

# **Sprawozdanie**

## Problem transportowy wielu produktów

Bartosz Wlazło, Przemysław Jekiel

22 stycznia 2019

# 1 Wstęp

## 1.1 Problematyka

Celem niniejszego ćwiczenia było rozwiązanie problemu transportowego wielu produktów.

## 1.2 Cel ćwiczenia

Po wykonaniu tego ćwiczenia powinniśmy opanować importowanie danych (przy użyciu biblioteki Pandas), umieć skonfigurować i poprawnie wyświetlić dane w części frontend oraz umieć korzystać z biblioteki optymalizacyjnej pycipopt.

## 1.3 Przyjęty przypadek użycia

Osoba X prowadzi sieć sklepów spożywczych. Największe koszty, które ponosi firma są związane z transportem produktów z magazynów do sklepów. W celu obniżenia ich udała się ona do firmy OPTTech w celu opracowania narzędzia do planowania tras i ilości zabieranych produktów.

# 2 Narzędzia

Do obliczeń oraz wizualizacji danych i wyników zostały użyte następujące narzędzia:

- Python - mikroframework Flask - realizacja zadania frontend tj. wizualizacja danych i wyników oraz GUI użytkownika
- Python - biblioteka Pycipopt - realizacja zadania backend tj. stworzenia modelu problemu i obliczenia wyniku
- Python - biblioteka Pandas - realizacja zadania backend tj. import i obróbka importowanych z pliku bazowodanego danych

# 3 Założenia projektowe

## 3.1 Uproszczające

W celu szybszej pracy kody próbka danych został ograniczona do:

- 5 sklepów

- 3 magazynów
- 4 typów produktów
- nieskończonej liczby środków transportu
- wartości ograniczeń są wyrażone w liczbach całkowitych

Zostało także przyjęte uproszczenie iż każdy produkt jest magazowany w tej samej liczbie.

### 3.2 Przyjęte zmienne

- $I$  - zbiór sklepów  $i$
- $i$  -  $i$ -ty sklep
- $J$  - zbiór magazynów  $j$
- $j$  -  $j$ -ty magazyn
- $K$  - zbiór produktów  $k$
- $k$  -  $k$ -ty produkt
- $M_j$  - pojemność  $j$ -tego magazynu
- $c_{i,j,k}$  - koszt dostawy do  $i$ -tego sklepu z  $j$ -tego magazynu  $k$ -tego produktu
- $d_{i,k}$  - zapotrzebowanie  $i$ -tego sklepu na  $k$ -ty produkt

### 3.3 Szukane

$x_{i,j,k}$  - ilość wysłanych sztuk produktu  $k$  do  $i$  sklepu z  $j$  magazynu

### 3.4 Model matematyczny

#### 3.4.1 Funkcja celu

$$\text{minimize} \sum_{i=1}^n \sum_{j=1}^m \sum_{k=1}^K c_{i,j,k} * x_{i,j,k}$$

### 3.4.2 Ograniczenia

1.  $\sum_{j=1}^m x_{i,j,k} = d_{i,k}$  dla  $i=1,\dots,n$ ,  $k=1,\dots,K$
2.  $\sum_{i=1}^n \sum_{k=1}^K x_{i,j,k} \leq M_j$  dla  $j=1,\dots,m$
3.  $x_{i,j,k} \geq 0$  dla  $i=1,\dots,n$ ,  $j=1,\dots,m$ ,  $k=1,\dots,K$

## 4 Założenia projektowe

### 4.1 Dane ograniczeń

#### 4.1.1 *Pojemności magazynów*

Magazyn	Pojemność
Magazyn 1	3000
Magazyn 2	3000
Magazyn 3	3000

#### 4.1.2 *Dostępność produktów w danych magazynach*

Magazyn	Produkt
Magazyn 1	2, 4
Magazyn 2	1, 2, 3
Magazyn 3	2, 3, 4

## 4.2 Dane

### 4.2.1 Zapotrzebowanie sklepów na dany produkt

Magazyn	Produkt	Zapotrzebowanie
Sklep 1	1	80
Sklep 1	2	85
Sklep 1	3	300
Sklep 1	4	6
Sklep 2	1	270
Sklep 2	2	160
Sklep 2	3	400
Sklep 2	4	7
Sklep 3	1	250
Sklep 3	2	130
Sklep 3	3	350
Sklep 3	4	4
Sklep 4	1	160
Sklep 4	2	60
Sklep 4	3	200
Sklep 4	4	3
Sklep 5	1	180
Sklep 5	2	40
Sklep 5	3	150
Sklep 5	4	5

