Projekt zaliczeniowy

Programowanie genetyczne w języku Python

Prowadzący: prof. dr hab. inż. Oleksandr Petrov

Autor: Bartosz Nguyen Van

Kierunek: Informatyka i Ekonometria

Kraków, 2022

Cel projektu

Celem projektu jest znalezienie najkrótszej trasy łączącej 71 wybranych sklepów "Castorama" w Polsce. Jest to tzw. "problem komiwojażera".

W celu osiągnięcia postawionego celu będzie minimalizowana suma odległości pomiędzy poszczególnymi lokalizacjami sklepów.

Biblioteki

Użyte biblioteki w projekcie:

In [44]:
```
import numpy as np
import random
import pandas as pd
import math
import matplotlib.pyplot as plt
```

Wczytanie i opis danych

In [45]:
```
#wczytanie danych

coordinates = pd.read_csv("wspolrzedne.csv", delimiter =";")

print(coordinates.head())
```

```
   ID       Lat      Long                                      Name
0   1  52.24131  20.93087     Castorama ul. Górczewska 124 01-460 Warszawa
1   2  52.17226  20.93926   Castorama ul. Al. Krakowska 75 02-183 Warszawa
2   3  52.20193  20.93330       Castorama ul. Popularna 71 02-473 Warszawa
3   4  52.30321  21.05485       Castorama ul. Głębocka 15A 03-287 Warszawa
4   5  52.23717  21.11879      Castorama ul. Grochowska 21 04-186 Warszawa
```

Powyżej zaprezentowano 5 pierwszych wierszych w wykorzystywanym zbiorze danych . Lokalizacja sklepów jest określona za pomocą współrzędnych geograficznych:

- latitude, czyli szerokości geograficznej
- longitude, czyli długości geograficznej

W pierwszej kolumnie jest **ID** danego sklepu natomiast w kolumnie **Name** można znaleźć dokładny adres poszczególnych sklepów.

Przygotowanie danych

Dane zostaną przekształcone do listy krotek. Krotka zawiera odpowiednio szerokość i długość geograficzną.

In [46]:
```
cities = []

for i in range(0, len(coordinates)):
    city = (coordinates.iloc[i,1], coordinates.iloc[i,2])
    cities.append(city)
```

Tak się prezentują przekształcone dane:

In [47]:
```
#wszystkie miejsce w jednej liscie
print(cities[1:5])
```

```
[(52.17226, 20.93926), (52.20193, 20.9333), (52.30321, 21.0548499), (52.23717, 21.11879)]
```

Stworzenie funkcji

Funkcja obliczająca dystans pomiędzy współrzędnymi geograficznymi

Pierwsza stworzona funkcja w projekcie pozwoli na obliczenie dystansu (w kilometrach) pomiędzy dwoma miejscami, których lokalizacja jest określona za pomocą współrzędnych geograficznych.

Dla uproszczenia przyjęto najkrótszy dystans nad powierzchnią ziemi (tj. według lotu ptaka). Nie uwzględniono w dystansie ulic itp.

Użyto tzw. formuły "haversine".

In [48]:
```
# x[0] latitude1 - szerokosc geo1
# y[0] latitude2 - szerokosc geo2

# x[1] longitude1 - dlugosc geo 1
# y[1] longitude2 - dlugosc geo 1

def distance_between_coordinates(x, y):

    #promien ziemski
    R = 6371e3 #w metrach

    #szerokosc na radiany
    phi1 = x[0] * math.pi/180
    phi2 = y[0] * math.pi/180

    #delta latitude
    dphi = (abs(x[0]-y[0]))* math.pi/180

    #delta longitude
    dlambda = (abs(x[1]-y[1]))* math.pi/180

    #zmienne pomocnicze
    a = math.sin(dphi/2) * math.sin(dphi/2) + math.cos(phi1) * math.cos(phi2) * math.sin(dla
mbda/2) * math.sin(dlambda/2)
    c = 2 * math.atan2(math.sqrt(a), math.sqrt(1-a))

    #odleglosc w metrach
    d = R * c

    return d/1000 # zamiana na kilometry
```

Funkcja tworząca macierz odległości między wszystkimi lokalizacjami

In [49]:
```
#obliczenie odleglosci pomiedzy miejscami

def calculate_distance(cities):

    distances = np.zeros((len(cities), len(cities)))

    for i in range(len(cities)):
        for j in range(len(cities)):
            distance = distance_between_coordinates(cities[i], cities[j])
            distances[i][j] = distance

    return distances
```

