

# Ödev 1 – Rastgele Noktalarda Temel TSP

Ders: Bilgisayar Oyunlarında Yapay Zeka

Öğrenci: Mine Büşra Hazer

Tarih: 20.10.2025

## Amaç

Bu ödevin amacı, **Gezgin Satıcı Problemi (TSP)** için temel kavramları uygulamaktır. Rastgele 2D noktalar oluşturulmuş, bu noktalar **NetworkX** kullanılarak tam bir grafik olarak modellenmiştir. TSP'ye yaklaşık çözüm bulmak için **En Yakın Komşu (Nearest Neighbor)** sezgisel algoritması uygulanmıştır.

## Yöntem

- Rastgele Nokta Üretimi:**
  - 50x50 alanda 5 nokta rastgele oluşturulmuştur.
  - random.seed(42) değeri kullanılarak her çalışmada aynı noktalar üretilmiştir.
- Graf Modeli:**
  - Her nokta bir düğüm olarak eklenmiş, tüm düğümler birbirine bağlanmıştır.
  - Kenar ağırlıkları Öklid mesafesi ile hesaplanmıştır.
- Sezgisel Algoritma (En Yakın Komşu):**
  - Başlangıç noktasından başlanır.
  - En yakın ziyaret edilmemiş komşu seçilerek ilerlenir.
  - Tüm noktalar gezildikten sonra başlangıç noktasına dönlür.
  - Bu yöntem hızlıdır ve küçük grafiklerde iyi sonuç verir.
- Görselleştirme:**
  - Pembe noktalar:** şehirler
  - Gri çizgiler:** tüm olası bağlantılar
  - Kırmızı çizgi:** bulunan tur

## Sonuç

Algoritma, tüm noktaları bir kez ziyaret eden kapalı bir tur üretmiştir. Yöntem basit ve hızlıdır, ancak her zaman en kısa rotayı garanti etmez. İlerleyen ödevlerde bu yöntem, daha gelişmiş yaklaşımlar ile karşılaştırma için temel oluşturacaktır.

