

Projektowanie algorytmów i metody sztucznej inteligencji

Projekt 3 - Grafy

22.04.2021, godz. 15:15

Aleksandra Rozmus 252954

Spis treści

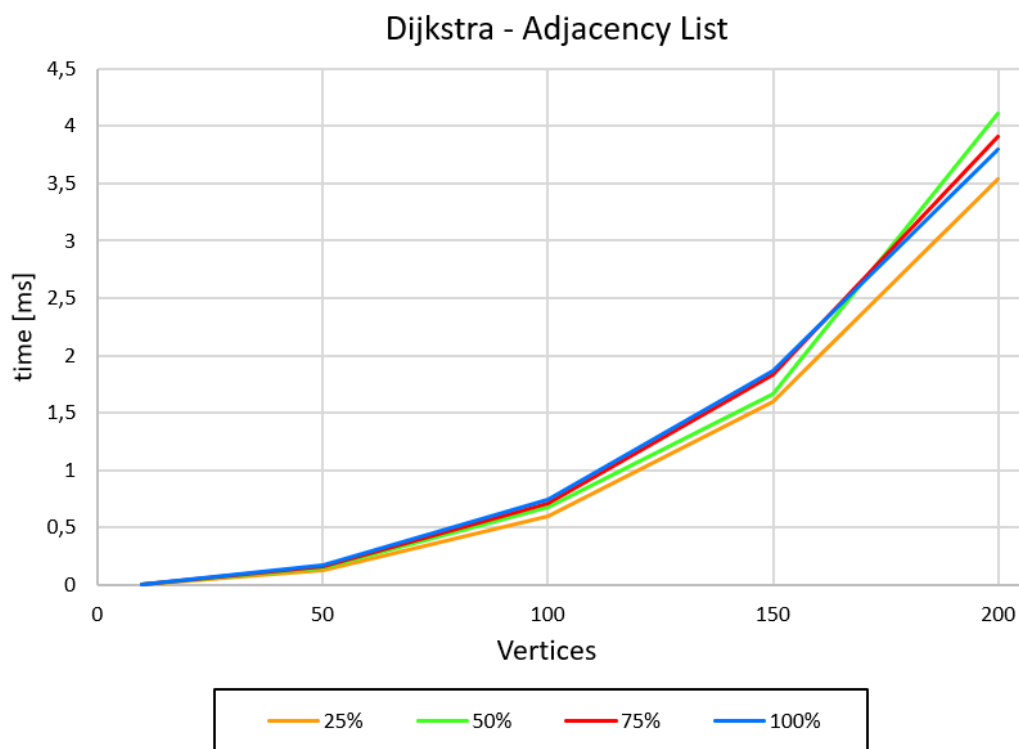
1	Pomiary	2
2	Wygenerowane wykresy	3
3	Wnioski	5

1 Pomiar

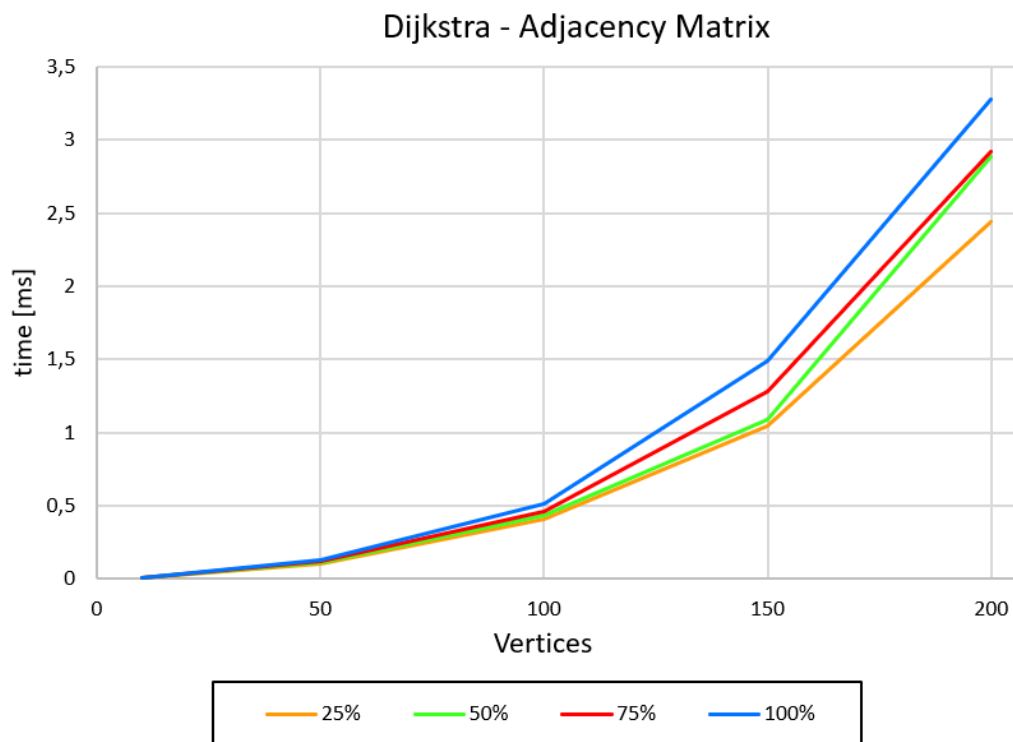
Tablica 1: Uśrednione wyniki pomiarów dla poszczególnych algorytmów