Funkcja zwrająca losową kolejność odwiedzanych miejsc

In [50]:
```
#kolejnosc odwiedzanych miejsc

def random_city_order(n):
    order_list = []
    for i in range(n):
        order_list.append(i)
    random.shuffle(order_list)
    return order_list
```

Funkcja - wizualizacja na wykresie

In [51]:
```
#wizualizacja na wykresie

def draw_plot(cities, t):

    wspx, wspy = zip(*cities)

    plt.scatter(wspx, wspy)

    for j in range(len(t)):
        plt.annotate(t[j], cities[t[j]])

    for j in range(len(t)-1):
        lines = plt.plot((cities[t[j]][0],cities[t[j+1]][0]), (cities[t[j]][1],cities[t[j+1
]][1]))
        plt.setp(lines, color = 'r', linewidth = 1.5)

    plt.show()
```

Obliczenie całkowitej odległości dla drogi

In [52]:
```
def total_distance(distances, t):
    distance = 0

    for i in range(len(t)-1):
        distance += distances[t[i]][t[i+1]]
    return distance
```

Zamiana losowych dwóch lokalizacji miejscami

In [53]:
```
#zamiana losowych dwoch miast miejscami
#do mutacji

def switch_order(t):

    new_t = t[:]

    switch_point1 = random.randint(0, len(t)-1)
    switch_point2 = random.randint(0, len(t)-1)

    temp = new_t[switch_point1]
    new_t[switch_point1] = new_t[switch_point2]
    new_t[switch_point2] = temp

    return new_t
```

Wybór n najlepszych dróg

In [54]:
```
#wybor n najlepszych
def choose_n_best(lst, n, rev=False):
    index = range(len(lst))
    s = sorted(index, reverse=rev, key=lambda i: lst[i])
    return s[:n]
```

Krzyżówka

In [55]:
```
#krzyzowanie

def crossover(o1, o2, P=0.5):
    if random.random() > P:

        crossover_point = random.randint(0, len(o1)-1)

        crossed_o1 = o1[:crossover_point]
        for element in o2:
            if element not in crossed_o1:
                crossed_o1.append(element)

        crossed_o2 = o2[:crossover_point]
        for element in o1:
            if element not in crossed_o2:
                crossed_o2.append(element)

        return crossed_o1, crossed_o2
    else:
        return o1, o2
```

Mutacja

In [56]:
```
#mutacja
def mutation(o1, P = 0.3):
    mutated_o1 = o1[:]
    if random.random() > P:
        mutated_o1 = switch_order(o1)

    return mutated_o1
```

