

# Sprawozdanie

## Projekt 3 – Problem najkrótszej drogi

Wojciech Konury

### Wprowadzenie

W projekcie należało zbadać czas wykonywania dwóch różnych algorytmów rozwiązujących problem znalezienia najkrótszej drogi (ścieżki) w grafie między dwoma wierzchołkami. Algorytmy badano dla dwóch reprezentacji grafów w programie oraz dla różnych wypełnień grafów.

#### Algorytmy:

- Dijkstra
- Bellman-Ford (możliwe ujemne wagi połączeń)

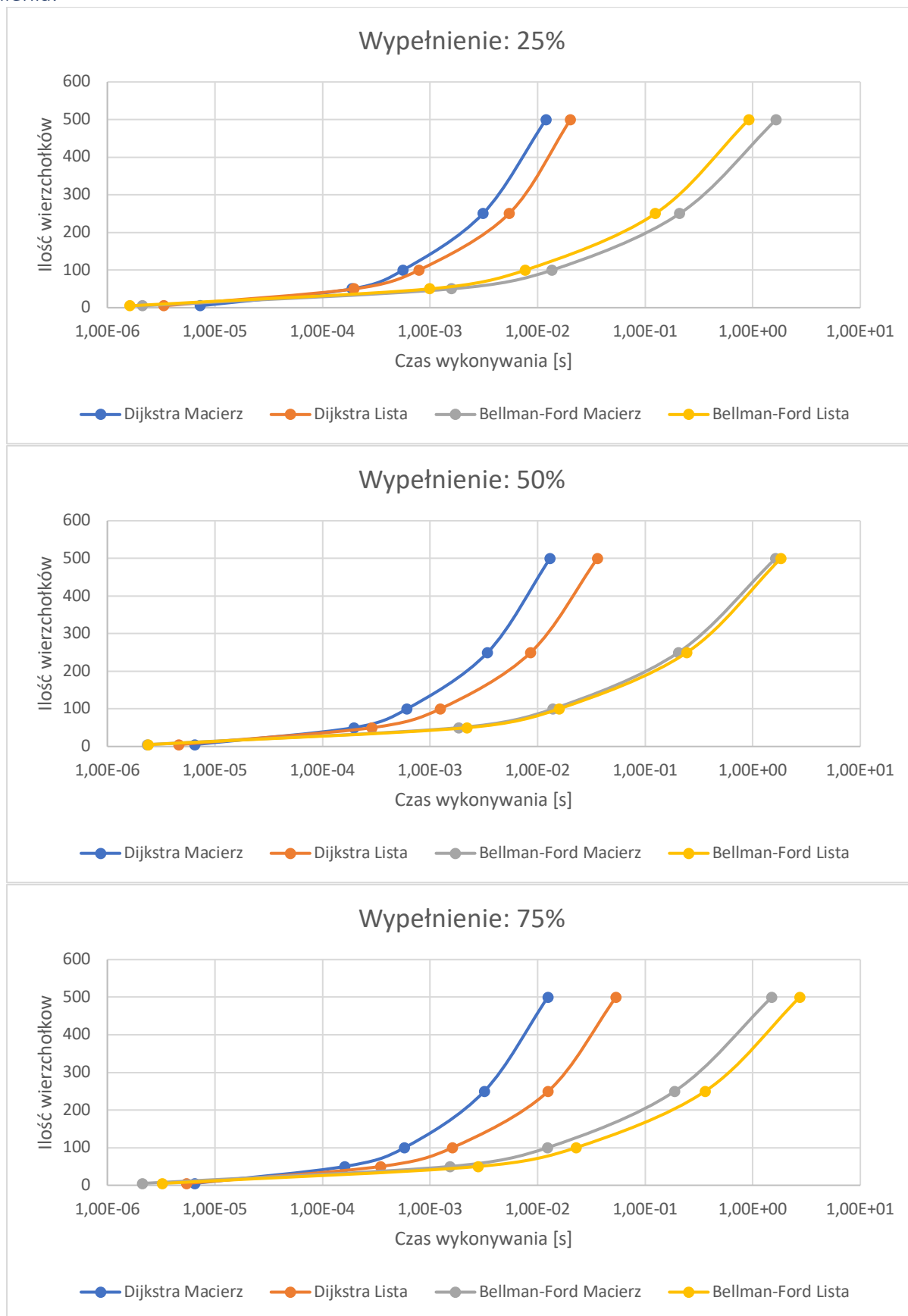
#### Reprezentacje grafów:

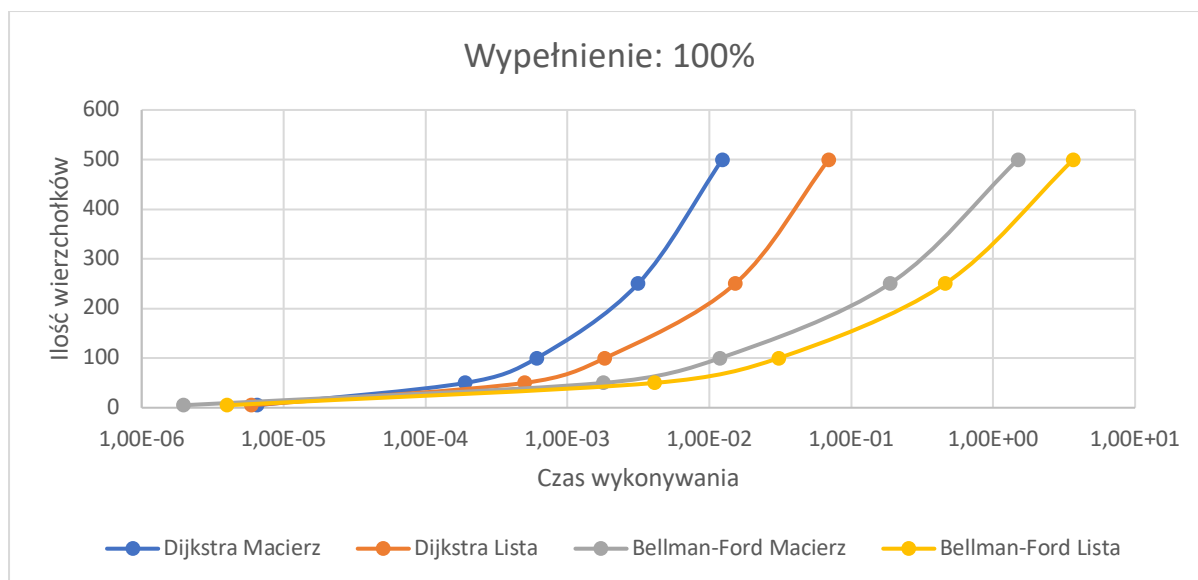
- Macierz sąsiedztwa
- Lista sąsiadów

Wyniki pomiarów czasu wykonywania algorytmów w zależności od reprezentacji grafu oraz stopnia wypełnienia grafu

Dijkstra Macierz		Dijkstra Lista		Bellman-Ford Macierz		Bellman-Ford List	
Wierzchołki	Średni czas [s]	Wierzchołki	Średni czas [s]	Wierzchołki	Średni czas [s]	Wierzchołki	Średni czas [s]
25%							
5	7.26E-06	5	3.36E-06	5	2.12E-06	5	1.62E-06
50	0.00018803	50	0.00019683	50	0.0015871	50	0.000993521
100	0.000560914	100	0.000789972	100	0.0136322	100	0.00770881
250	0.00310136	250	0.0054229	250	0.209129	250	0.124083
500	0.0119504	500	0.0202147	500	1.65096	500	0.92015
50%							
5	6.47E-06	5	4.64E-06	5	2.36E-06	5	2.41E-06
50	0.000195322	50	0.000288539	50	0.00184794	50	0.00220523
100	0.000611525	100	0.00124398	100	0.0139255	100	0.0158149
250	0.00342917	250	0.00862509	250	0.203779	250	0.244584
500	0.0130453	500	0.0359903	500	1.63044	500	1.82653
75%							
5	6.47E-06	5	5.48E-06	5	2.12E-06	5	3.24E-06
50	0.000160442	50	0.000346745	50	0.00153584	50	0.0027994
100	0.000579607	100	0.00161419	100	0.0123506	100	0.0227296
250	0.00319391	250	0.0125467	250	0.188027	250	0.361226
500	0.0124364	500	0.0535358	500	1.50438	500	2.73603
100%							
5	6.49E-06	5	5.92E-06	5	1.97E-06	5	4.01E-06
50	0.000190648	50	0.000497353	50	0.00179752	50	0.00411187
100	0.000607609	100	0.00183699	100	0.0118909	100	0.030739
250	0.00314792	250	0.0152151	250	0.18703	250	0.45861
500	0.0123249	500	0.0693943	500	1.49371	500	3.67175

