Algoritma Analizi ve Tasarımı Dersi

2023-2024 Bahar Dönemi

**ÖDEV # RAPORU**

Teslim Tarihi: 26.03.2024

Öğrenci Adı: İbrahim İNCE

Öğrenci Numarası: 212802050

* **Giriş**

Geliştirdiğim algoritmanın amacı verilen Data-Set 'teki veri ikililerini bir koordinat kabul edip, herhangi bir koordinattan yola çıkarak diğer tüm koordinatları en az maliyetle ve en hızlı gezmekti. Algoritma dosya içerisinde verilen koordinat değerlerini okur; optimal maliyeti, kod çalışırken geçen süreyi ve gezilen koordinatları sırasıyla ekrana yazdırır.

* **Geliştirme ve Çalıştırma Ortamları**

Bütün veriler için Jupyter Notebook ' da Python dilinde kodlama yaptım.

* **Algoritma Akışı**

distance(point1, point2): point1 ve point2 arasındaki öklid mesafesini hesapla

bu mesafeyi döndür

total\_distance(points, order):

total = 0

for i = 0 to len(order) - 2:

total = total + distance(points[order[i]], points[order[i + 1]])

total = total + distance(points[order[-1]], points[order[0]]) # Dönüş

total değeri döndür

read\_coordinates\_from\_file(filename):

points = boş liste

dosyayı aç filename 'r' olarak

size = dosyanın ilk satırını oku ve tamsayıya çevir

for \_ = 0 to size - 1:

x, y = dosyanın bir sonraki satırını virgülle ayır ve float'a çevir

points'e (x, y) ekle

dosyayı kapat

points değerini döndür

simulated\_annealing\_tsp(points,initial\_temp=1000,final\_temp=0.1, cool\_rate= 0.99):

n = points'un uzunluğu

current\_order = [0, 1, ..., n-1]'in rastgele bir karışımı

current\_distance = total\_distance(points, current\_order)

T = initial\_temp

while T > final\_temp:

new\_order = current\_order[:]

city1, city2 = rastgele iki farklı şehir seç

new\_order[city1], new\_order[city2] = new\_order[city2], new\_order[city1]

new\_distance = total\_distance(points, new\_order)

delta = new\_distance - current\_distance

if delta < 0 or rastgele sayı < exp(-delta / T):

current\_order = new\_order

current\_distance = new\_distance

T = T \* cool\_rate

current\_distance, current\_order değerlerini döndür

filename = 'tsp\_85900\_1.txt'

start\_time = zamanın başlangıcını al

points = read\_coordinates\_from\_file(filename)

optimal\_cost, optimal\_order = simulated\_annealing\_tsp(points)

end\_time = zamanın bitişini al

execution\_time = end\_time - start\_time

Optimal maliyet değerini, optimal\_order'ı ve execution\_time'ı yazdır

* **Sonuç**
* TCP güvenilir veri iletişimi kurmak için bilgisayar ağlarında kullanılan bir iletişim protokolüdür. Temel amacı iletişm sırasında kullanılan veri paketlerinin kaybolmaması, bozulmaması, belli bir sırada iletilmesi veri akışı kontorülü yapmaktır. Genellikle güvenilir veri iletişimi gerektiren uygulamalarda tercih edilir.