

# Projektowanie Algorytmów i Metody Sztucznej Inteligencji

## Grafy

Prowadzący: Mgr inż. Marcin Ochman

Autor: Adrian Sobecki

Numer indeksu: 248942

Termin zajęć: Wtorek 15:15-16:55

Termin oddania: 05.05.2020

## Wstęp

Naszym zadaniem była implementacja grafu za pomocą macierzy sąsiedztwa oraz listy sąsiedztwa. Następnie dla naszych reprezentacji należało napisać jeden algorytm rozwiązujący problem najkrótszej ścieżki w grafie od wybranego wierzchołka do każdego innego. Podjąłem decyzję o realizacji algorytmu Dijkstry. Ostatnim etapem były testy efektywności i opracowanie otrzymanych wyników.

## Opis reprezentacji grafu

### a) Macierz sąsiedztwa

Reprezentacja grafu wykorzystująca macierz  $n \times n$  ( $n$  – ilość wierzchołków grafu) nazywaną macierzą sąsiedztwa. Dla każdego wierzchołka (wiersza) mamy pełny obraz relacji pomiędzy każdym pozostałym wierzchołkiem tego grafu (kolumny). W zależności od naszej implementacji w macierzy możemy przechowywać informacje o długości krawędzi (wadze), a w przypadku braku połączenia wartość np. -1. Mamy wtedy szybki dostęp do wagi krawędzi pomiędzy dwoma wierzchołkami. Nakłady pamięciowe nie zależą od gęstości grafu, zawsze przechowujemy macierz  $n \times n$ .

### b) Lista sąsiedztwa

Reprezentacja grafu wykorzystująca tablice o wymiarze  $n$  ( $n$  – ilość wierzchołków grafu) list/tablic przechowujących pary liczb (index wierzchołka i waga krawędzi). Dla każdego wierzchołka grafu przechowujemy wszystkich jego sąsiadów (pary liczb) na tablicy/liście odpowiadającej indexowi naszego wierzchołka. Reprezentacja ta wymaga liniowego przeszukiwania sąsiadów w celu znalezienia wagi pomiędzy dwoma wierzchołkami. Nakłady pamięciowe zależą od ilości wierzchołków grafu jak i jego gęstości.

## Algorytm Dijkstry

Algorytm znajdujący najkrótszą ścieżkę pomiędzy wybranym wierzchołkiem a każdym pozostałym, lub również pomiędzy dwoma wybranymi wierzchołkami (zakończenie programu w momencie znalezienia najkrótszej ścieżki, bez wyznaczenia dla pozostałych wierzchołków). Tworzymy tablicę  $n$  wymiarową, którą inicjalizujemy wartościami nieskończoność (wagi dojsć). Zaczynamy od wybranego wierzchołka. Przechodzimy przez jego sąsiadów dodając do kolejki priorytetowej wartości krawędzi łączących oba wierzchołki (o ile dany wierzchołek sąsiedni nie został już wcześniej dodany) oraz zmieniając wagi dojsć (o ile są mniejsze). Następnie przechodzimy do wierzchołka, dla którego waga przejścia jest najmniejsza i całą czynność powtarzamy, aż sprawdzimy wszystkie wierzchołki (kolejka priorytetowa będzie pusta). Pesymistyczna złożoność wynosi  $O(E + V \log V)$ .

Po wcześniejszym sprawdzeniu poprawności działania metody do generowania grafów oraz algorytmu Dijkstry nadszedł czas na sprawdzenie efektywności. W tym celu napisałem specjalną funkcję przyjmującą 3 argumenty, które pozwalają na realizację każdego badanego przypadku. Testy przebiegły bezproblemowo.