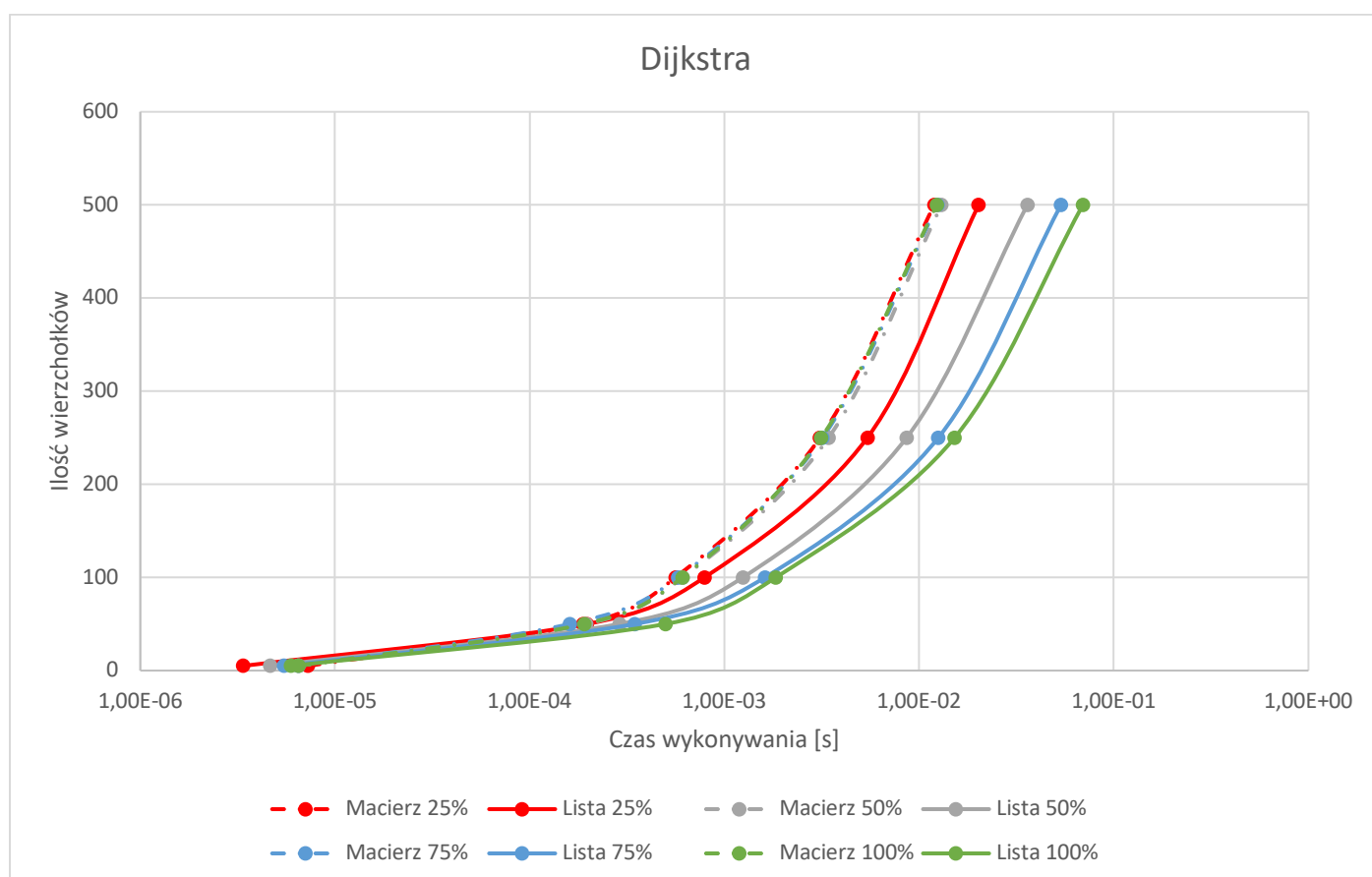
Porównanie czasu wykonywania w zależności od algorytmu i reprezentacji grafu dla różnych stopni wypełnienia:





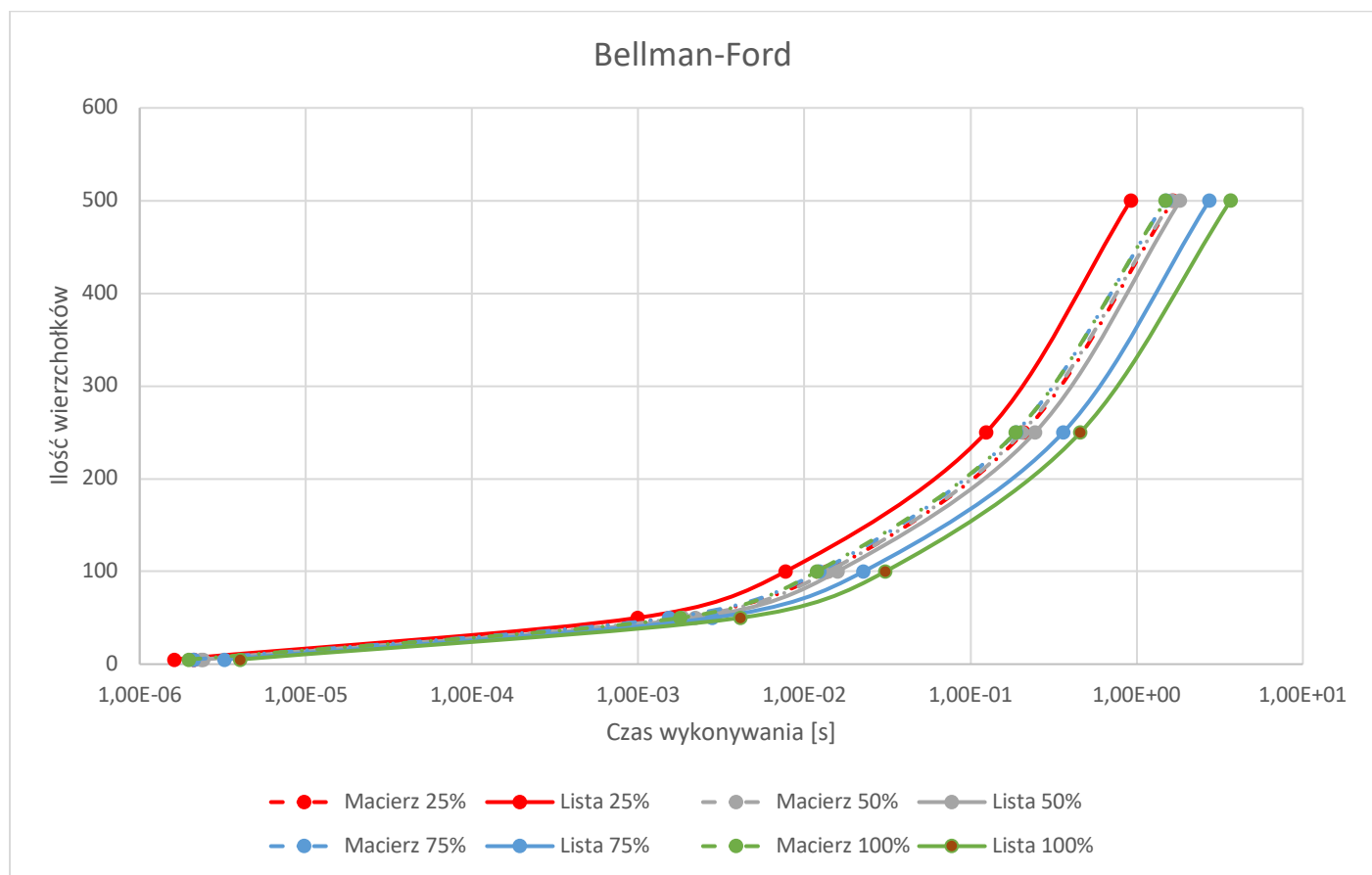
Dla wszystkich stopni wypełnienia grafów algorytm Dijkstry jest szybszy od algorytmu Bellmana-Forda

Porównanie czasu wykonywania algorytmu Dijkstry w zależności od stopnia wypełnienia i reprezentacji grafu:



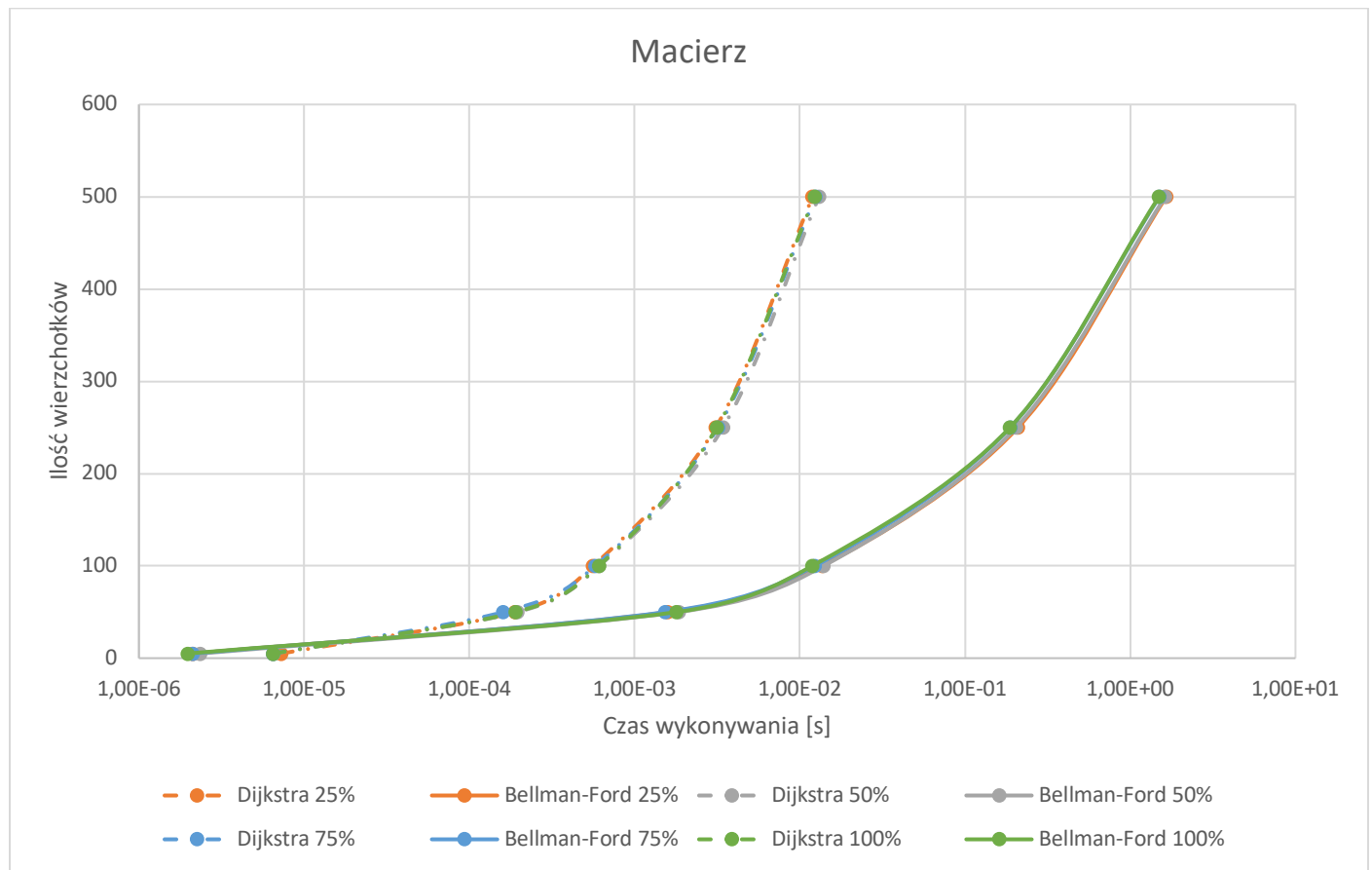
Dla algorytmu Dijkstra najszybszą implementacją grafu jest macierz, która wykonuje się niezależnie od stopnia wypełnienia. Czas wykonywania algorytmu, gdzie graf jest w formie listy zależy od stopnia wypełnienia grafu.

Porównanie czasu wykonywania algorytmu Bellmana-Forda w zależności od stopnia wypełnienia i reprezentacji grafu:



Również jak w algorytmie Dijkstry stopień wypełnienia grafu w formie macierzy nie wpływa na czas wykonywania się algorytmu.

Porównanie czasu wykonywania algorytmu Dijkstry i Bellmana Forda w reprezentacji grafu jako macierzy sąsiedztwa w zależności od stopnia wypełnienia:



Porównanie czasu wykonywania algorytmu Dijkstry i Bellmana Forda w reprezentacji grafu jako listy sąsiadów w zależności od stopnia wypełnienia:

