

# Sprawozdanie: Algorytmy Optymalizacji Dyskretnej - Lista 4

Krzysztof Zając

27 stycznia 2026

## 1 Wstęp

Celem niniejszego sprawozdania jest analiza i przedstawienie wyników implementacji algorytmów maksymalnego przepływu oraz skojarzeń w grafach dwudzielnych. Zadania obejmowały implementację algorytmów Edmonds-Karp oraz relabel-to-front (lub wariantu najkrótszej ścieżki powiększającej - SAP), modelowanie problemu jako Programowanie Liniowe (LP) oraz badanie skojarzeń w grafach losowych.

## 2 Opis implementacji algorytmów (Zadania 1, 2, 4)

### 2.1 Struktura FlowNetwork

Podstawą implementacji jest klasa `FlowNetwork`, reprezentująca sieć przepływową za pomocą list sąsiedztwa. Każda krawędź przechowuje informację o wierzchołku docelowym, aktualnej przepustowości rezydualnej, indeksie krawędzi powrotnej oraz oryginalnej przepustowości.

### 2.2 Algorytm Edmondsa-Karpa

Zaimplementowano klasyczny algorytm Edmondsa-Karpa, który jest realizacją metody Forda-Fulkersona wykorzystującą przeszukiwanie wszerz (BFS) do znajdowania najkrótszych ścieżek powiększających w grafie rezydualnym.

- **Złożoność:**  $O(VE^2)$ .
- **Działanie:** W każdej iteracji BFS znajduje najkrótszą ścieżkę (w sensie liczby krawędzi) z źródła do ujścia. Przepływ jest zwiększany o "wąskie gardło" tej ścieżki.

### 2.3 Algorytm Shortest Augmenting Path (SAP)

Jako algorytm alternatywny użyto podejścia 'Shortest Augmenting Path' (implementacja wzorowana na metodach typu Goldberg-Tarjan/Relabel, ale z wykorzystaniem etykiet odległości do szybkiego znajdowania dopuszczalnych krawędzi).

- **Złożoność:** Teoretycznie  $O(V^2E)$  dla ogólnej metody, w praktyce często znacznie szybszy od EK na gęstych grafach.
- **Optymalizacje:** Użycie "Gap Heuristic" oraz wskaźników `current_arc` pozwala na unikanie ponownego przeglądania nasyconych krawędzi.

## 2.4 Maksymalne skojarzenie w grafie dwudzielnym (Zadanie 2)

Problem znalezienia maksymalnego skojarzenia w grafie dwudzielnym zredukowano do problemu maksymalnego przepływu: 1. Dodano super-źródło  $S$  połączone krawędziami o pojemności 1 ze wszystkimi wierzchołkami zbioru  $V_1$ . 2. Dodano super-ujście  $T$ , do którego wchodzi krawędzie o pojemności 1 od wszystkich wierzchołków zbioru  $V_2$ . 3. Istniejące krawędzie między  $V_1$  a  $V_2$  otrzymały pojemność 1 (lub  $\infty$ ). 4. Maksymalny przepływ w takiej sieci odpowiada liczności maksymalnego skojarzenia (z uwagi na całkowitoliczbowość przepływu).

## 3 Opis modeli Programowania Liniowego (Zadanie 3)

Dla problemu maksymalnego przepływu wygenerowano modele w formacie LP (rozwiązywalne przez GLPK).

### 3.1 Zmienne decyzyjne

Dla każdej krawędzi skierowanej  $(u, v)$  w grafie zdefiniowano zmienną decyzyjną  $f_{u,v}$  oznaczającą wielkość przepływu na tej krawędzi.

### 3.2 Funkcja celu

Celem jest maksymalizacja wypływu ze źródła  $s$ :

$$\max \sum_{v \in N(s)} f_{s,v} - \sum_{u \in N^{-1}(s)} f_{u,s}$$

(W praktyce często wystarczy suma wypływów, jeśli do źródła nic nie wraca).