		Dijkstra -List	Dijkstra -Matrix	Bellman Ford-List	Bellman Ford -Matrix
Vertices	Density	time [ms]			
10	25%	0,003474	0,002828	0,00387	0,004248
50		0,129712	0,10341	1,0779	1,008936
100		0,595214	0,408286	13,15632	13,09406
150		1,592284	1,048208	50,55789	50,47741
200		3,538224	2,443352	131,3564	130,8298
10	50%	0,004142	0,004234	0,00872	0,0073
50		0,145894	0,110196	2,465212	2,470472
100		0,672486	0,429398	25,58902	25,55724
150		1,660032	1,083186	93,45741	93,55615
200		4,1063	2,88446	247,5379	249,5225
10	75%	0,005714	0,005034	0,01246	0,011498
50		0,158402	0,116096	3,79664	3,731802
100		0,711276	0,458692	37,83001	37,92254
150		1,82723	1,277806	135,2425	135,3841
200		3,908292	2,922138	342,8925	345,0482
10	100%	0,006376	0,005178	0,014854	0,014942
50		0,170992	0,128148	4,937506	4,862224
100		0,739966	0,509678	45,17755	45,04288
150		1,871414	1,491996	157,8504	157,8599
200		3,80003	3,274886	391,5704	390,5031

