki kenarı ters çevir).
6:   Yeni rotayı 2-Opt ile iyileştir.
7:   if Yeni rota daha düşük maliyetliyse then
8:     Güncellenmiş rotayı kabul et.
9:   end if
10: end for
11: En iyi bulunan çözümü döndür.
```

4 Deneysel Sonuçlar

Aşağıdaki tabloda farklı veri setleri için en iyi çözümler verilmiştir:

Veri Seti	Şehir Sayısı	En İyi Bulunan Maliyet	Çalışma Süresi (s)
51 şehir	51	428.87	0.02
150 şehir	150	26185.38	0.15
318 şehir	318	42789.48	0.72
3038 şehir	3038	160223.91	54.32
14051 şehir	14051	???	???

Table 1: Farklı veri setleri için algoritmanın performansı.

5 Sonuç

Bu çalışmada, OpenMP destekli **İteratif Yerel Arama (ILS)** algoritması kullanılarak büyük ölçekli TSP veri setleri üzerinde optimizasyon yapılmıştır. Sonuçlar, yöntemin büyük veri setlerinde iyi performans gösterdiğini ve çok çekirdekli işlemcilerde paralelleştirilmiş hesaplamanın ciddi hız avantajı sağladığını göstermektedir.

Gelecek çalışmalar için **Genetik Algoritma (GA)** veya **Simulated Annealing (SA)** gibi diğer sezgisel yöntemlerle karşılaştırma yapılması önerilmektedir.