Uzyskane wyniki dla 100 grafów każdego typu

a) 10 wierzchołków

Reprezentacja grafu	Gęstość grafu			
	25%	50%	75%	100%
lista	0,000997s	0,000999s	0,002030s	0,001994s
macierz	<0,000001s	0,001994s	0,002019s	0,001997s

b) 50 wierzchołków

Reprezentacja grafu	Gęstość grafu			
	25%	50%	75%	100%
lista	0,012005s	0,017950s	0,022968s	0,028950s
macierz	0,011999s	0,015988s	0,017978s	0,036917s

c) 100 wierzchołków

Reprezentacja grafu	Gęstość grafu			
	25%	50%	75%	100%
lista	0,036927s	0,060848s	0,088793s	0,120704s
macierz	0,035903s	0,044855s	0,049869s	0,057866s

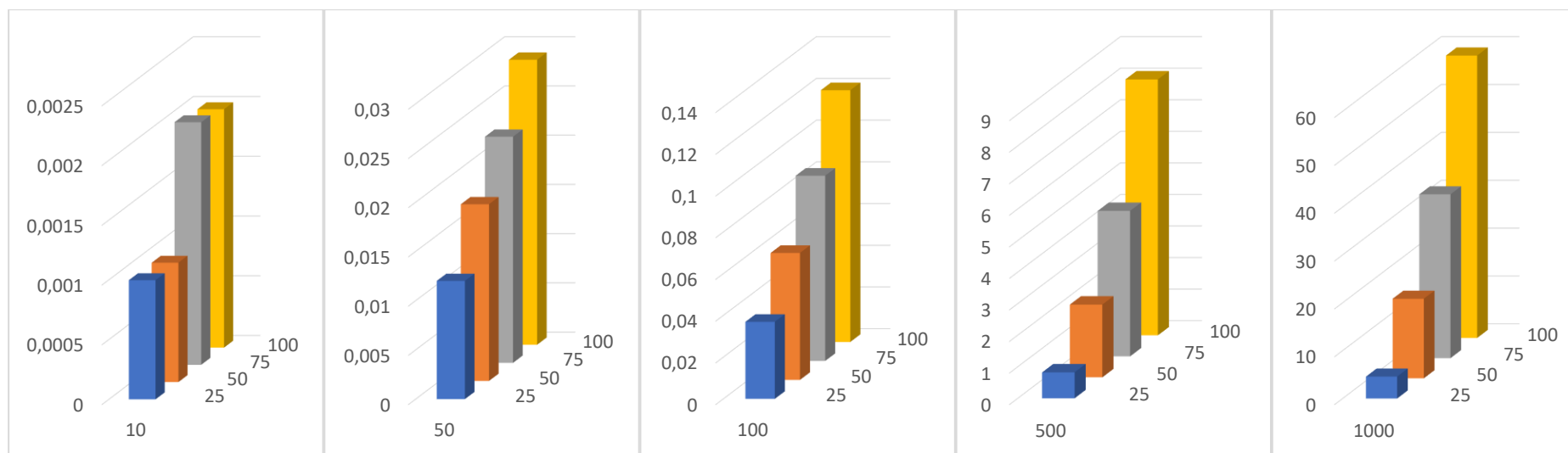
d) 500 wierzchołków

Reprezentacja grafu	Gęstość grafu			
	25%	50%	75%	100%
lista	0,820837s	2,30385s	4,60771s	8,10836s
macierz	0,509651s	0,613360s	0,813854s	0,947494s

