

Metody Obliczeniowe w Nauce i Technice

Laboratorium 2 - Rozwiązywanie układów równań liniowych

Sprawozdanie

Jakub Pinowski

2020-03-09

1 Wstęp

Do rozwiązywania problemów zadanych na laboratorium użyty został Python 3.7 (w przypadku pierwszego i drugiego zadania w postaci Jupyter Notebook'a) z pakietami numpy, scipy, networkx oraz matplotlib.

2 Metoda Gaussa-Jordana

Pierwszym zadaniem było zaimplementowanie oraz przetestowanie funkcji rozwiązującej układ równań liniowych metodą Gaussa-Jordana.

2.1 Porównanie z `numpy.linalg.solve`

W porównaniu użyta została losowa macierz o rozmiarze 600 x 600 z wartościami z przedziału $[10, 500]$.

2.1.1 Wyniki

Największa różnica wyników podzielona przez wartość obliczoną przez numpy była rzędu 10^{-10} . Funkcje dały więc w praktyce takie same rezultaty.

2.1.2 Czas działania

- Gauss-Jordan: 94.77914547920227
- Numpy: 0.048226356506347656

Widzimy, że pod względem czasu działania zaimplementowany algorytm wypada bardzo słabo przy funkcji bibliotecznej

3 Faktoryzacja LU

Zaimplementowana funkcja wykonująca faktoryzację LU zwraca 3 macierze P, L oraz U. Dla wejściowej macierzy A prawdziwe jest równanie $PA = LU$. P to macierz powstała na skutek częściowego poszukiwania elementu wiodącego podczas wykonywania faktoryzacji i pozwala na ułożenie wierszy macierzy A w odpowiedniej kolejności, tak by równanie było prawdziwe i można było użyć macierzy L i U do rozwiązywania układu równań $Ax = B$.

Funkcja została przetestowana dla macierzy 300 x 300

3.1 Wyniki

Po wymnożeniu macierzy L przez U otrzymaliśmy macierz której elementy były odchyłone maksymalnie o wartości rzędu 10^{-7} od elementów macierzy PA, faktoryzacja przebiegła zatem pomyślnie.

3.2 Porównanie czasu z `scipy.linalg.lu`

- Moja faktoryzacja LU: 5.8890955448150635s
- SciPy LU: 0.0007963180541992188s

Widzimy więc ponownie wyższość funkcji bibliotecznych nad funkcją zaimplementowaną na potrzeby laboratorium.

4 Analiza obwodu elektrycznego

Analiza miała polegać na obliczeniu wartości natężenia prądu w poszczególnych częściach obwodu, reprezentowanego jako graf ważony (V, E) (waga krawędzi jest jej rezystancją).

4.1 Algorytm

Algorytm korzysta z praw Kirchhoff'a do wyznaczenia układu równań liniowych w których niewiadomymi są natężenia na poszczególnych krawędziach. Robi to w dwóch etapach.

4.1.1 I prawo Kirchhoff'a

Z pierwszego prawa Kirchhoff'a wiemy, że różnica sumy prądów wchodzących i sumy prądów wychodzących jest w każdym węźle równa zero. Pozwala nam to na zbudowanie $|V| - 1$ równań, ponieważ węzły pomiędzy którymi wpięta jest siła elektromotoryczna traktujemy jako jeden.

4.1.2 II prawo Kirchhoff'a

Drugie prawo Kirchhoff'a pozwala na skonstruowanie dodatkowych równań przy pomocy cykli w grafie. Wiemy, że sumując napięcia w takim cyklu (ze znakiem odpowiadającym kierunkowi płynącego prądu) otrzymamy wartość 0. Również tyczy się to cyklu z krawędzią zawierającą źródło SEM pomiędzy S i T , której nie uwzględniamy w grafie. Jednak, żeby nie wprowadzać redundacji w układzie równań musimy dobrać cykle w taki sposób, aby równania były liniowo niezależne. Można to osiągnąć poprzez wyznaczenie najmniejszego zbioru cykli budujących graf¹. Takie podejście daje nam $|E| - |V| + 1$ niezależnych cykli, dodając do tego dowolny cykl przechodzący przez źródło SEM dostajemy $|E| - |V|$ równań. Dodatkowe równanie jest potrzebne, żeby wprowadzić do układu informację o wartości przyłożonej SEM.

