

Politechnika Wrocławska
Wydział Elektroniki
Sprawozdanie z projektu

Projektowanie Algorytmów i Metody Sztucznej Inteligencji

Grafy



Politechnika
Wrocławska

1. Wprowadzenie

Celem projektu jest zbadanie wydajności algorytmu Bellmana – Forda lub Dijkstry w znajdowaniu najkrótszej drogi od wybranego wierzchołka do wszystkich pozostałych w warunkach różnej ilości wierzchołków oraz gęstości grafu. W projekcie testowany był algorytm Bellmana – Forda w zależności od sposobu reprezentacji grafu (w postaci macierzy sąsiedztwa i listy sąsiedztwa) oraz gęstości grafu.

Grafy, na których został testowany algorytm różniły się gęstością (ilością krawędzi), ilością węzłów oraz samą reprezentacją grafu.

Testowane reprezentacje grafu: Macierz sąsiedztwa, Lista sąsiedztwa

Testowane gęstości: 25%, 50%, 75%, 100% (graf pełny)

Testowane ilości węzłów: 5, 10, 30, 50, 100

2. Opis algorytmu Bellmana-Forda

Algorytm Bellmana-Forda w porównaniu z algorytmem Dijkstry jest wolniejszy, jednakże bardziej uniwersalny. Algorytm ten jest bowiem w stanie obsługiwać grafy z ujemnymi wartościami krawędzi oraz wykrywać ujemne cykle, które być może zostały stworzone przez te wartości.

Złożoność obliczeniowa algorytmu:

- Dla reprezentacji w postaci listy sąsiedztwa wynosi $O(VE)$
- Dla reprezentacji w postaci macierzy sąsiedztwa wynosi $O(V^3)$

Złożoność pamięciowa algorytmu (przy założeniu, że bierzemy pod uwagę również sam graf):

- Dla reprezentacji w postaci listy sąsiedztwa wynosi $O(V+E)$
- Dla reprezentacji w postaci macierzy sąsiedztwa wynosi $O(V+V^2) = O(V^2)$

3. Zasada działania programu

Dzięki wykorzystaniu klas (Graph, List i Matrix) można zauważyć, iż program został napisany w sposób obiektowy. Wyżej wymienione klasy są powiązane ze sobą za pomocą dziedziczenia, które graf opisuje ogólnie jako „ojciec” i szczegółowo jako „dziecko”. Powiązania w grafie są budowane na podstawie krawędzi i listy krawędzi.

Wykorzystując możliwości algorytmu jesteśmy w stanie wyznaczyć najmniejszą odległość od ustalonego wierzchołka S do wszystkich pozostałych w skierowanym grafie bez cykli o ujemnej długości. Warunek nieujemności cyklu jest spowodowany faktem, że w grafie o ujemnych cyklach najmniejsza odległość między niektórymi wierzchołkami jest nieokreślona, ponieważ zależy od liczby przejść w cyklu.

Dla podanego grafu macierz A dla każdej pary wierzchołków u i v zawiera wagę krawędzi (u,v), przy czym jeśli krawędź (u,v) nie istnieje, to przyjmujemy, że jej waga wynosi nieskończoność.