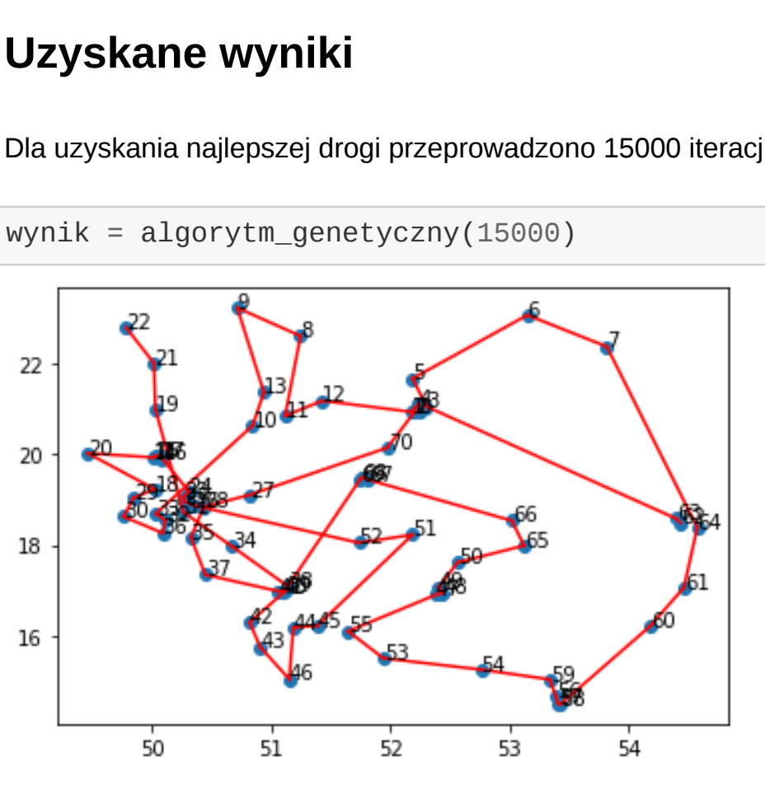
Algorytm genetyczny

In [57]:
```
def algorytm_genetyczny(liczba_pokolen, n_pop = 16):

    #miasta
    miasta = cities

    #obliczenie macierzy odleglosci
    macierz_odleglosci = calculate_distance(miasta)

    #losowanie populacji
    populacja = [random_city_order(71) for i in range(n_pop)]


    for j in range(liczba_pokolen):

        #dlugosc trasy dla kazdego osobnika
        dlugosci_tras = [total_distance(macierz_odleglosci, populacja[i]) for i in range(len
(populacja))]

        #wybieranie rodzicow
        wybrani_rodzice_id = choose_n_best(dlugosci_tras,10)
        wybrani_rodzice = [populacja[i] for i in wybrani_rodzice_id]


        #losowe dobieranie w pary
        losowe_pary = [random.sample(wybrani_rodzice, 2) for i in range(5)]

        dzieci = []

        #krzyzowka
        for para in losowe_pary:
            c1, c2 = crossover(para[0], para[1])
            dzieci.append(c1)
            dzieci.append(c2)

        #mutacja
        dzieci = [mutation(d) for d in dzieci]

        #dlugosci tras dzieci
        dlugosci_tras_dzieci = [total_distance(macierz_odleglosci, dzieci[i]) for i in range
(len(dzieci))]

        #wybranie najlepszych dzieci
        wybrane_dzieci_id = choose_n_best(dlugosci_tras_dzieci, 6)
        wybrane_dzieci = [dzieci[i] for i in wybrane_dzieci_id]

        #nowa populacja
        populacja = wybrani_rodzice + wybrane_dzieci

        #najlepsza aktualna dlugosc i trasa
        dlugosci_tras_nowa_populacja = [total_distance(macierz_odleglosci, populacja[i]) for
i in range(len(populacja))]
        najlepsza_trasa_id = choose_n_best(dlugosci_tras_nowa_populacja, 1)
        najlepsza_trasa = populacja[najlepsza_trasa_id[0]]
        najlepsza_dlugosc = dlugosci_tras_nowa_populacja[najlepsza_trasa_id[0]]

  #      print("Iteracja nr ",  j+1)


    draw_plot(miasta, najlepsza_trasa)
    print("Dlugosc trasy: ", round(najlepsza_dlugosc,2), "km")
    print("Najlepsza trasa: ", najlepsza_trasa)

    return najlepsza_trasa
```

Uzyskane wyniki

Dla uzyskania najlepszej drogi przeprowadzono 15000 iteracji w stworzonym algorytmie genetycznym.

In [60]:
```
wynik = algorytm_genetyczny(15000)
```



```
Dlugosc trasy:  4599.77 km
Najlepsza trasa:  [22, 21, 19, 31, 34, 38, 39, 41, 37, 35, 28, 27, 70, 2, 0, 4, 1, 12, 11, 8,
9, 13, 10, 33, 32, 36, 30, 29, 18, 20, 14, 15, 17, 16, 24, 23, 25, 26, 52, 51, 45, 44, 46, 4
3, 42, 40, 69, 68, 67, 66, 65, 50, 49, 48, 47, 55, 53, 54, 59, 56, 57, 58, 60, 61, 64, 7, 6,
5, 3, 63, 62]
```

Na wykresie powyżej pokazano jak prezentuje się otrzymana trasa.

Poniżej pokazano kolejność sklepów w uzyskanej trasie.