W ten sposób uzyskaliśmy $|E| + 1$ równań w układzie z $|E|$ niewiadomymi. Rozwiązanie takiego równania z niekwadratową macierzą, przy pomocy metody najmniejszych kwadratów, dostarcza nam funkcja `scipy.linalg.lstsq(A, B)`.

4.2 Wizualizacja

Do wizualizacji otrzymanych rozwiązań użyte zostały funkcje `networkx` używające biblioteki `matplotlib`. Kierunki krawędzi reprezentują kierunek przepływu prądu, grubość uzależniona jest od rezystancji (grubsza krawędź = mniejszy opór), a kolor pokazuje wartość natężenia prądu w danym fragmencie (ciemniejsza krawędź = większy prąd). Dodatkowo wierzchołki ustawiają się w zależności od ilości połączeń i natężenia prądu na krawędziach. Im więcej krawędzi do danego wierzchołka i im mniejsze natężenia prądu na tych krawędziach, tym bliżej wierzchołek jest danej grupy wierzchołków, oznacza to grupowanie się wierzchołków między którymi płynie mały prąd.

4.3 Weryfikacja poprawności

Dodatkowo zaimplementowana została funkcja sprawdzająca poprawność rozwiązania. Test polega na sprawdzeniu I prawa Kirchhoff'a dla wszystkich wierzchołków oraz na sprawdzeniu II prawa Kirchhoff'a dla kilku ścieżek od S do T o rozłącznych zbiorach krawędzi (suma spadków napięć powinna być równa SEM).

4.4 Program

Skrypty zaimplementowane na potrzeby zadania

- `circuit_main.py` - główny program, przyjmuje jako argumenty nazwy plików z zapisanymi grafami

¹Jest to podział na takie cykle, że każdy zawiera krawędź, która nie występuje w żadnym innym cyklu w tym podziale. W bibliotece `networkx` realizuje go funkcja `networkx.cycle_basis(Graph)`

- circuit_analyzer.py - główny algorytm rozwiązujący problem zadanego obwodu
- circuit_generator.py - funkcje do generowania przykładowych grafów i zapisywania ich do plików + wygenerowanie przykładowych grafów
- circuit_plotter.py - funkcja do wyświetlania grafu z rozwiązaniem
- circuit_check.py - funkcja testująca rozwiązanie
- circuit_reader.py - funkcja czytająca graf z pliku (każda linia: wierzchołek_1 wierzchołek_2 rezystancja; linia rozpoczynająca się przez STU zawiera wierzchołek_S wierzchołek_T SEM)

4.5 Przykładowe wyniki działania programu

Każdy podany graf przeszedł pomyślnie test zawarty w pliku circuit_check.py

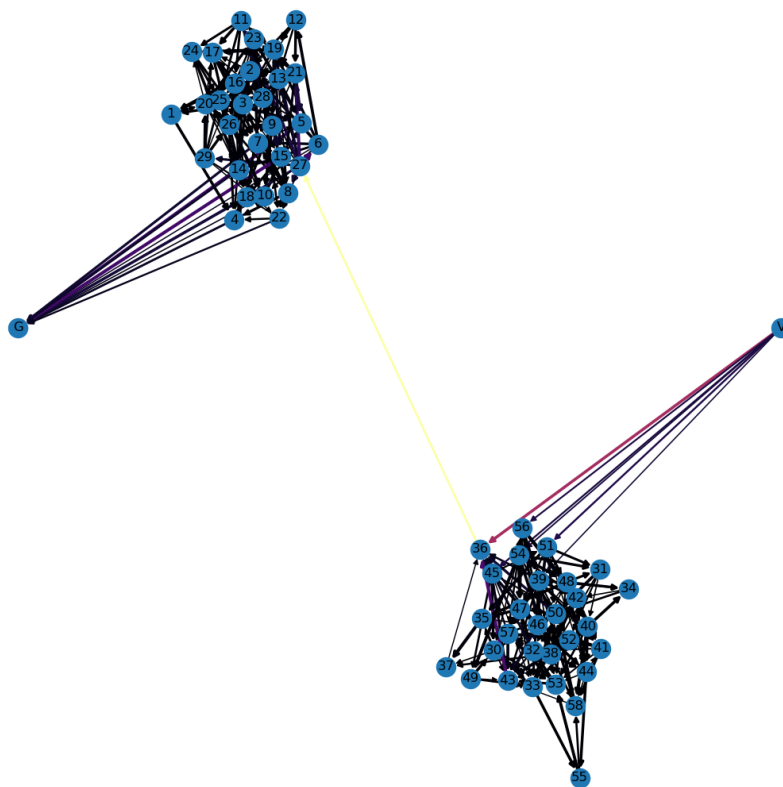


Figure 1: Dwa spójne grafy połączone mostkiem

