

# Podział sieci logistycznej firmy kurierskiej

Jakub Komosa, Olaf Naruszko

April 9, 2025

## Contents

|          |                                  |          |
|----------|----------------------------------|----------|
| <b>1</b> | <b>Opis problemu</b>             | <b>2</b> |
| <b>2</b> | <b>Opis działania systemu</b>    | <b>2</b> |
| <b>3</b> | <b>Teoria i koncepcje</b>        | <b>3</b> |
| 3.1      | Reprezentacja grafu              | 3        |
| 3.2      | Podział grafu                    | 4        |
| 3.3      | Efektywność i plan działania     | 4        |
| <b>4</b> | <b>Format danych wejściowych</b> | <b>4</b> |
| <b>5</b> | <b>Podsumowanie</b>              | <b>6</b> |

# 1 Opis problemu

W ramach projektu rozważany jest problem podziału ogólnokrajowej sieci logistycznej firmy kurierskiej na trzy niezależne regiony operacyjne. Inspiracją do wyboru tematu była potrzeba lepszej optymalizacji i organizacji pracy dużej firmy kurierskiej.

Sieć logistyczna reprezentowana jest w postaci grafu, gdzie:

- wierzchołki oznaczają centra logistyczne (np. magazyny, punkty odbioru, huby przeładunkowe),
- krawędzie reprezentują trasy przesyłek między punktami.

Wraz z rozwojem firmy rośnie potrzeba regionalizacji operacji logistycznych. Celem projektu jest dokonanie takiego podziału grafu, aby:

- każdy region zawierał zbliżoną liczbę węzłów,
- liczba połączeń między regionami była jak najmniejsza,
- sieć połączeń w obrębie każdego regionu była możliwie spójna i wydajna,
- możliwe było przetwarzanie danych i podejmowanie decyzji lokalnie, bez konieczności ciągłej komunikacji między regionami.

# 2 Opis działania systemu

System realizuje pełen proces analizy i podziału sieci logistycznej firmy kurierskiej, reprezentowanej jako graf nieskierowany zapisany w pliku wejściowym w formacie `.csrrg`. Działanie systemu składa się z następujących etapów:

1. **Wczytanie danych:** System odczytuje dane z pliku `.csrrg`, zawierającego niestandardowy opis grafu logistycznego. Dane wejściowe obejmują m.in. listy indeksów węzłów, grupy połączeń oraz wskaźniki strukturalne.
2. **Konwersja danych do struktur pamięci:** Na podstawie odczytanych danych tworzona jest wewnętrzna reprezentacja grafu w postaci listy sąsiedztwa oraz macierzy sąsiedztwa. Struktury te umożliwiają efektywne przeszukiwanie oraz manipulację grafem.
3. **Wstępna analiza grafu:** System przeprowadza podstawową analizę, m.in. określa liczbę węzłów, krawędzi oraz spójność grafu. W razie potrzeby może także uwzględnić wagi krawędzi, np. odpowiadające liczbie przesyłek między punktami.

4. **Partycjonowanie grafu:** Główna funkcjonalność systemu polega na podziale grafu na trzy zrównoważone partycje (regiony logistyczne), tak aby:

- liczba węzłów w każdej partycji była podobna,
- liczba przeciętych krawędzi (łączących różne partycje) była jak najmniejsza,
- partycje były spójne wewnętrznie.

W tym celu wykorzystywany jest algorytm podziału grafu.

5. **Generowanie wyników:** Po zakończeniu partycjonowania system generuje:

- przypisanie każdego węzła do jednej z trzech partycji,
- listę i liczbę krawędzi przecinających granice między-partycyjne,
- (opcjonalnie) wizualizację grafu z kolorowaniem partycji.

6. **Zapis wyników:** Wyniki są zapisywane do pliku wyjściowego w formacie tekstowym, umożliwiając dalszą analizę lub wizualizację danych z wykorzystaniem zewnętrznych narzędzi.

System został zaprojektowany w sposób modularny, co umożliwia łatwe rozszerzanie o nowe algorytmy podziału, obsługę dodatkowych formatów danych, a także integrację z graficznym interfejsem użytkownika (GUI).

## 3 Teoria i koncepcje

### 3.1 Reprezentacja grafu

Grafy mogą być reprezentowane na kilka sposobów:

- **Macierz sąsiedztwa** – intuicyjna, ale wymaga dużo pamięci przy dużych, rzadkich grafach.
- **Lista sąsiedztwa** – dynamiczna struktura opisująca połączenia między węzłami. Efektywna przy ograniczonej liczbie krawędzi.
- **Struktury pomocnicze** – metadane, takie jak liczba węzłów, krawędzi i informacje o grupach.
- **Format CSR (Compressed Sparse Row)** – przechowuje tylko niezerowe elementy macierzy, korzystając z trzech tablic: wskaźników wierszy, indeksów kolumn oraz (opcjonalnie) wartości.

### 3.2 Podział grafu

Podział grafu polega na rozdzieleniu węzłów na oddzielne grupy w taki sposób, aby jak najmniej krawędzi łączyło węzły należące do różnych grup.

W projekcie rozważamy dwa podejścia podstawowe:

- Wykrywanie spójnych komponentów przy użyciu algorytmów przeszukiwania DFS lub BFS.
- Prosty podział grafu na części o zbliżonej liczbie węzłów, przy jednoczesnym minimalizowaniu liczby krawędzi między grupami.

W kontekście sieci logistycznej taki podział pozwala ograniczyć liczbę przesyłek realizowanych między regionami, poprawiając efektywność i lokalność operacyjną systemu.

### 3.3 Efektywność i plan działania

Implementacja opiera się na:

- **Efektywności pamięciowej** – format CSR umożliwia przechowywanie grafu o złożoności  $O(n+m)$ .
- **Szybkim dostępie do danych** – struktury umożliwiają szybkie wyszukiwanie sąsiadów węzła.
- **Modularnej realizacji** – odczyt danych, budowa struktur, implementacja algorytmów podziału oraz generowanie reprezentacji macierzowej są realizowane w oddzielnych modułach.

## 4 Format danych wejściowych

Plik wejściowy `.csrrg` zawiera opis grafu w pięciu sekcjach, przy czym liczby całkowite oddzielone są średnikami (;). Każda sekcja odpowiada:

1. **Maksymalna liczba węzłów w wierszu** – pojedyncza liczba (np. 18).
2. **Indeksy węzłów** – lista wszystkich indeksów.
3. **Wskaźniki początku wierszy** – tablica wskaźników do listy z punktu 2.
4. **Lista połączeń (krawędzie)** – pary węzłów tworzących krawędzie.
5. **Wskaźniki grup połączeń** – identyfikacja oddzielnych grup krawędzi.

### Przykładowy fragment:

```
18
3;5;6;9;10;...
0;0;8;11;18;27;...
0;72;39;91;4;54;...
0;6;12;16;21;...
```

### Walidacja pliku

Program sprawdza:

- Poprawność formatowania (liczby oddzielone średnikami).
- Obecność dokładnie pięciu sekcji.
- Spójność długości i zakresów wskaźników.

### Przykładowe wywołania:

```
./graph_divider graf.csrrg
```

## 5 Podsumowanie

Zaprojektowany system umożliwia efektywny podział dużej sieci logistycznej firmy kurierskiej na mniejsze, logicznie wydzielone regiony operacyjne. Dzięki zastosowaniu reprezentacji grafowej oraz algorytmów partycjonowania możliwe jest osiągnięcie równowagi pomiędzy liczbą punktów logistycznych w każdej partycji a minimalizacją połączeń między nimi.

Projekt łączy aspekty teoretyczne z praktycznymi wymaganiami rzeczywistych systemów logistycznych – wspiera decentralizację operacji, ogranicza potrzebę komunikacji między regionami i pozwala na efektywniejsze zarządzanie zasobami. Dzięki modularnej budowie system może być w przyszłości łatwo rozszerzony o nowe algorytmy, dodatkowe formaty wejściowe oraz graficzny interfejs użytkownika.

Rozwiązanie to może znaleźć zastosowanie nie tylko w logistyce, ale również w wielu innych dziedzinach wykorzystujących strukturę grafową, takich jak analiza sieci społecznych, zarządzanie infrastrukturą czy planowanie systemów rozproszonych.