

Gezgin Satıcı Problemi için Sezgisel Yaklaşımlar ve Performans Analizi

Alperen Mengünoğul

Yazılım Mühendisliği, Celal Bayar Üniversitesi, Türkiye.

Abstract

Bu çalışma, Gezgin Satıcı Problemi (TSP) için çeşitli sezgisel algoritmaların Python programlama dili ile uygulanmasını ve büyük ölçekli veri kümeleri üzerindeki performanslarını sunmaktadır. Nearest Neighbor, 2-Opt ve Simulated Annealing algoritmaları farklı veri boyutlarında test edilmiş ve sonuçlar karşılaştırmalı olarak değerlendirilmiştir.

1 Giriş

Gezgin Satıcı Problemi (TSP), bir satıcının belirli sayıda şehri yalnızca bir kez ziyaret edip başlangıç noktasına döneceği en kısa rotayı bulmayı amaçlayan kombinatoriyel optimizasyon problemidir. NP-zor sınıfında yer alan bu problem, lojistik, robotik, mikroçip üretimi ve biyoinformatik gibi pek çok gerçek hayat uygulamasında önemli yer tutmaktadır. Bu çalışmada, TSP'ye yönelik sezgisel algoritmalar kullanılarak yaklaşık çözümler üretilmiş ve bu algoritmaların farklı büyüklükteki veri kümeleri üzerindeki başarımları analiz edilmiştir.

2 Literatür Taraması

Gezgin Satıcı Problemi üzerine yapılan çalışmalar literatürde geniş yer tutmaktadır. Bu bölümde TSP'ye yönelik bazı önemli çalışmalar özetlenmiştir:

- **Lin ve Kernighan (1973)**: TSP için geliştirdikleri etkili sezgisel algoritma ile birçok klasik problemi başarılı şekilde çözmüşlerdir.
- **Kirkpatrick et al. (1983)**: Simulated Annealing yöntemi ile küresel optimizasyon problemlerinde kullanılabilecek genel bir çözüm yaklaşımı sunmuşlardır.

- **Gendreau et al. (1994)**: Tabu Search algoritmasını araç rotalama problemleri üzerine uyarlayarak TSP'ye dolaylı katkı sağlamışlardır.
- **Holland (1992)**: Genetik algoritmalar üzerine öncü çalışmaları ile evrimsel yöntemlerin TSP çözümünde nasıl kullanılabileceğini göstermiştir.
- **Applegate et al. (2006)**: TSP üzerine yürütülen en kapsamlı sayısal çalışmalar arasında yer almakta olup, Concorde TSP çözücüsünü geliştirmişlerdir.

3 Yöntem

Bu çalışmada üç farklı sezgisel algoritma kullanılmıştır: Nearest Neighbor, 2-Opt ve Simulated Annealing. Küçük boyutlu veri kümelerinde hızlı çözüm üretmesi nedeniyle Nearest Neighbor algoritması tercih edilmiştir. Ancak daha kaliteli çözümler elde etmek amacıyla 2-Opt ve Simulated Annealing algoritmaları da sırasıyla rotaları iyileştirmek üzere uygulanmıştır. Büyük boyutlu veri kümelerinde ise Simulated Annealing algoritmasının gelişmiş ve parametre ayarlı versiyonları kullanılmıştır. Tüm algoritmalar Python programlama dili ile geliştirilmiştir.

4 Deneysel Sonuçlar

Geliştirilen algoritmalar, farklı büyüklükteki beş veri seti üzerinde test edilmiştir: 51, 150, 318, 3038 ve 14051 şehirli dosyalar. Her biri için elde edilen yaklaşık optimal maliyet değerleri ve hesaplama süreleri aşağıdaki tabloda sunulmuştur.

Table 1 Farklı boyuttaki veri setleri için algoritma sonuçları

Dosya Boyutu	Yaklaşık Maliyet	Ortalama Süre (dakika)
51	433.05	≈ 3
150	27857.12	≈ 10
318	46821.27	≈ 18
3038	152102.45	≈ 38
14051	575715.88	≈ 68

Her bir örnek için elde edilen çözümler ayrıca path olarak da kaydedilmiş ve dosya formatına uygun şekilde çıktılandırılmıştır.

5 Sonuç

Bu çalışma, TSP için sezgisel algoritmaların uygulanabilirliğini ve etkinliğini ortaya koymuştur. Özellikle Simulated Annealing algoritması, orta ve büyük boyutlu veri setlerinde istikrarlı performans göstermiştir. Gelecekte bu algoritmaların paralel hesaplama teknikleri ve GPU destekli yaklaşımlar ile entegrasyonu planlanmaktadır. Ayrıca ikinci aşamada diğer çözüm yöntemleri ile karşılaştırmalı analizlerin yapılması hedeflenmektedir.