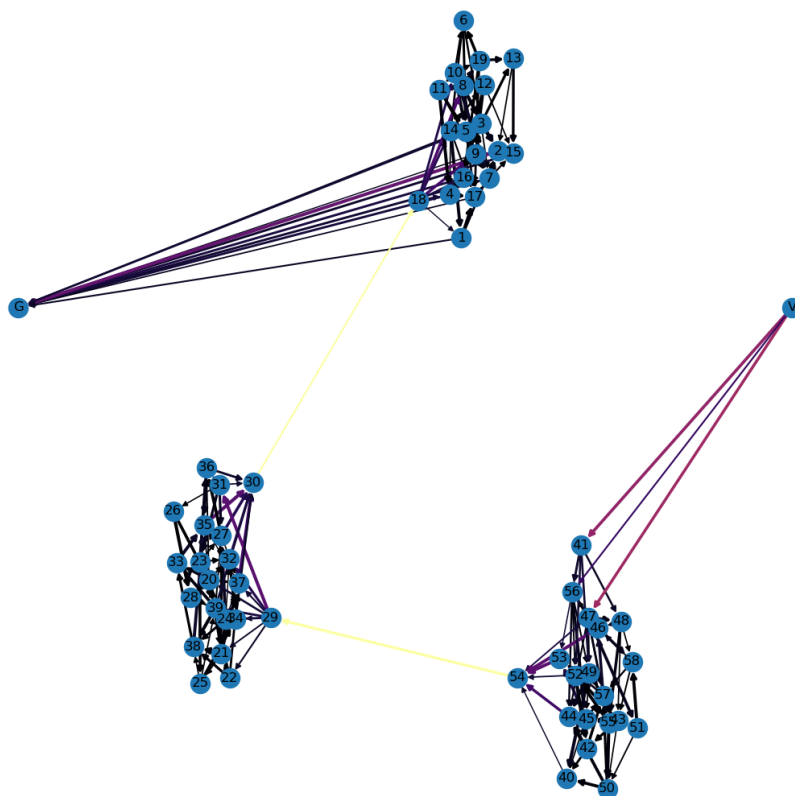


Figure 2: Trzy spójne grafy połączone mostkami

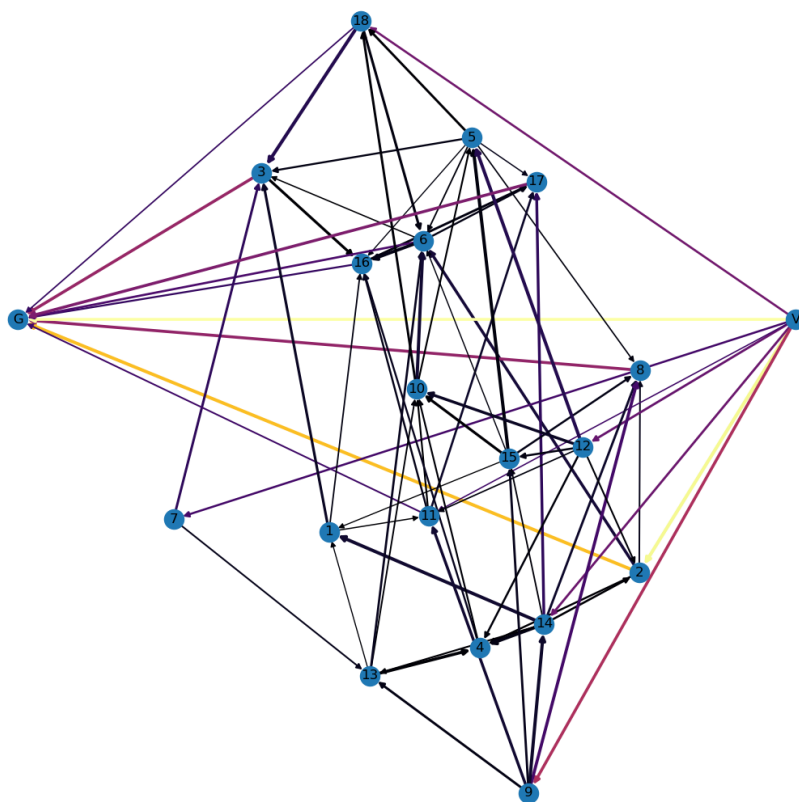


Figure 3: Graf spójny o 20 wierzchołkach