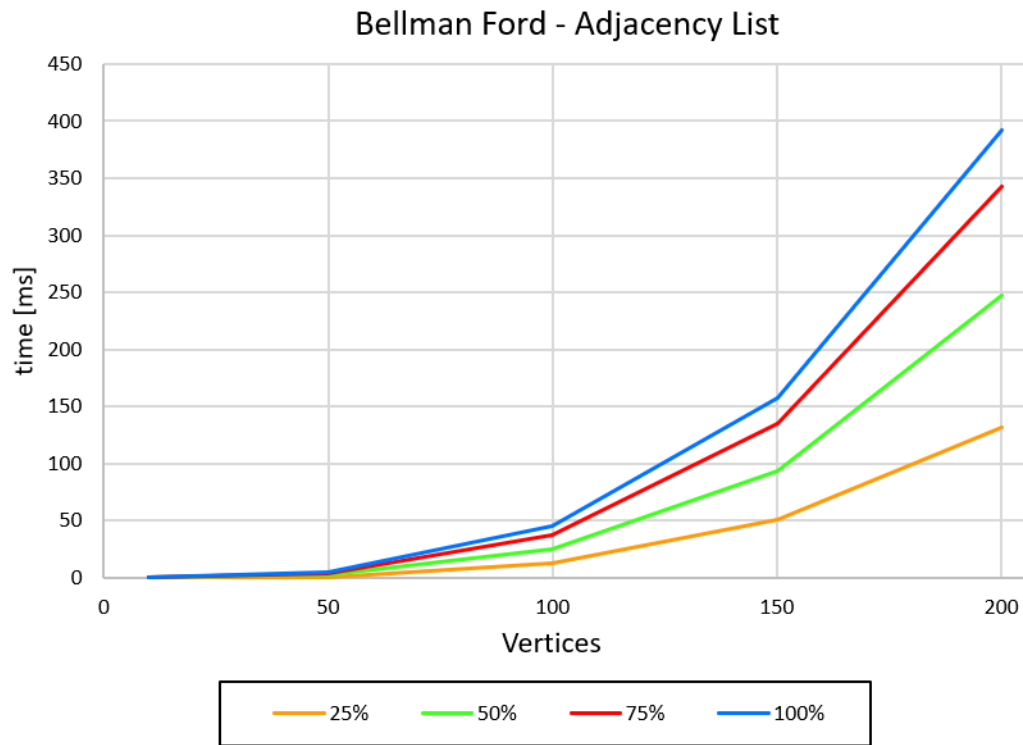
2 Wygenerowane wykresy



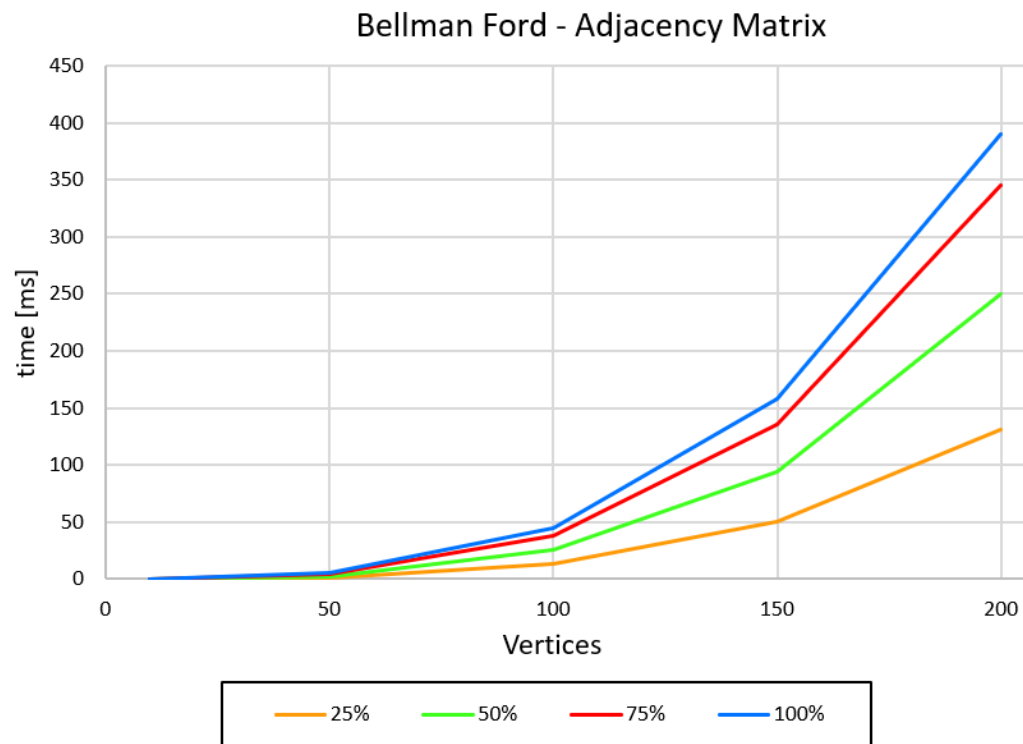
Rysunek 1: Czas działania algorytmu Dijkstry na liście sąsiedztwa



Rysunek 2: Czas działania algorytmu Dijkstry na macierzy sąsiedztwa



Rysunek 3: Czas działania algorytmu Bellmana-Forda na liście sąsiedztwa



Rysunek 4: Czas działania algorytmu Bellmana-Forda na macierzy sąsiedztwa

3 Wnioski

- Algorytm Bellmana Forda jest wolniejszy, jednak można stosować go w przypadku występowania ujemnych kosztów.
- Wraz ze wzrostem ilości krawędzi oraz gęstości wzrasta czas trwania algorytmu.
- Algorytm Dijkstry przeprowadzony na macierzy sąsiedztwa okazał się trochę szybszy niż na liście sąsiedztwa.
- Wyniki pomiarów algorytmu Bellmana Forda dla macierzy sąsiedztwa i listy sąsiedztwa okazały się zbliżone do siebie.