### 3.3 Ograniczenia

1. **Ograniczenie przepustowości:** Dla każdej krawędzi  $(u, v)$ , przepływ nie może przekraczać przepustowości  $c_{u,v}$ :

$$0 \leq f_{u,v} \leq c_{u,v}$$

2. **Zachowanie przepływu:** Dla każdego węzła  $v \in V \setminus \{s, t\}$ , suma wpływów musi równać się sumie wypływów:

$$\sum_u f_{u,v} = \sum_w f_{v,w}$$

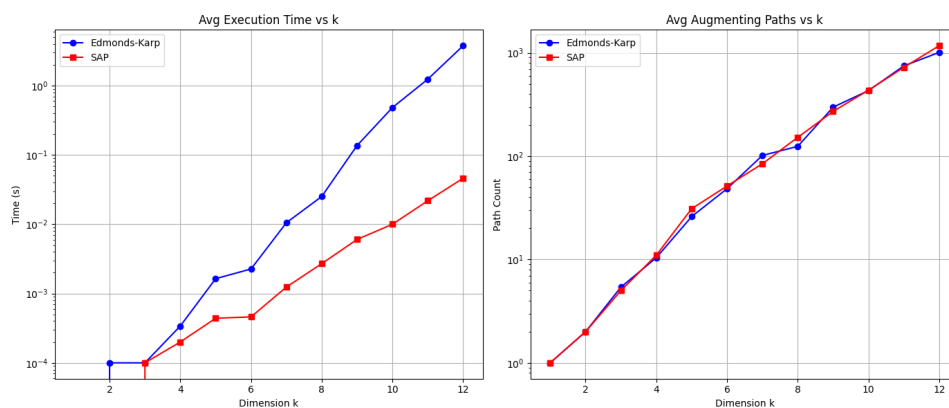
Model generowany przez program zapisuje te równania w standardzie CPLEX LP, np.:

```
Maximize
  obj: f_0_1 + f_0_2
Subject To
  c1: f_0_1 - f_1_3 = 0
  ...
```

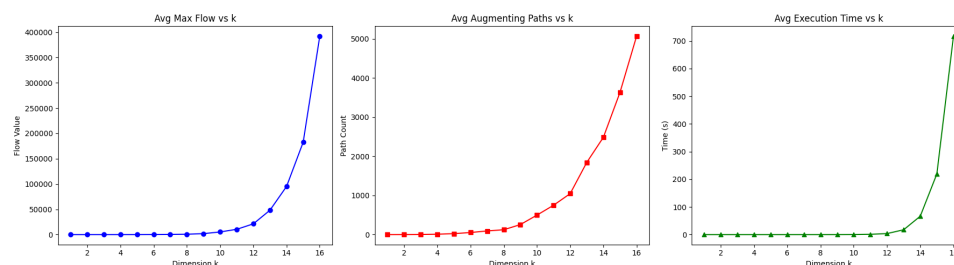
## 4 Wyniki eksperymentów

### 4.1 Zadanie 1: Przepływ w hipersześcianie

Przeprowadzono testy dla wymiarów  $k$  od 1 do 16. Porównano czas działania algorytmów Edmonds-Karp oraz SAP.



Rysunek 1: Porównanie czasu wykonania i liczby ścieżek powiększających dla algorytmów EK i SAP w funkcji wymiaru  $k$ .



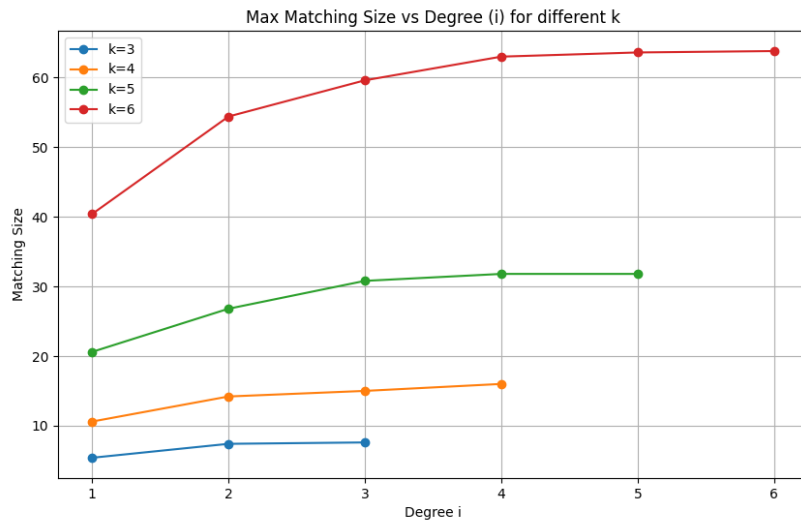
Rysunek 2: Porównanie czasu wykonania, flow value i liczby ścieżek powiększających dla algorytmu EK w funkcji wymiaru  $k$ .