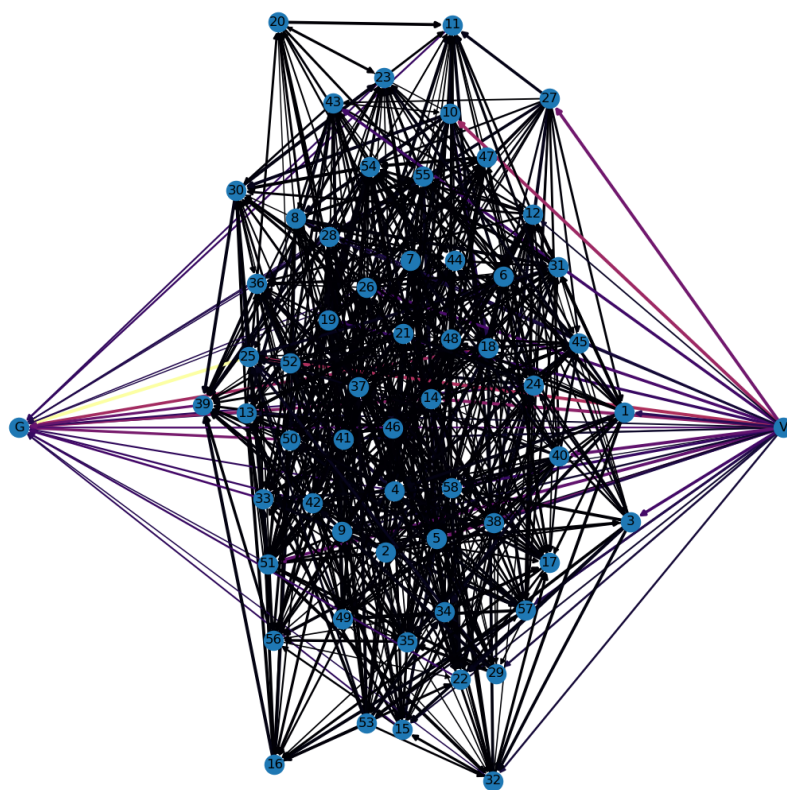


Figure 4: Graf spójny o 60 wierzchołkach

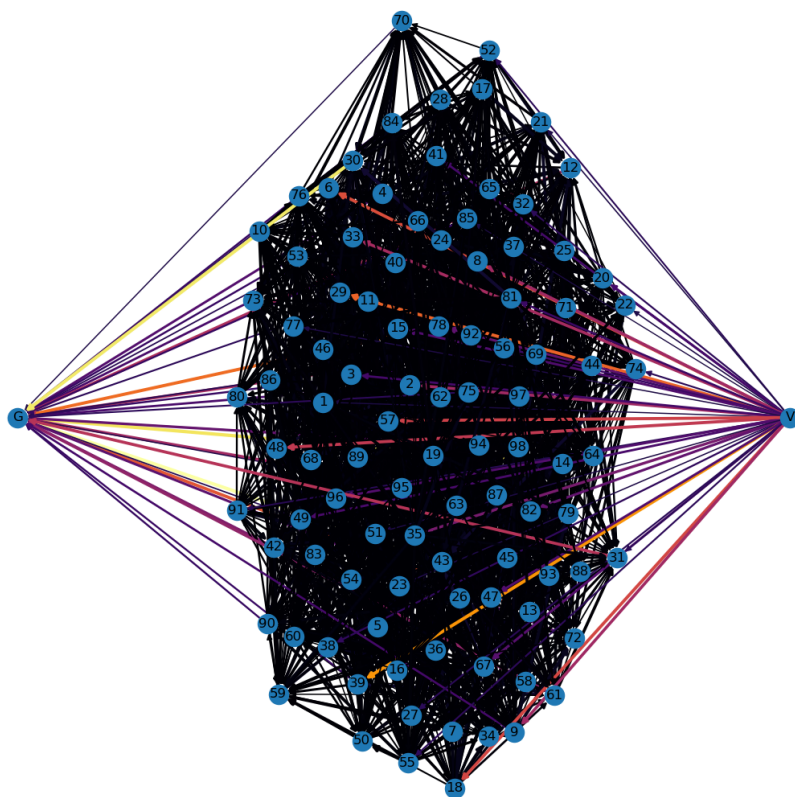


Figure 5: Graf spójny o 100 wierzchołkach

