

# Projektowanie algorytmów i metody sztucznej inteligencji

## Sprawozdanie - Grafy

### 1. Procedura testowa

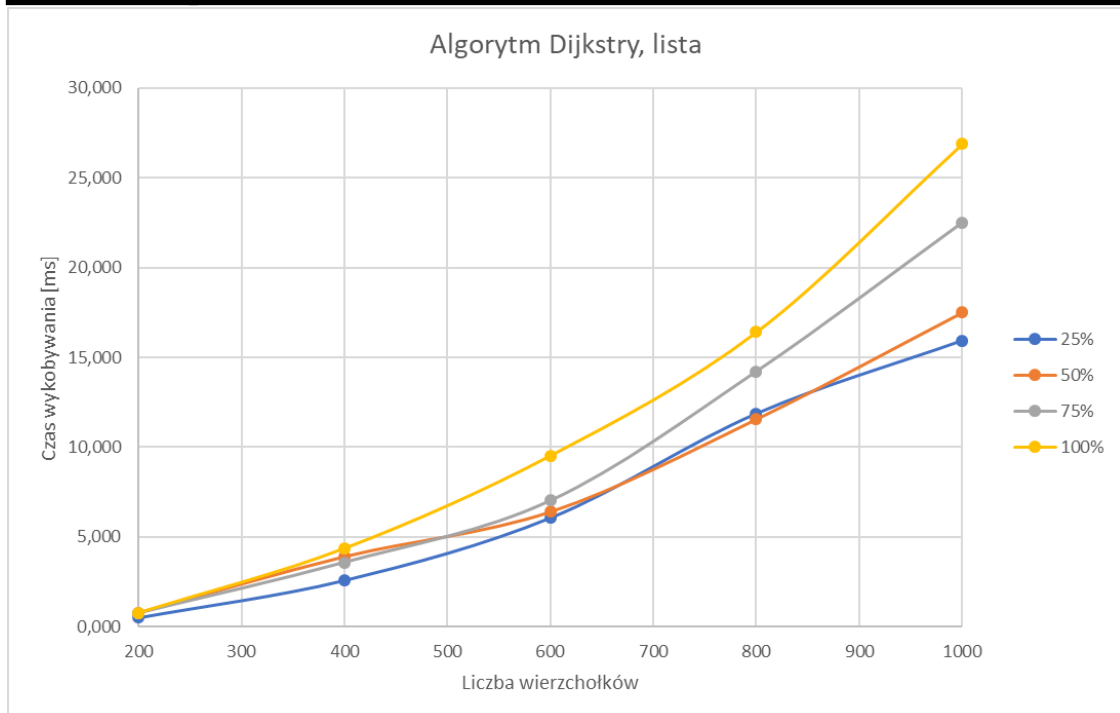
Wygenerowano cztery zestawy danych zawierających grafy o gęstościach 25%, 50%, 75% oraz 100%. Każdy zestaw składał się z 100 wygenerowanych (pseudo)losowo grafów o liczbach wierzchołków; 200, 400, 600, 800 oraz 1000.

Otrzymane zestawy przepuszczono przez program wyznaczający najkrótszą drogę dojścia od wierzchołka 0 do wszystkich innych wierzchołków grafu, wykorzystującego algorytm Dijkstry oraz Bellmana-Forda.

Wyniki dla danej liczby wierzchołków sumowano i uśredniano.

