**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**

1. **Rastgele Nokta Üretimi:**
   * 50x50 alanda 5 nokta rastgele oluşturulmuştur.
   * random.seed(42) değeri kullanılarak her çalışmada aynı noktalar üretilmiştir.
2. **Graf Modeli:**
   * Her nokta bir düğüm olarak eklenmiş, tüm düğümler birbirine bağlanmıştır.
   * Kenar ağırlıkları Öklid mesafesi ile hesaplanmıştır.
3. **Sezgisel Algoritma (En Yakın Komşu):**
   * Başlangıç noktasından başlanır.
   * En yakın ziyaret edilmemiş komşu seçilerek ilerlenir.
   * Tüm noktalar gezildikten sonra başlangıç noktasına dönülür.
   * Bu yöntem hızlıdır ve küçük grafiklerde iyi sonuç verir.
4. **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

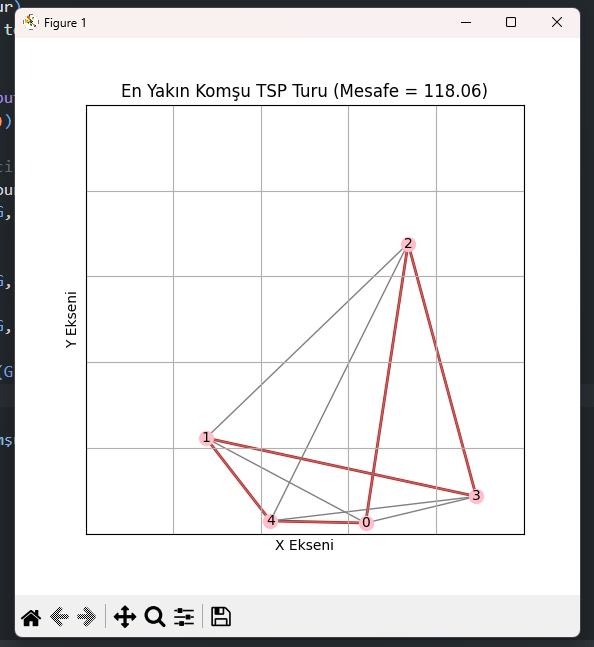
plt.title(f"En Yakın Komşu TSP Turu (Mesafe = {total\_distance:.2f})")

plt.xlim(0,AREA\_SIZE)

plt.ylim(0,AREA\_SIZE)

plt.show()

**Oluşan Grafik:**



Şekil 1: Pembe noktalar: şehirleri, gri çizgiler: tüm olası bağlantıları, kırmızı çizgi: bulunan turu gösterir.

**GitHub Linki:**

[**https://github.com/busrahazer/progressive-tsp-research**](https://github.com/busrahazer/progressive-tsp-research)