

Michał Kaniowski, Mikołaj Kapera

Środa 15:00

Struktury danych:

- $C_{m,i}$ – koszt podróży pojazdu danego typu m
- l_i – opóźnienie, wynikające z rozkładu
- x_i, y_i – położenie miasta
- x_0, y_0 – miasto początkowe
- v_m – prędkość pojazdu danego typu m

Postać rozwiązania

- J – kolejność odwiedzonych miast
- J_k – k -te miasto w kolejności

Funkcja celu

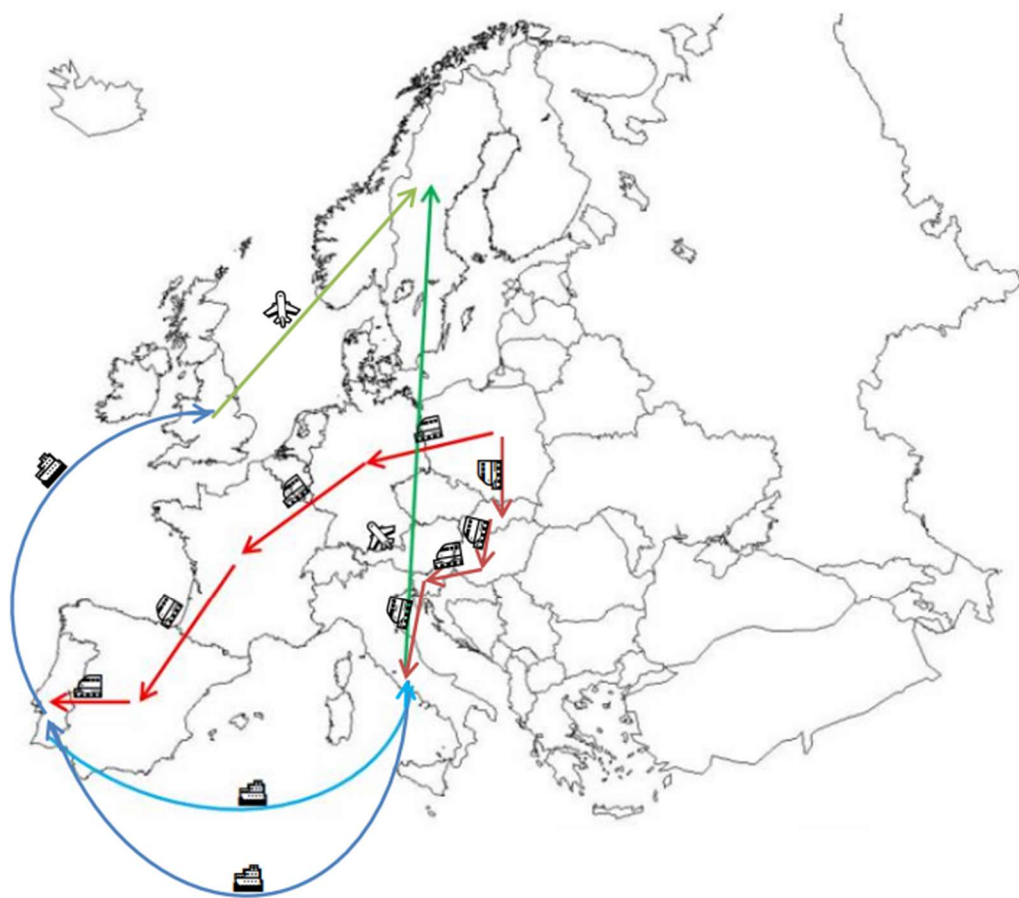
- Minimalizacja czasu podróży z uwzględnieniem kosztów transportu
- $f(J) = \sum_{i=0}^N t_i \cdot c_{m_i} + l_i \rightarrow \min$
- t_i – czas podróży między dwoma miastami
 - $t_i = \frac{s_i}{v_m}$
 - $s_{i,i+1} = \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}$
- c_{mi} – koszt transportu danego typu między miastami
- l_i – opóźnienie wynikające z rozkładu, dopełnienie do pełnej godziny
 - $l_i = [t] - t$

Warunki ograniczające

- Maksymalny koszt podróży
- Maksymalny czas podróży

Zmienne pomocnicze

- Rozkład środków transportu zależny od dnia i godziny określający połączenia między miastami
- Średnia prędkość środków transportu



Przykładowe struktury danych:

```
class Vehicle:
    def __init__(self, v: float, cost: float):
        self.v = v
        self.cost = cost

class Schedule:
    def __init__(self, time: int, bus: List[List], train: List
[List], ship: List[List], plane: List[List]):
        self.time = time
        self.bus = bus
        self.train = train
        self.ship = ship
        self.plane = plane

class City:
    def __init__(self, x: float, y: float, schedule: Schedule):
        self.x = x
        self.y = y
        self.schedule = schedule
```