In [61]:
```
kolejnosc_nazwy = [coordinates.iloc[i,3] for i in wynik]

for i in range(len(kolejnosc_nazwy)):
    print(i+1," ",kolejnosc_nazwy[i])
```

```
1   Castorama ul. Lwowska 17 37-700 Przemyśl
2   Castorama Al. Rejtana 67 35-326 Rzeszów
3   Castorama ul. Nowodąbrowska 127 33-100 Tarnów
4   Castorama ul. Pszczyńska 315 44-100 Gliwice
5   Castorama ul. Wiejska 141a 45-302 Opole
6   Castorama ul. Bolesława Krzywoustego 126a 51-421 Wrocław
7   Castorama ul. Legnicka 58 54-204 Wrocław
8   Castorama ul. Czekoladowa 3 55-040 Wrocław
9   Castorama ul. Zygmunta Krasińskiego 31 48-303 Nysa
10   Castorama ul. Armii Krajowej 40 47-200 Kędzierzyn-Koźle
11   Castorama ul. Obwodnica 16 42-600 Tarnowskie Góry
12   Castorama ul. Jana Pawła II 2 42-200 Częstochowa
13   Castorama ul. Wyszyńskiego 10 96-100 Skierniewice
14   Castorama ul. Popularna 71 02-473 Warszawa
15   Castorama ul. Górczewska 124 01-460 Warszawa
16   Castorama ul. Grochowska 21 04-186 Warszawa
17   Castorama ul. Al. Krakowska 75 02-183 Warszawa
18   Castorama ul. Energetyków 1 26-615 Radom
19   Castorama AL. Józefa Piłsudzkiego 2 26-110 Skarżysko-Kamienna
20   Castorama ul. Mełgiewska 16 c 20-234 Lublin
21   Castorama ul. Dzieci Zamojszczyzny 4 22-400 Zamość
22   Castorama ul. Stefana Żeromskiego 13 27-400 Ostrowiec Świętokrzyski
23   Castorama ul. Wrzosowa 42 25-211 Kielce
24   Castorama Al. Zjednoczonej Europy 26 44-240 Żory
25   Castorama ul. Obwiednia Północna 21 44-200 Rybnik
26   Castorama ul. Rybnicka 95 47-400 Racibórz
27   Castorama ul. Graniczna 80 43-400 Cieszyn
28   Castorama ul. Warszawska 186 43-300 Bielsko-Biała
29   Castorama ul. Zatorska 1 32-600 Oświęcim
30   Castorama ul. Szaflarska 176 34-400 Nowy Targ
31   Castorama ul. Zakopiańska 62 30-418 Kraków
32   Castorama ul. Walerego Sławka 1 30-633 Kraków
33   Castorama ul. Pilotów 6 31-462 Kraków
34   Castorama ul. Sosnowiecka 147 31-345 Kraków
35   Castorama ul. Długosza 82 41-219 Sosnowiec
36   Castorama ul. Roździeńskiego 198 40-315 Katowice
37   Castorama ul. Sportowa 31 41-500 Chorzów
38   Castorama ul. Al. Jana Nowaka-Jeziorańskiego 27 41-923 Bytom
39   Castorama ul. Tylna 17-23 62-800 Kalisz
40   Castorama ul. Ogrodowa 31 62-571 Stare Miasto k/Konina
41   Castorama ul. Komisji Edukacji Narodowej 1 59-300 Lubin
42   Castorama ul. Roberta Schumana 9 59-220 Legnica
43   Castorama ul. Jeleniogórska 77 59-900 Zgorzelec
44   Castorama ul. Al. Jana Pawła II 11 58-506 Jelenia Góra
45   Castorama ul. H. Wieniawskiego 21 58-306 Wałbrzych
46   Castorama ul. Graniczna 2a 54-610 Wrocław
47   Castorama ul. Wróblewskiego 31 93-566 Łódź
48   Castorama ul. Wydawnicza 13 92-333 Łódź
49   Castorama ul. Sikorskiego 2/6 91-497 Łódź
50   Castorama ul. Szosa Bydgoska 102 A 87-100 Toruń
51   Castorama ul. Szubińska 5 85-312 Bydgoszcz
52   Castorama ul. Pałucka 1 62-200 Gniezno
53   Castorama ul. Sienkiewicza 23 62-020 Swarzędz
54   Castorama ul. Murawa 39 61-655 Poznań
55   Castorama ul. Górecka 30A 60-201 Poznań
56   Castorama ul. Poniatowskiego 10 67-200 Głogów
57   Castorama ul. Wojska Polskiego 19 65-077 Zielona Góra
58   Castorama ul. Czartoryskiego 1 66-400 Gorzów Wielkopolski
59   Castorama ul. Tadeusza Kościuszki 73A 73-110 Stargard Szczeciński
60   Castorama ul. Wiosenna 80 70-807 Szczecin
61   Castorama ul. Południowa 21 71-001 Szczecin
62   Castorama ul. Ku Słońcu 67b 71-047 Szczecin
63   Castorama ul. Paderewskiego 2 75-736 Koszalin
64   Castorama ul. Hubalczyków 2 76-200 Słupsk
65   Castorama ul. Grunwaldzka 5 84-230 Rumia
66   Castorama ul. Ciepła 2 19-300 Ełk
67   Castorama ul. Narodowych Sił Zbrojnych 13 15-960 Białystok
68   Castorama ul. Warszawska 63B 05-300 Stojadła
69   Castorama ul. Głębocka 15A 03-287 Warszawa
70   Castorama ul. Grunwaldzka 262 80-314 Gdańsk
71   Castorama ul. Odyseusza 2 80-299 Gdańsk
```