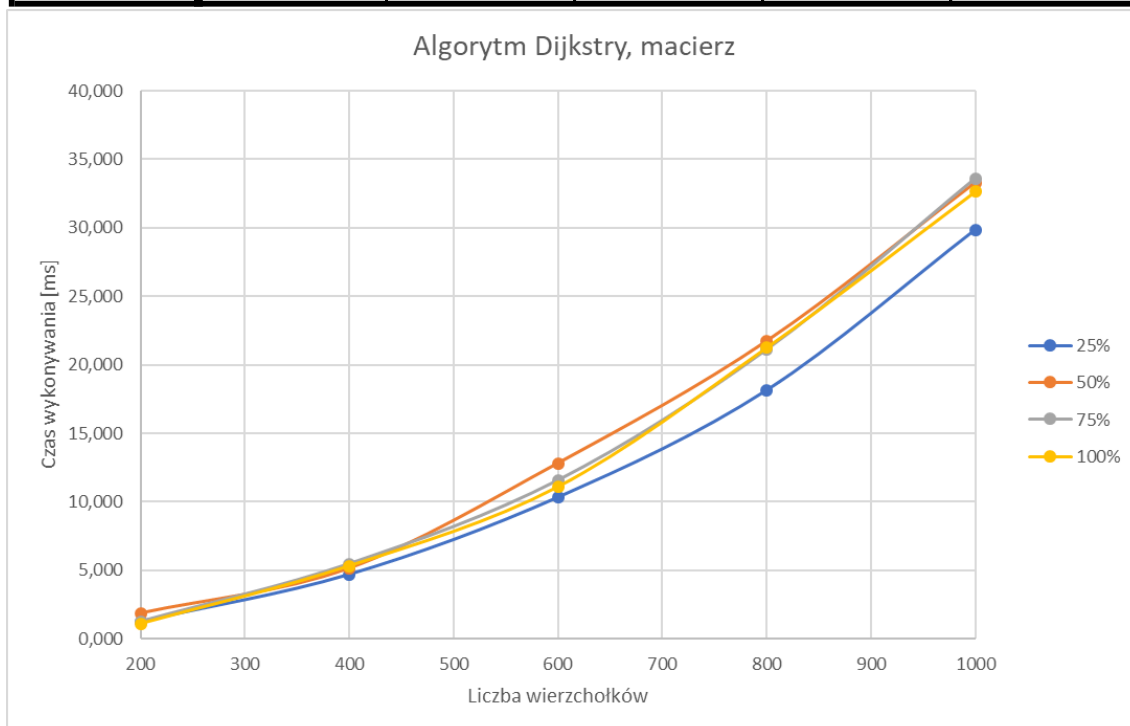
### 2. Wyniki: algorytm Dijkstry, reprezentacja listą sąsiedztwa

Algorytm Dijkstry, lista [ms]					
Gęstość	Liczba wierzchołków grafu				
	200	400	600	800	1000
25%	0,509	2,588	6,068	11,872	15,934
50%	0,781	3,905	6,405	11,560	17,496
75%	0,781	3,593	7,030	14,216	22,495
100%	0,781	4,374	9,529	16,403	26,869



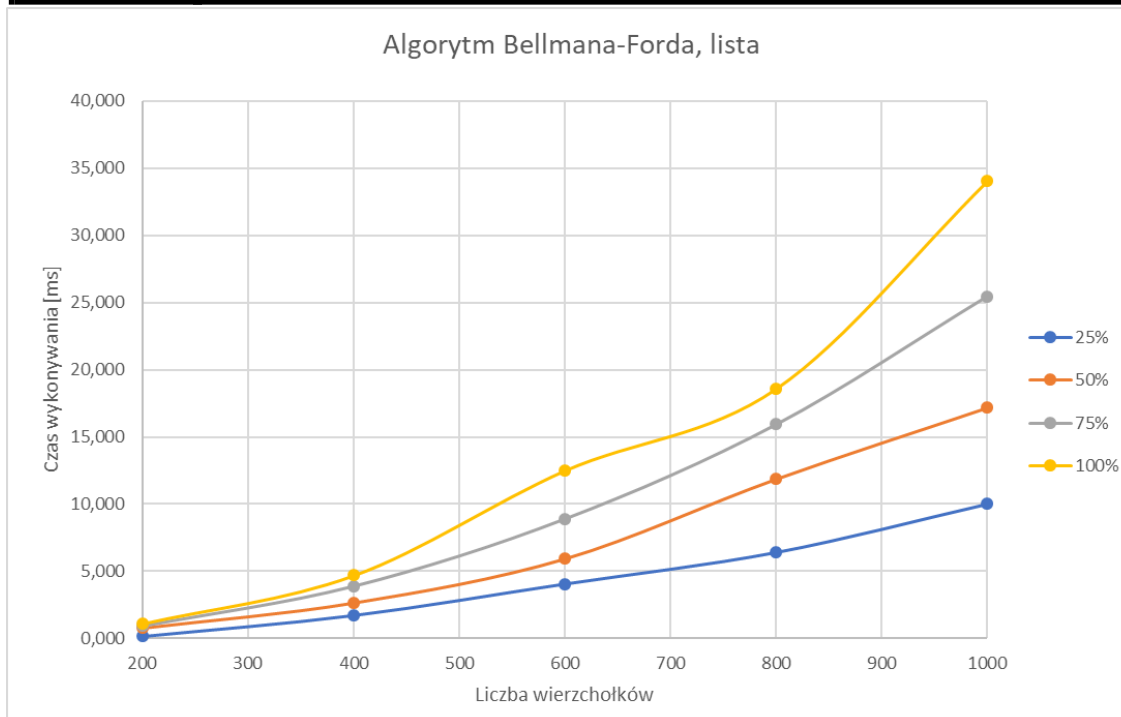
### 3. Wyniki: algorytm Dijkstry, reprezentacja macierzą sąsiedztwa