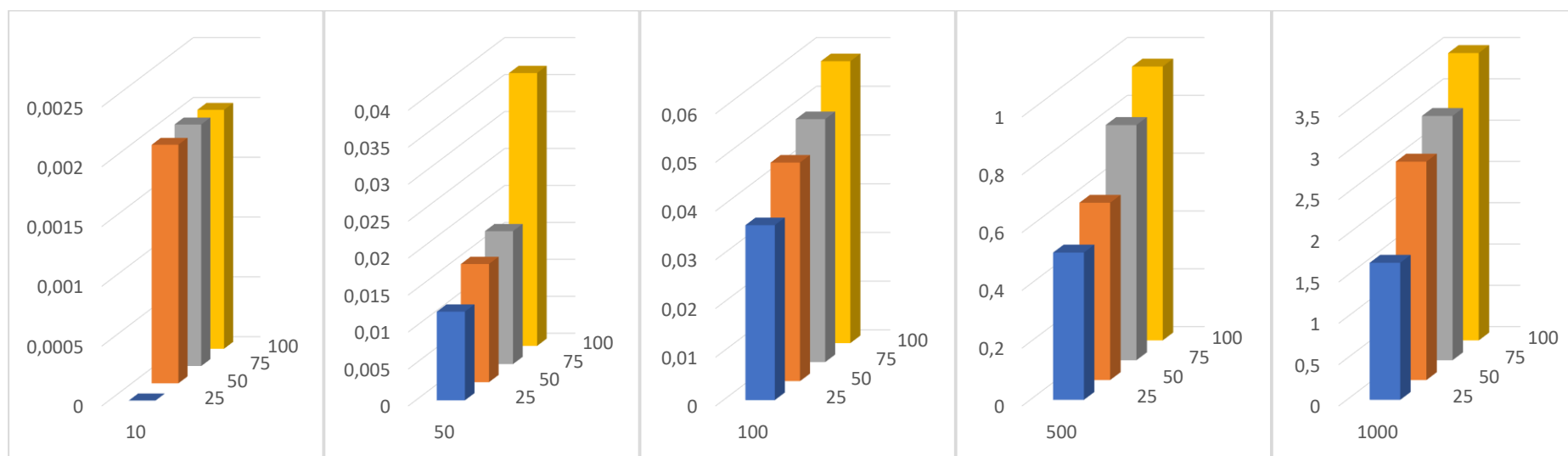
e) 1000 wierzchołków

Reprezentacja grafu	Gęstość grafu			
	25%	50%	75%	100%
lista	4,57876s	16,6236s	34,2206s	58,9944s
macierz	1,65959s	2,64493s	2,95809s	3,47973s

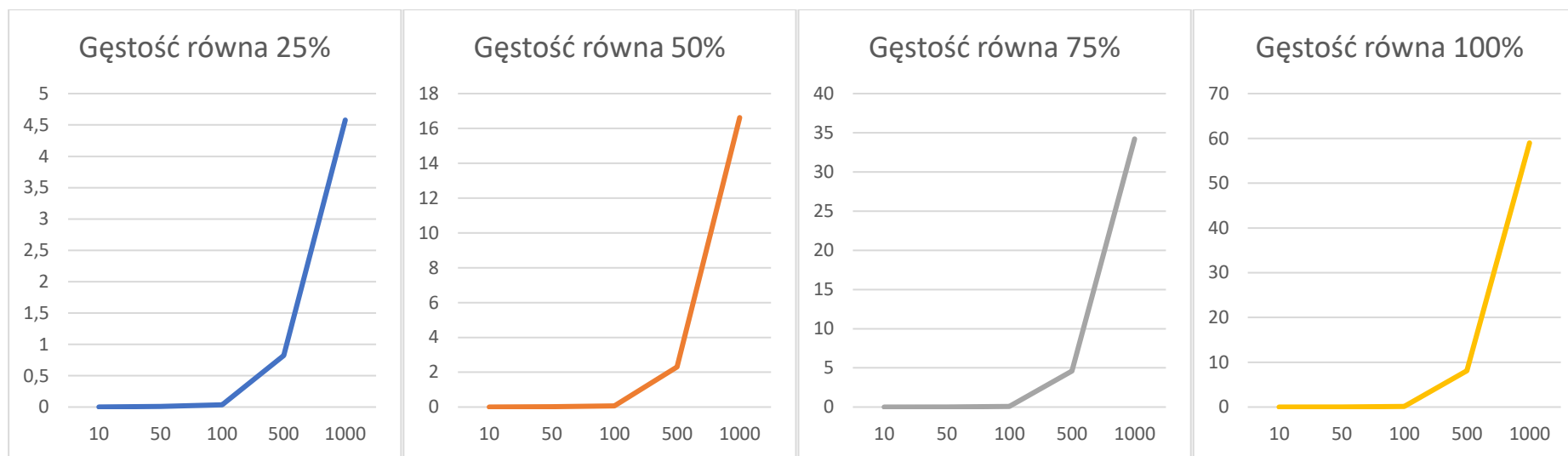
Reprezentacja grafu w postaci listy (wysokość – czas [s], głębokość – gęstość [%])