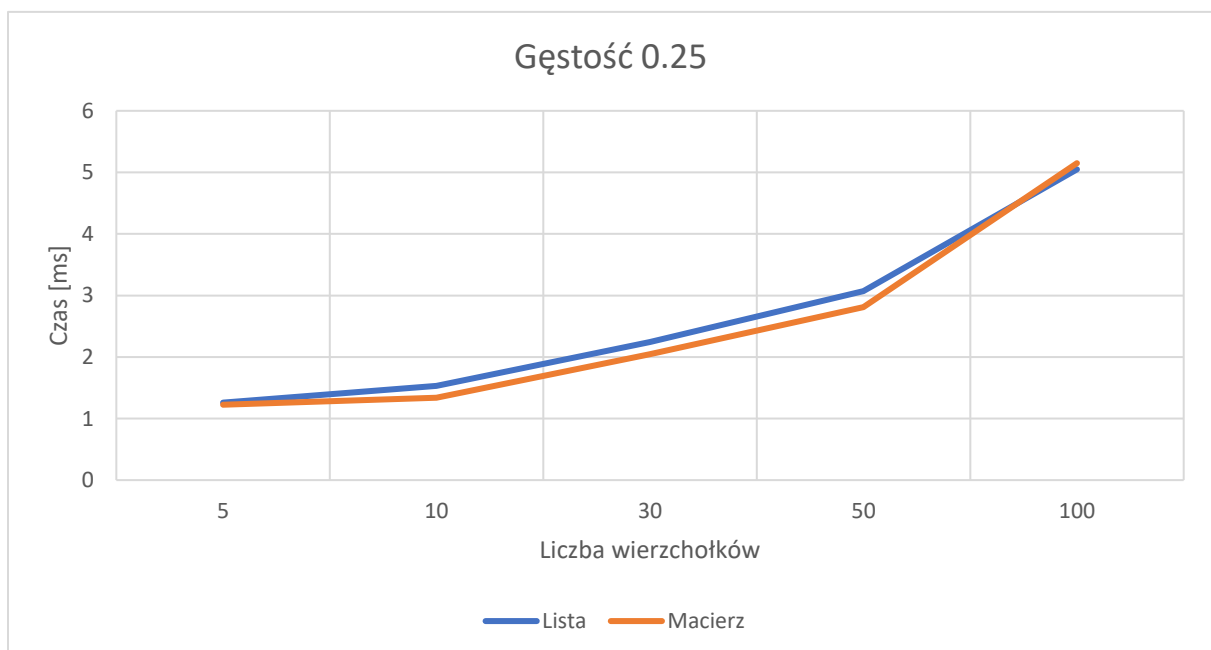
Algorytm Forda-Bellmana w każdym kroku oblicza górne oszacowanie $D(v)$ odległości od wierzchołka S do wszystkich pozostałych wierzchołków v_i . W pierwszym kroku przyjmujemy $D(v)=A(s,v)$. Gdy stwierdzimy, że $D(v)>D(u)+A(u,v)$, to każdorazowo polepszamy aktualne oszacowanie i podstawiamy $D(v):=D(u)+A(u,v)$. Algorytm kończy się, gdy żadnego oszacowania nie można już poprawić, macierz $D(v_i)$ zawiera najkrótsze odległości od wierzchołka S do wszystkich pozostałych. Algorytmy jako funkcje badają dane grafy i zwracają czas potrzebny na wykonanie działania algorytmu.

4. Wyniki

4.1 Gęstość 0.25

	Lista	Macierz
5	1,259	1,2248
10	1,5287	1,3402
30	2,2429	2,0474
50	3,0728	2,8115
100	5,0492	5,149