Algorytm Dijkstry, macierz [ms]					
Gęstość	Liczba wierzchołków grafu				
	200	400	600	800	1000
25%	1,250	4,686	10,310	18,121	29,837
50%	1,875	5,155	12,810	21,714	33,273
75%	1,250	5,467	11,560	21,089	33,586
100%	1,093	5,311	11,091	21,245	32,648



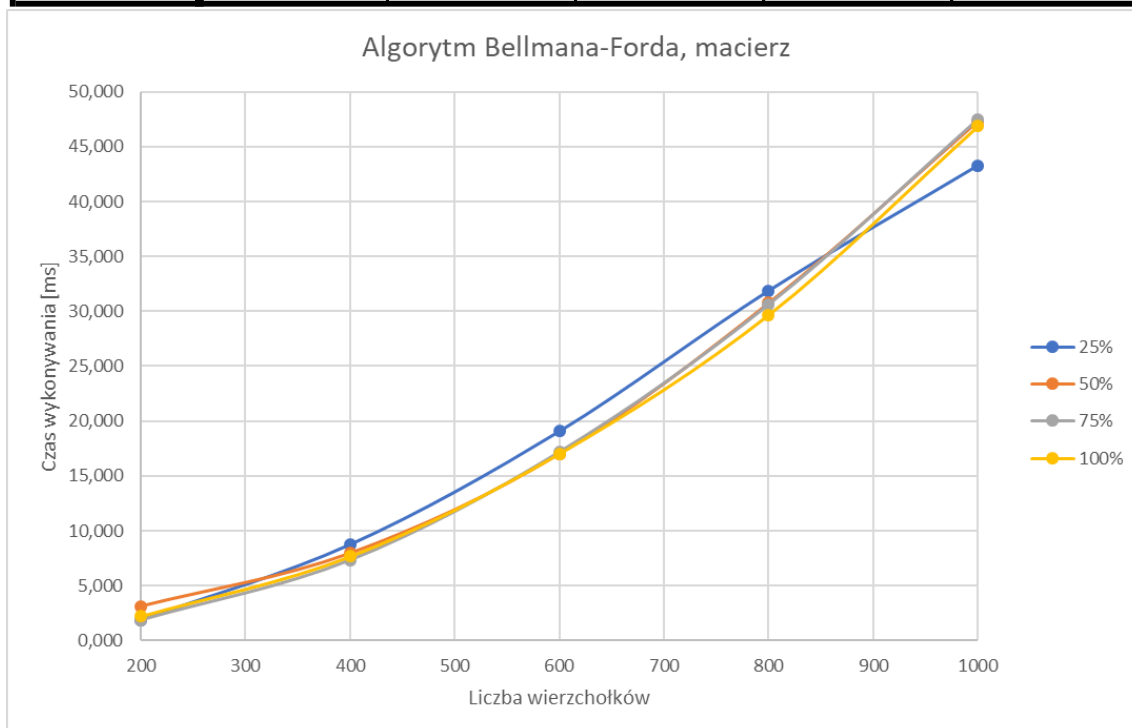
#### 4. Wyniki: algorytm Bellmana-Forda, reprezentacja listą sąsiedztwa

Algorytm Bellmana-Forda, lista [ms]					
Gęstość	Liczba wierzchołków grafu				
	200	400	600	800	1000
25%	0,156	1,718	4,062	6,405	9,998
50%	0,781	2,656	5,936	11,873	17,184
75%	0,937	3,905	8,904	15,934	25,463
100%	1,093	4,686	12,497	18,590	34,055



## 5. Wyniki: algorytm Bellmana-Forda, reprezentacja macierzą sąsiedztwa

Algorytm Bellmana-Forda, macierz [ms]					
Gęstość	Liczba wierzchołków grafu				
	200	400	600	800	1000
25%	1,875	8,748	19,058	31,868	43,271
50%	3,124	7,967	17,027	30,774	47,333
75%	1,875	7,342	17,183	30,618	47,489
100%	2,187	7,655	17,027	29,681	46,864



## 6. Wnioski

Algorytm Dijkstry w testach był szybszy niż algorytm Bellmana-Forda dla tego samego formatu reprezentacji grafu, zaś lista sąsiedztwa okazała się być wydajniejszą reprezentacją niż macierz sąsiedztwa.

Gęstość grafu wpływa na czas wykonywania algorytmów. Dla reprezentacji w postaci macierzy wpływ ten jest pomijalnie mały, w przeciwieństwie do listy dla której gęstość jest jednym z głównych parametrów decydujących o czasie wykonywania algorytmu.