Approximation

July 19, 2020

In [ ]:
```
import networkx as nx
import math


# This function computes the distance between two points.
def dist(x1, y1, x2, y2):
    return math.sqrt((x1 - x2) ** 2 + (y1 - y2) ** 2)


# This function receives a list of 2-tuples representing the points' coordinates,
# and returns the corresponding graph.
def get_graph(coordinates):
    g = nx.Graph()
    n = len(coordinates)
    for i in range(n):
        for j in range(i + 1):
            g.add_edge(i, j, weight=dist(coordinates[i][0], coordinates[i][1], coordinat
    return g

# This function computes the weight of the given cycle.
def cycle_length(g, cycle):
    # Checking that the number of vertices in the graph equals the number of vertices in
    assert len(cycle) == g.number_of_nodes()
    # Write your code here.
    return sum(g[cycle[i]][cycle[i + 1]]['weight'] for i in range(len(cycle) - 1)) + g[c
```

In [ ]:
```
# Copy your implementation of the 2-approximation algorithm here.
def approximation(g):
```

In [ ]:
```
# Example 1.
# Compare the output of your approximation to the optimal solution on the following exam
coordinates = [(181, 243), (101, 143), (100, 216), (167, 15), (37, 201), (163, 226), (2,
optimal_cycle = [0, 5, 9, 2, 4, 1, 8, 7, 6, 3, 10]
g = get_graph(coordinates)
optimal_length = cycle_length(g, optimal_cycle) # 813.5762308235903
print("Example 1. The length of an optimal cycle is", optimal_length)
print("Example 1. The length of the cycle found by 2-approximation is", approximation(g)
```

In [ ]:
```
# Example 2.
# Check how fast your algorithm works on the following graph on 300 vertices.
# Also see how close an approximation solution is to an optimal one.
coordinates = [(145, 176), (185, 244), (67, 192), (5, 137), (165, 154), (106, 286), (132
optimal_cycle = [0, 60, 6, 250, 215, 102, 199, 275, 298, 241, 231, 77, 203, 184, 187, 28
g = get_graph(coordinates)
optimal_length = cycle_length(g, optimal_cycle) # 3899.6569479386735
print("Example 2. The length of an optimal cycle is", optimal_length)
print("Example 2. The length of the cycle found by 2-approximation is", approximation(g)

# You might want to copy these coordinates to your Jupiter Notebook to visualize the dat
```