### Wnioski:

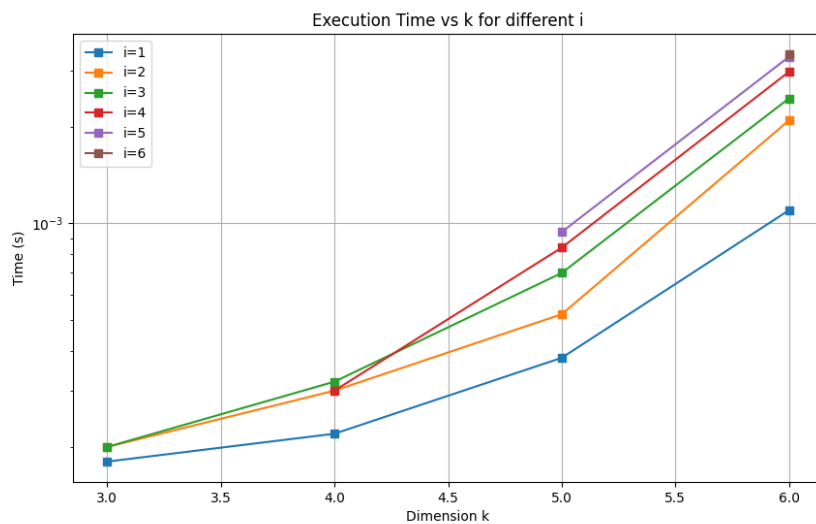
- Wraz ze wzrostem wymiaru  $k$ , liczba wierzchołków rośnie wykładniczo ( $2^k$ ), co drastycznie zwiększa czas obliczeń.
- Algorytm SAP wykazuje na ogół lepszą wydajność dla dużych grafów dzięki lepszemu zarządzaniu odległościami w grafie rezydualnym.
- Liczba znalezionych ścieżek powiększających jest skorelowana z wartością maksymalnego przepływu.

## 4.2 Zadanie 2: Skojarzenia w grafie dwudzielnym

Zbadano wpływ stopnia wierzchołka  $i$  na wielkość maksymalnego skojarzenia w losowym grafie dwudzielnym o  $2^k$  wierzchołkach po każdej stronie.



Rysunek 3: Rozmiar maksymalnego skojarzenia w zależności od stopnia  $i$  dla różnych  $k$ .



Rysunek 4: Czas wykonania algorytmu znajdowania skojarzenia w zależności od  $k$ .

#### Wnioski:

- Dla małych wartości  $i$  skojarzenie jest niepełne.
- Wraz ze wzrostem  $i$  (liczby krawędzi wychodzących z  $V_1$ ), rozmiar skojarzenia szybko dąży do  $2^k$  (skojarzenie doskonałe). Istnieje "próg", powyżej którego z dużym prawdopodobieństwem znajdujemy skojarzenie doskonałe.

### 4.3 Poprawność wyników (GLPK)

Dla małych instancji ( $k \leq 4$ ) wyniki własnych implementacji zostały zweryfikowane przy użyciu solvera GLPK (wczytującego wygenerowane modele LP).

- Wartości funkcji celu (Max Flow / Max Matching) były identyczne dla obu metod.

- Potwierdza to poprawność implementacji algorytmów grafowych oraz generatora modeli LP.

## 5 Podsumowanie

Eksperymenty potwierdziły teoretyczne założenia dotyczące złożoności obliczeniowej oraz pokazały elastyczność redukcji problemów (skojarzenia) do przepływów. Weryfikacja za pomocą programowania liniowego stanowi dowód poprawności zaimplementowanych rozwiązań.