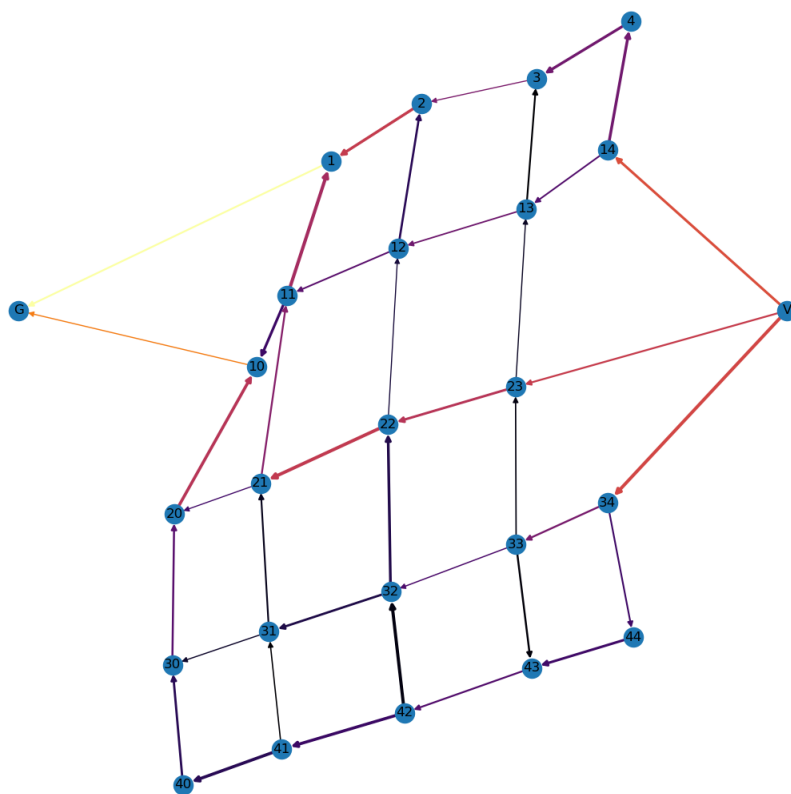


Figure 6: Siatka 2D 5 x 5

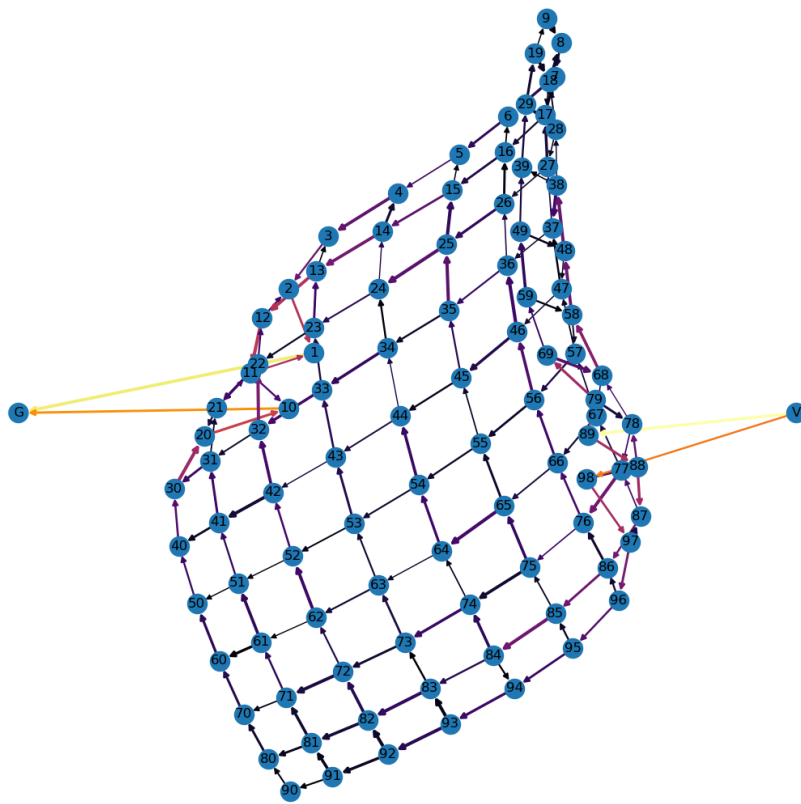


Figure 7: Siatka 2D 10 x 10

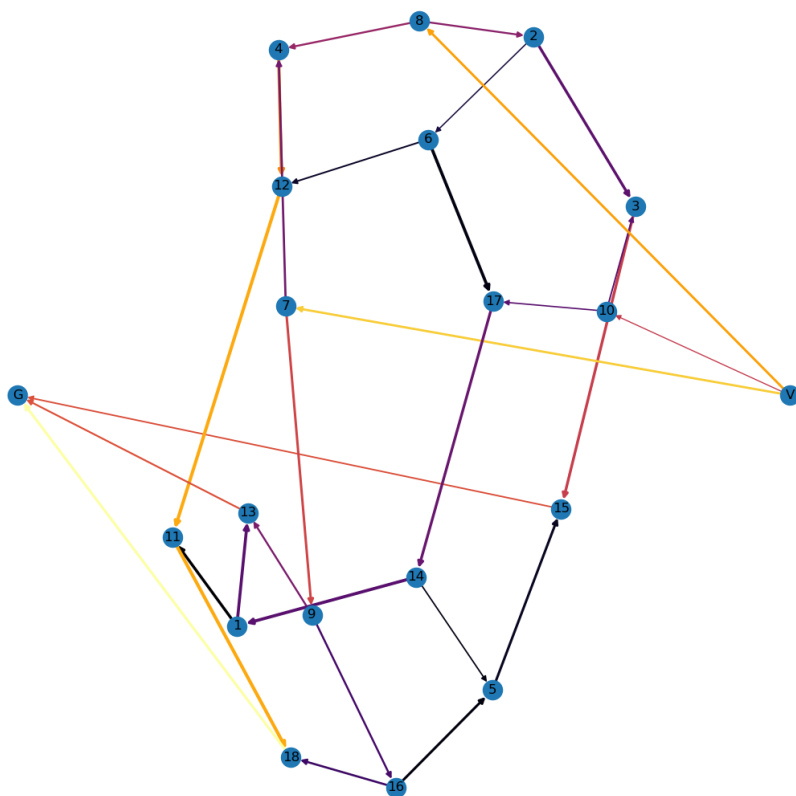


Figure 8: Graf kubiczny o 20 wierzchołkach