### 4.2.2 Wagi produktów

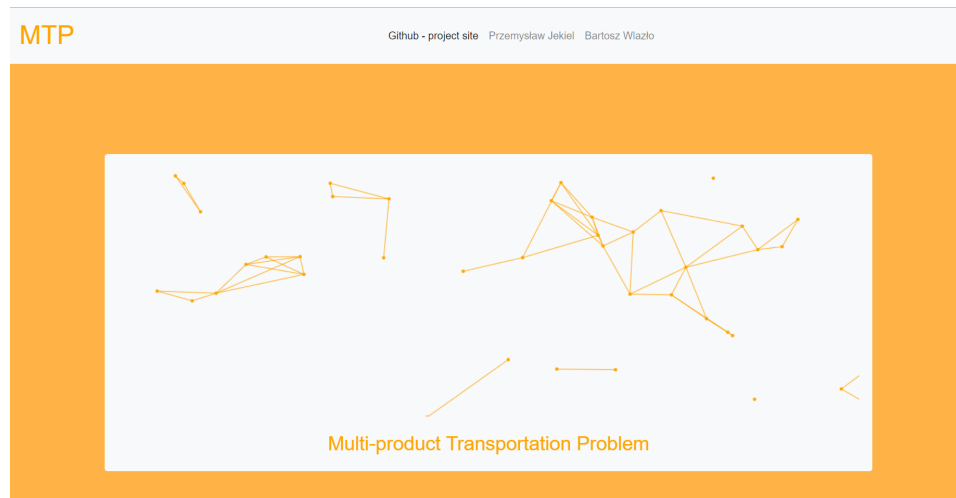
Produkt	Waga
Produkt 1	5
Produkt 2	2
Produkt 3	3
Produkt 4	4

#### 4.2.3 *Koszt dostawy do danego sklepu z magazynu*

Sklep	Magazyn	Koszt
Sklep 1	Magazyn 1	4
Sklep 2	Magazyn 1	5
Sklep 3	Magazyn 1	6
Sklep 4	Magazyn 1	8
Sklep 5	Magazyn 1	10
Sklep 1	Magazyn 2	6
Sklep 2	Magazyn 2	4
Sklep 3	Magazyn 2	3
Sklep 4	Magazyn 2	5
Sklep 5	Magazyn 2	8
Sklep 1	Magazyn 3	9
Sklep 2	Magazyn 3	7
Sklep 3	Magazyn 3	4
Sklep 4	Magazyn 3	3
Sklep 5	Magazyn 3	4

## 5 Interfejs użytkownika

### 5.1 Widok strony głównej



### 5.2 Widok strony z wynikami

MTP

Github - project site Przemysław Jekiel Bartosz Wlazło

Całkowity koszt wynosi: 43536.0

Wynik

Sklep:	Magazyn:	Produkt:	Ilość:
Sklep 5	Magazyn 3	Produkt 4	5.0
Sklep 1	Magazyn 1	Produkt 2	85.0
Sklep 3	Magazyn 3	Produkt 4	4.0
Sklep 1	Magazyn 1	Produkt 4	6.0
Sklep 4	Magazyn 3	Produkt 2	60.0
Sklep 4	Magazyn 3	Produkt 3	200.0
Sklep 3	Magazyn 2	Produkt 2	130.0
Sklep 3	Magazyn 2	Produkt 3	350.0

## 6 Wynik

Dla podanych danych całkowity koszt wynosi: 43536.0

Tabela z wynikami

Sklep	Magazyn	Produkt	Ilość sztuk
Sklep 5	Magazyn 3	Produkt 4	5.0
Sklep 1	Magazyn 1	Produkt 2	85.0
Sklep 3	Magazyn 3	Produkt 4	4.0
Sklep 1	Magazyn 1	Produkt 4	6.0
Sklep 4	Magazyn 3	Produkt 2	60.0
Sklep 4	Magazyn 3	Produkt 3	200.0
Sklep 3	Magazyn 2	Produkt 2	130.0
Sklep 3	Magazyn 2	Produkt 3	350.0
Sklep 2	Magazyn 2	Produkt 1	270.0
Sklep 4	Magazyn 2	Produkt 1	160.0
Sklep 2	Magazyn 1	Produkt 4	7.0
Sklep 2	Magazyn 2	Produkt 3	400.0
Sklep 2	Magazyn 2	Produkt 2	160.0
Sklep 5	Magazyn 2	Produkt 1	180.0
Sklep 3	Magazyn 2	Produkt 1	250.0
Sklep 1	Magazyn 2	Produkt 3	300.0
Sklep 5	Magazyn 3	Produkt 3	150.0
Sklep 5	Magazyn 3	Produkt 2	40.0
Sklep 4	Magazyn 3	Produkt 4	3.0
Sklep 1	Magazyn 2	Produkt 1	80.0

## 7 Analiza otrzymanych wyników

Analiza wyników jest trudna do osiągnięcia ze względu na charakter zadania. Porównanie z wynikami rzeczywistymi jest niemożliwe ze względu na brak dostępu do takowych. Model został wykonany na czysto teoretycznym wycinku rzeczywistości.

## 8 Wnioski

Zbudowanie modelu w skali pozwala na uniknięcie kosztów związanych z wykonywaniem testów w rzeczywistym środowisku. Oszczędzane są również inne zasoby takie jak chociażby czas.

Dzięki temu ćwiczeniu opanowaliśmy wiedzę z zakresu:

- Frontend - konfiguracji oraz obsługi mikroframeworku flask w języku programowania Python



- Backend - operacji na danych dzięki obsłudze biblioteki Pandas w języku programowania Python
- Backend - optymalizacji dzięki bibliotece pycipopt w języku programowania Python
- tworzenia plików tekstowych z wykorzystaniem LaTeX

## 9 Kod

Kod jest dostępny w repozytorium: <https://github.com/pj30/Projekt-MMPPiZ>