Tab. 1

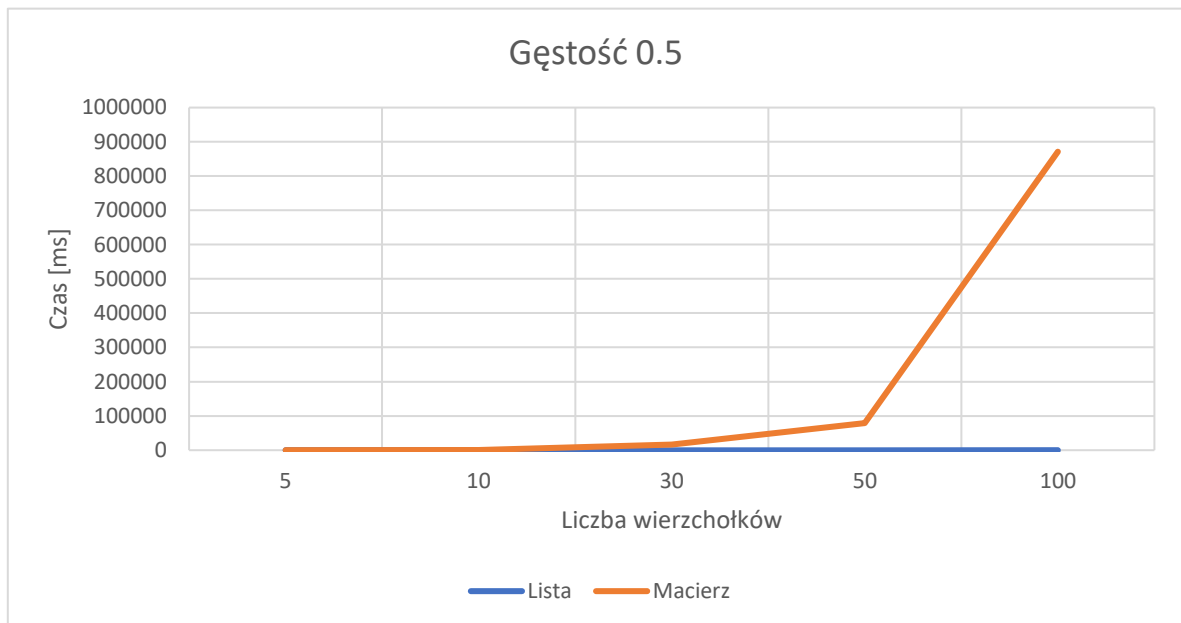


Wyk. 1

4.2 Gęstość 0.5

	Lista	Macierz
5	1,1972	62,1924
10	1,3561	550,5274
30	2,2454	16682,3893
50	2,8909	79227,2426
100	4,8988	871150,9267

Tab. 2

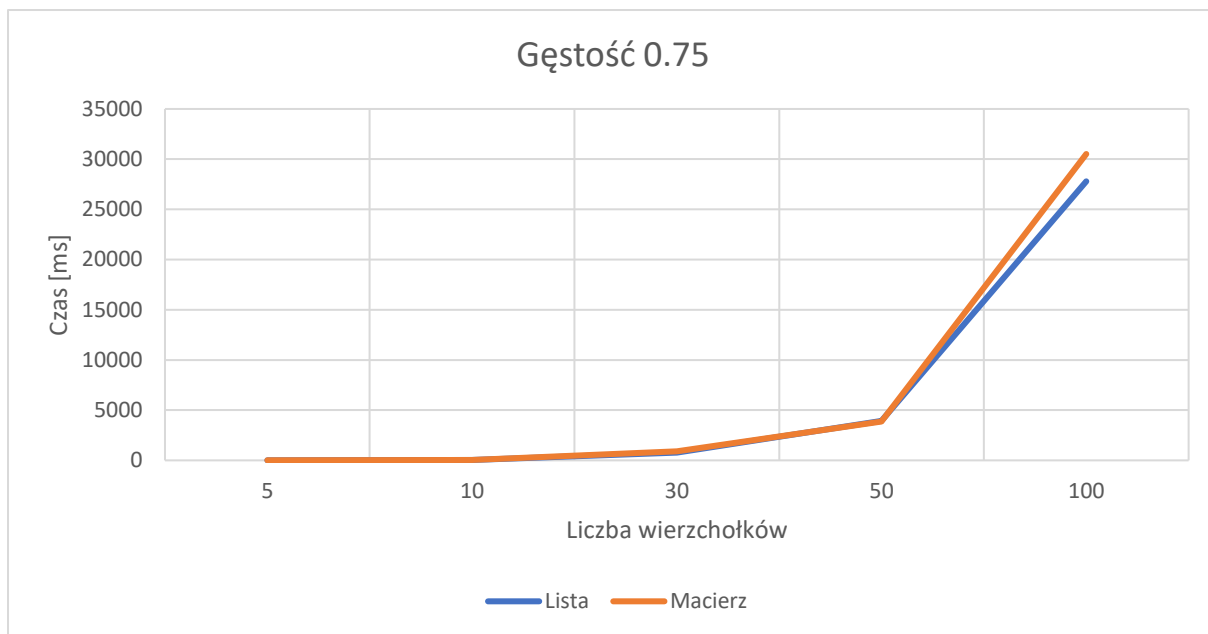


Wyk. 2

4.3 Gęstość 0.75

	Lista	Macierz
5	6,3878	7,6843
10	38,7873	43,1632
30	770,3111	888,9572
50	3916,1611	3881,2352
100	27789,7761	30512,7881