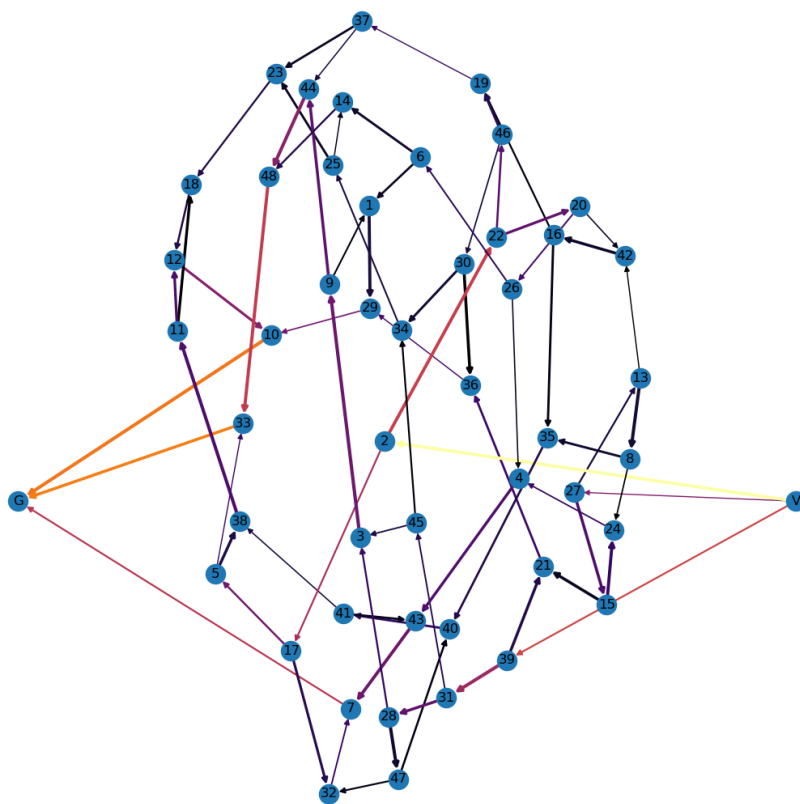


Figure 9: Graf kubiczny o 50 wierzchołkach

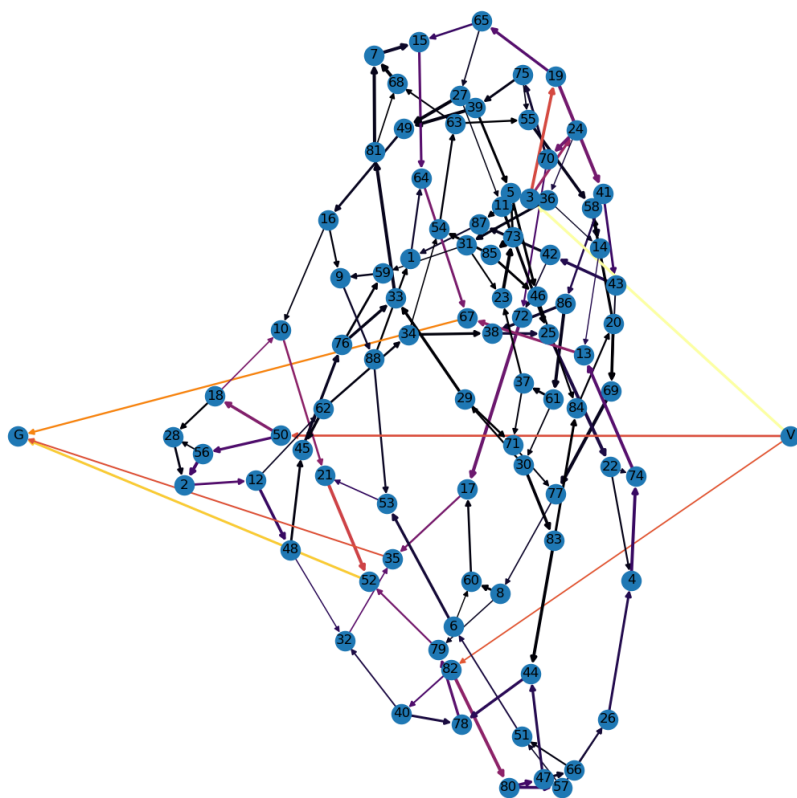


Figure 10: Graf kubiczny o 90 wierzchołkach