## Kaynak Kod:

```
# =====  
  
# Assignment 1 - Basic TSP on Random Points  
  
# Objective: Practice graph abstraction, random instance generation, simple  
# heuristics, and visualization.  
# • Generate random points in a 2D area and represent them as a graph using  
# networkx.  
# • Implement at least one simple TSP heuristic (Nearest Neighbor or Greedy  
# Insertion).  
# • Visualize the generated points and the tour (matplotlib or networkx draw).  
# • Submit: Python script, generated graph screenshot, short explanation of  
# the heuristic.  
# =====  
  
import random  
import math  
import matplotlib.pyplot as plt  
import networkx as nx  
  
# 1. Parametreler  
NUM_POINTS = 5  
AREA_SIZE = 50  
  
# 2. Seed Değeri (Her zaman aynı rastgele diziyi üretir.)  
random.seed(42)  
  
# 3. Rastgele noktalar oluştur  
points = []  
for i in range(NUM_POINTS): # Noktaları (x, y) şeklinde saklıyorum  
    x = random.uniform(0, AREA_SIZE)  
    y = random.uniform(0, AREA_SIZE)  
    points.append((x,y))  
  
# 4. NetworkX grafiği oluşturma  
G = nx.Graph()  
  
# Düğümleri ekleme  
for i, p in enumerate(points):  
    G.add_node(i, pos=p) # Her düğümün (x,y) pozisyonu  
  
# Kenarları ekleme - her nokta diğer tüm noktalara bağlı  
for i in range(NUM_POINTS):  
    for j in range(i + 1, NUM_POINTS):  
        distance = math.dist(points[i], points[j]) # İki nokta arasındaki  
        # Öklid mesafesi  
        G.add_edge(i,j, weight=distance)
```

```

# 5. En Yakın Komşu Algoritması
def nearest_neighbor_tsp(G, start=0):
    visited = [start]
    current = start
    total_dist = 0

    while len(visited) < len(G.nodes):
        # Daha ziyaret edilmemiş komşular ve uzaklıkları
        neighbors = [(n, G[current][n]['weight']) for n in
G.neighbors(current) if n not in visited]
        # En kısa mesafeli komşuyu bulma
        next_node, dist = min(neighbors, key=lambda x: x[1])
        visited.append(next_node)
        total_dist += dist
        current = next_node

    # En son başlangıç noktasına geri dön
    total_dist += G[current][start]['weight']
    visited.append(start)

    return visited, total_dist

# 6. Algoritmayı çalıştırma
tour, total_distance = nearest_neighbor_tsp(G)
print("Tur Sırası:", tour)
print("Toplam Mesafe:", total_distance)

# 7. Görselleştirme
pos = nx.get_node_attributes(G, 'pos') # node pozisyonlarını al
plt.figure(figsize=(5,5))

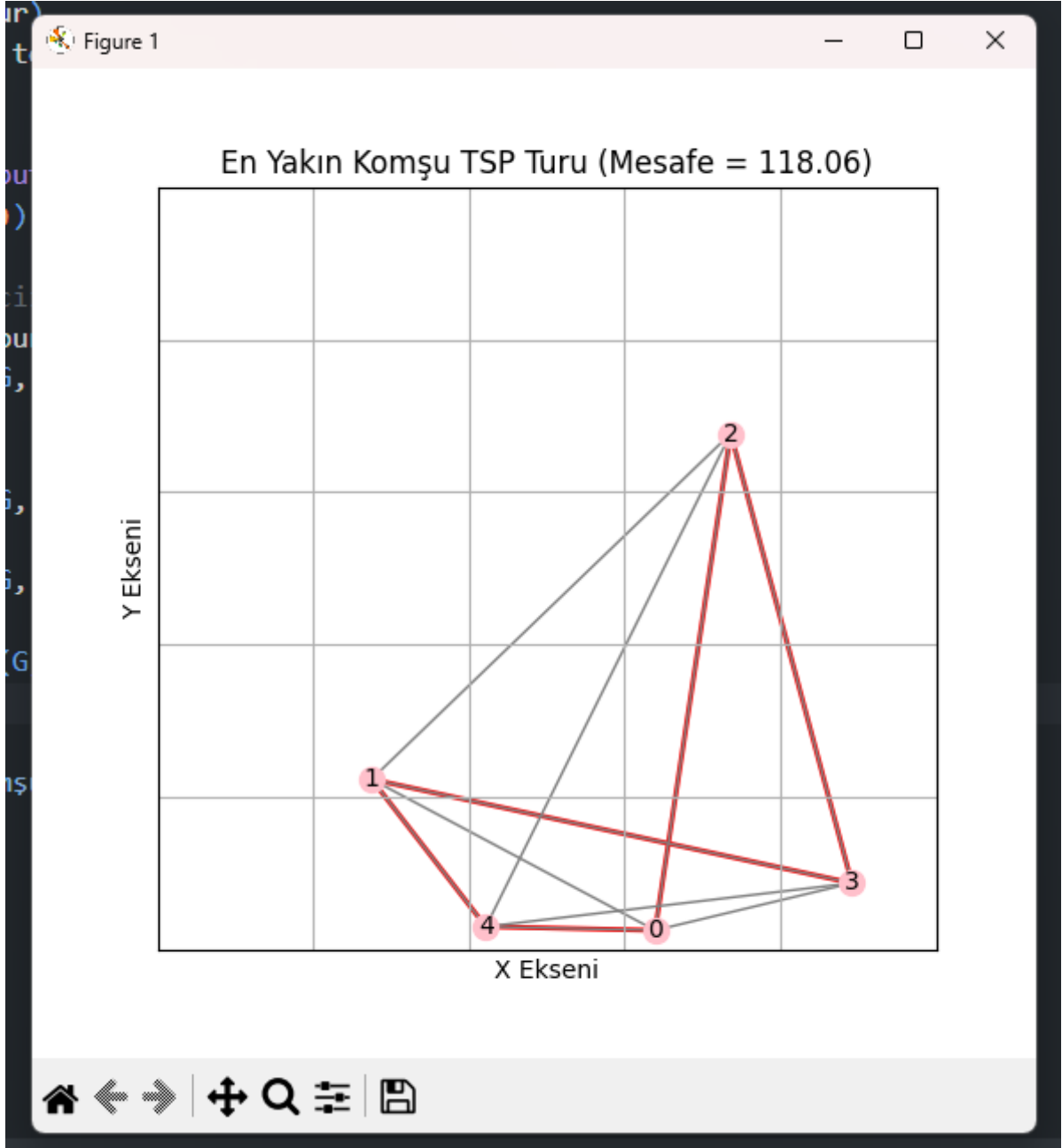
# Tur sırasını kırmızı çizgiyle göster
path_edges = list(zip(tour[:-1], tour[1:]))
nx.draw_networkx_edges(G, pos, edgelist=path_edges, edge_color='red', width=2)

# Düğümlerin çizimi
nx.draw_networkx_nodes(G, pos, node_color='pink', node_size=100)
# Kenarların çizimi
nx.draw_networkx_edges(G, pos, edge_color='gray')
# Node isimleri
nx.draw_networkx_labels(G, pos, font_color='black', font_size=10)

# 8. Grafik ayarları
plt.title(f"En Yakın Komşu TSP Turu (Mesafe = {total_distance:.2f})")
plt.xlim(0, AREA_SIZE)
plt.ylim(0, AREA_SIZE)
plt.show()

```

## Oluřan Grafik:



řekil 1: Pembe noktalar: řehirleri, gri çizgiler: tüm olası baęlantıları, kırmızı çizgi: bulunan turu gösterir.

## GitHub Linki:

<https://github.com/busrahazer/progressive-tsp-research>