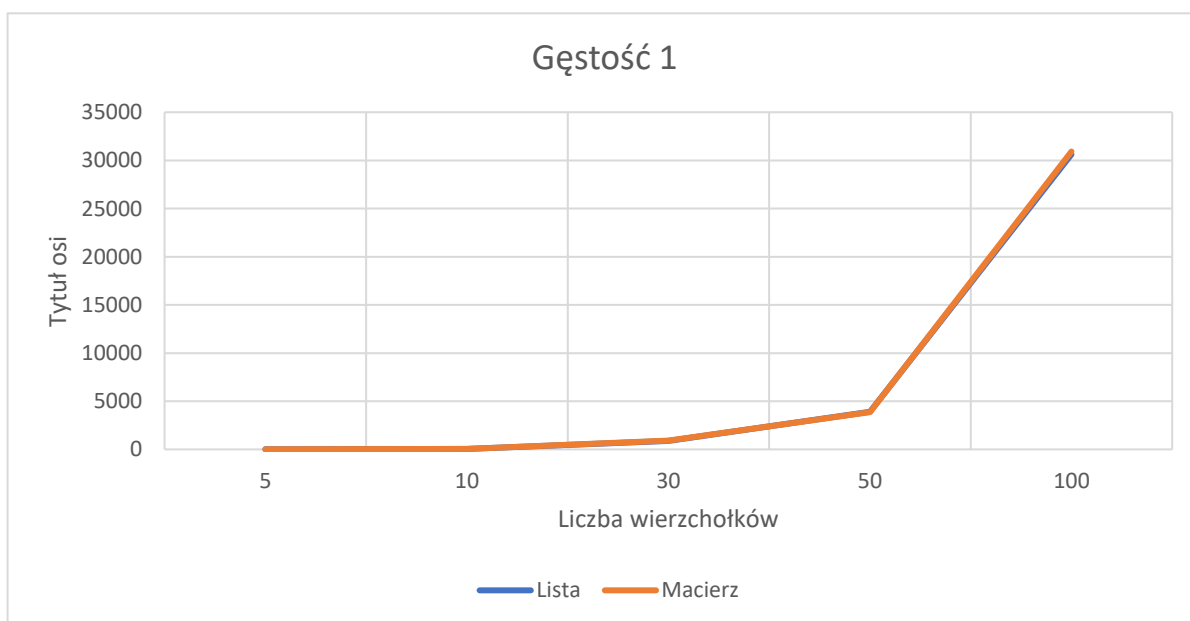
Tab.3



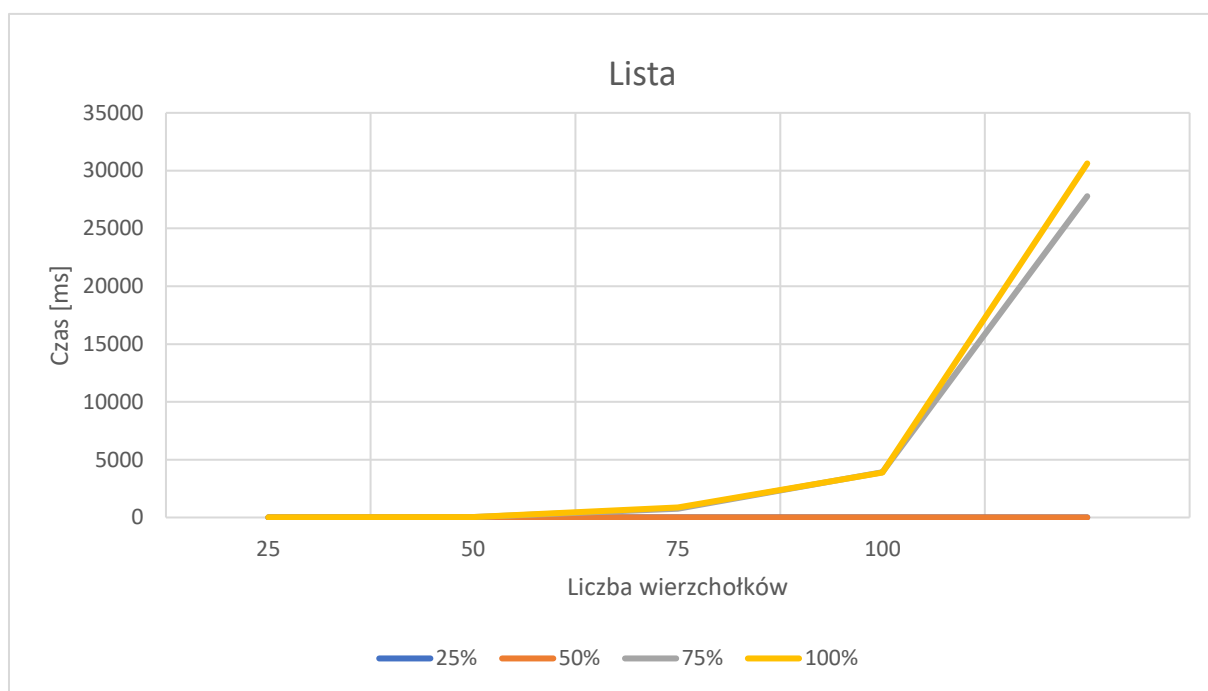
4.4 Gęstość 1

	Lista	Macierz
5	10,8793	10,8176
10	46,4754	49,4765
30	871,9475	904,3662
50	3889,3163	3872,4544
100	30624,4683	30916,6733

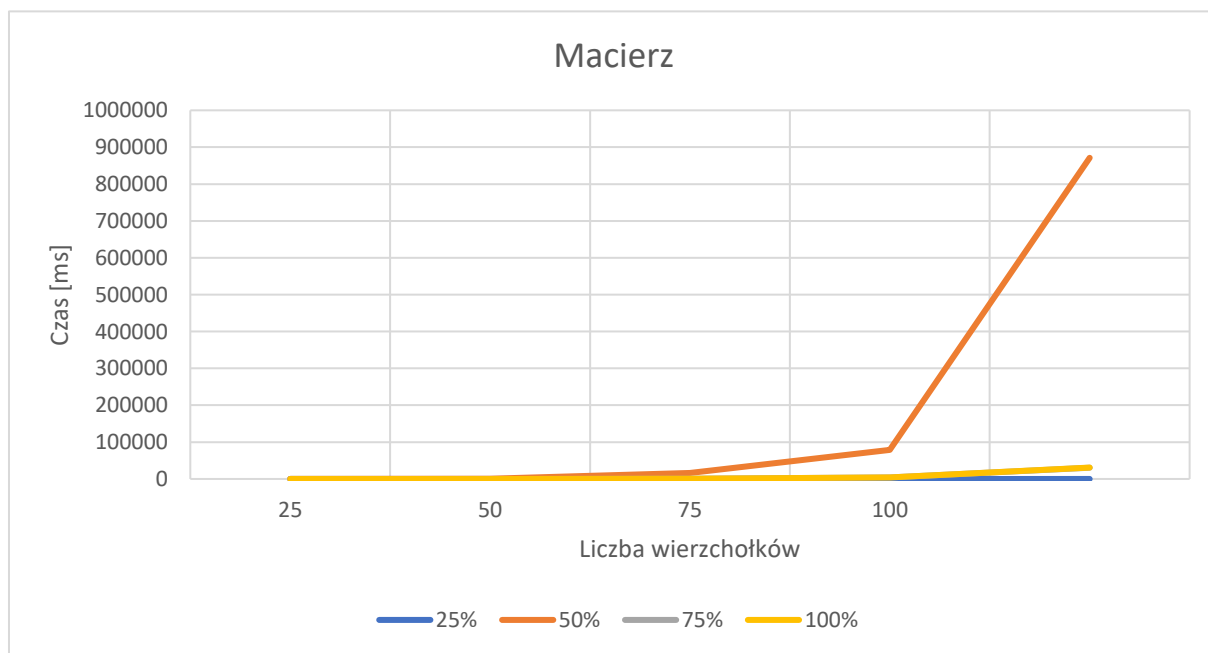
Tab. 4



4.5 Wykres działania listy w zależności od gęstości



4.6 Wykres działania macierzy w zależności od gęstości



5. Wnioski

Czas działania dla większej ilości wierzchołków oraz większej gęstości grafu w przypadku listy rośnie. Czas działania macierzy sąsiedztwa prawie nie zależy od gęstości, może to być spowodowane tym, że niezależnie czy w danym polu jest określona waga czy nieskończoność, porównanie jest wykonywane. Na czas może wpływać również budowa listy i macierzy. Przeszukiwanie całej tablicy jest wolniejsze niż dostęp do danych w zaimplementowanej liście. Dla listy sąsiedztwa, im więcej sąsiadów ma dany wierzchołek, tym dłużej trwa przeszukiwanie, więc dla małej gęstości grafu jest to metoda wydajniejsza.