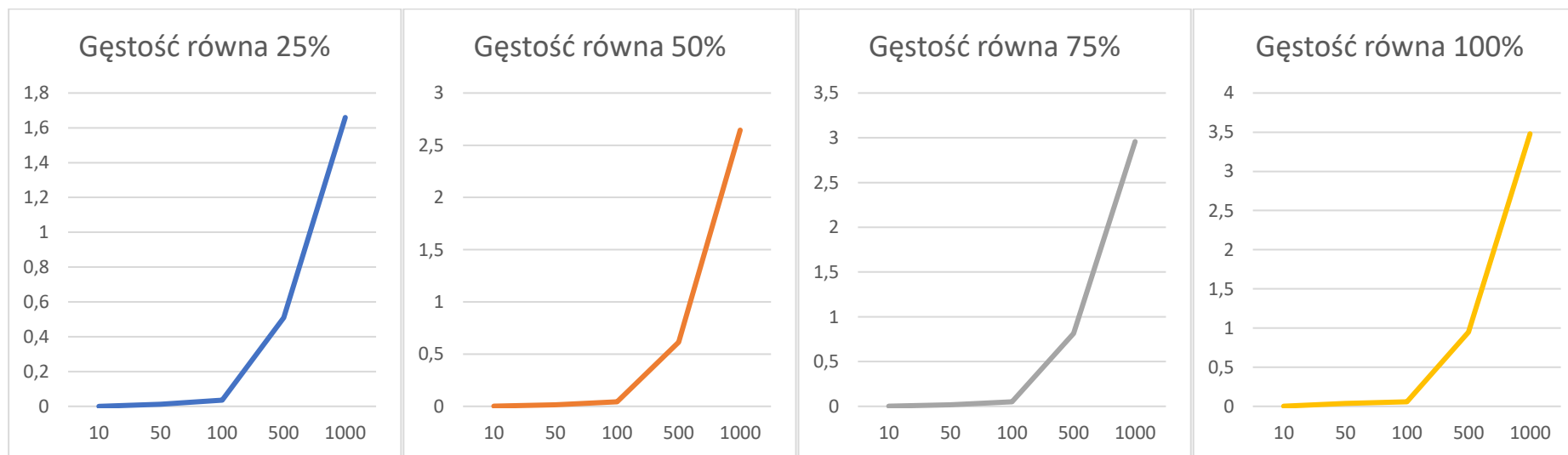
Reprezentacja grafu w postaci macierzy (wysokość – czas [s], głębokość – gęstość [%])



Reprezentacja grafu w postaci listy (wykresy czasu [s] od ilości wierzchołków)



Reprezentacja grafu w postaci macierzy (wykresy czasu [s] od ilości wierzchołków)



## Wnioski

Algorytm Dijkstry dla grafu reprezentowanego jako macierz sąsiedztwa uległ grafowi reprezentowanego jako lista sąsiedztwa tylko dla liczby wierzchołków równej 10. Wraz ze wzrostem liczby wierzchołków różnicę pomiędzy dwoma reprezentacjami rosną. Powodem wyższości reprezentacji macierzowej jest bezpośrednie zwracanie wartości wagi krawędzi pomiędzy dwoma wierzchołkami, gdzie w reprezentacji liniowej musimy za każdym razem przeszukiwać cały zbiór wierzchołków sąsiednich. Z pierwszej serii wykresów można wywnioskować, że reprezentacja za pomocą listy sąsiedztwa radzi sobie gorzej ze wzrostem gęstości grafu. Natomiast z drugiej, że czas potrzebny na wykonanie algorytmu rośnie wraz ze wzrostem liczby wierzchołków w ten sam sposób nie zależnie od reprezentacji oraz gęstości.

## Literatura

[https://eduinf.waw.pl/inf/alg/001\\_search/0122.php](https://eduinf.waw.pl/inf/alg/001_search/0122.php)