Sprawozdanie

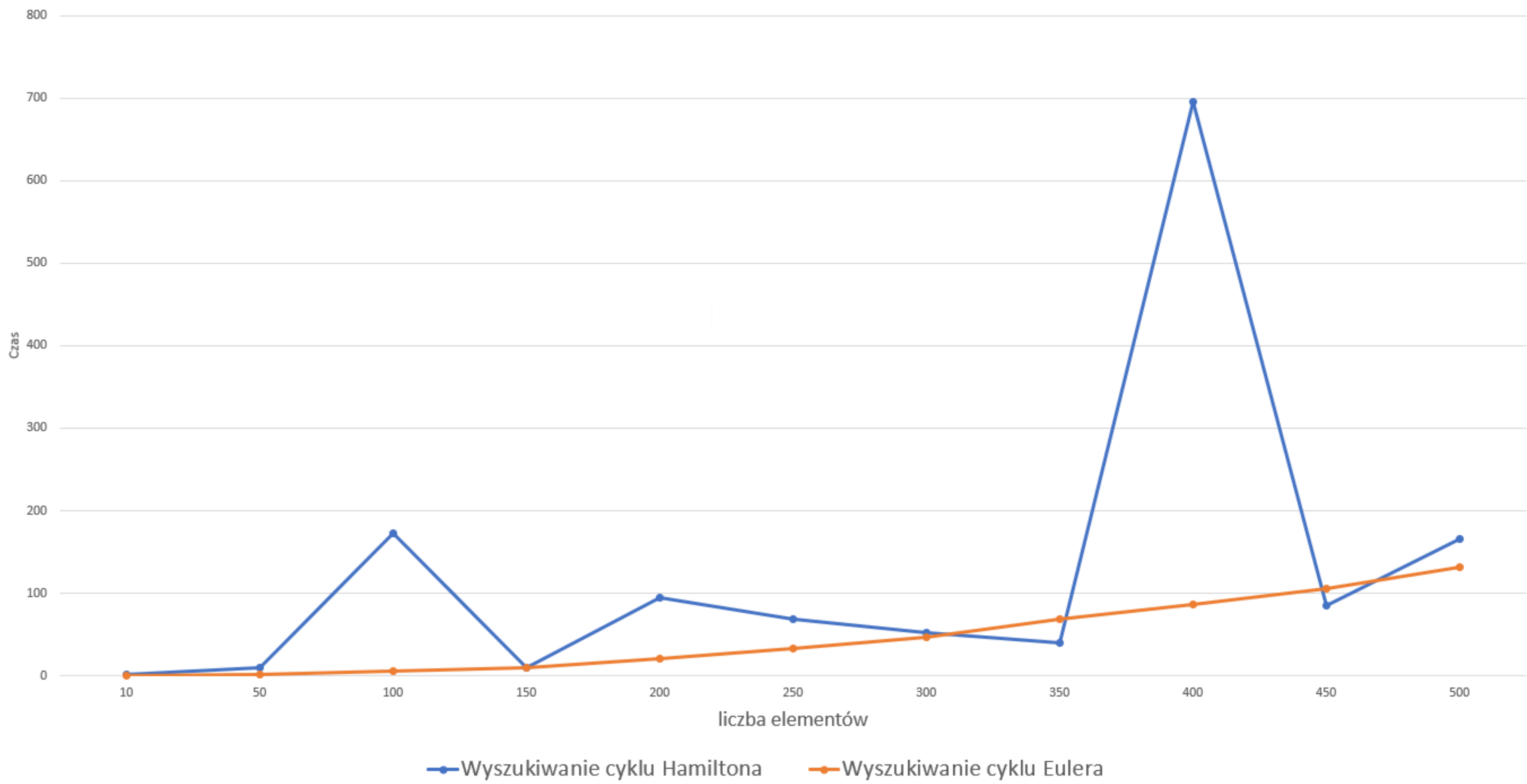
Algorytmy z powracaniem

Mikołaj Pluta

1) Celem ćwiczenia jest porównanie działania algorytmów z powracaniem – w szczególności algorytmów znajdujących cykle Eulera i Hamiltona w grafach. Powyższe algorytmy zaimplementowane zostały w języku Python.

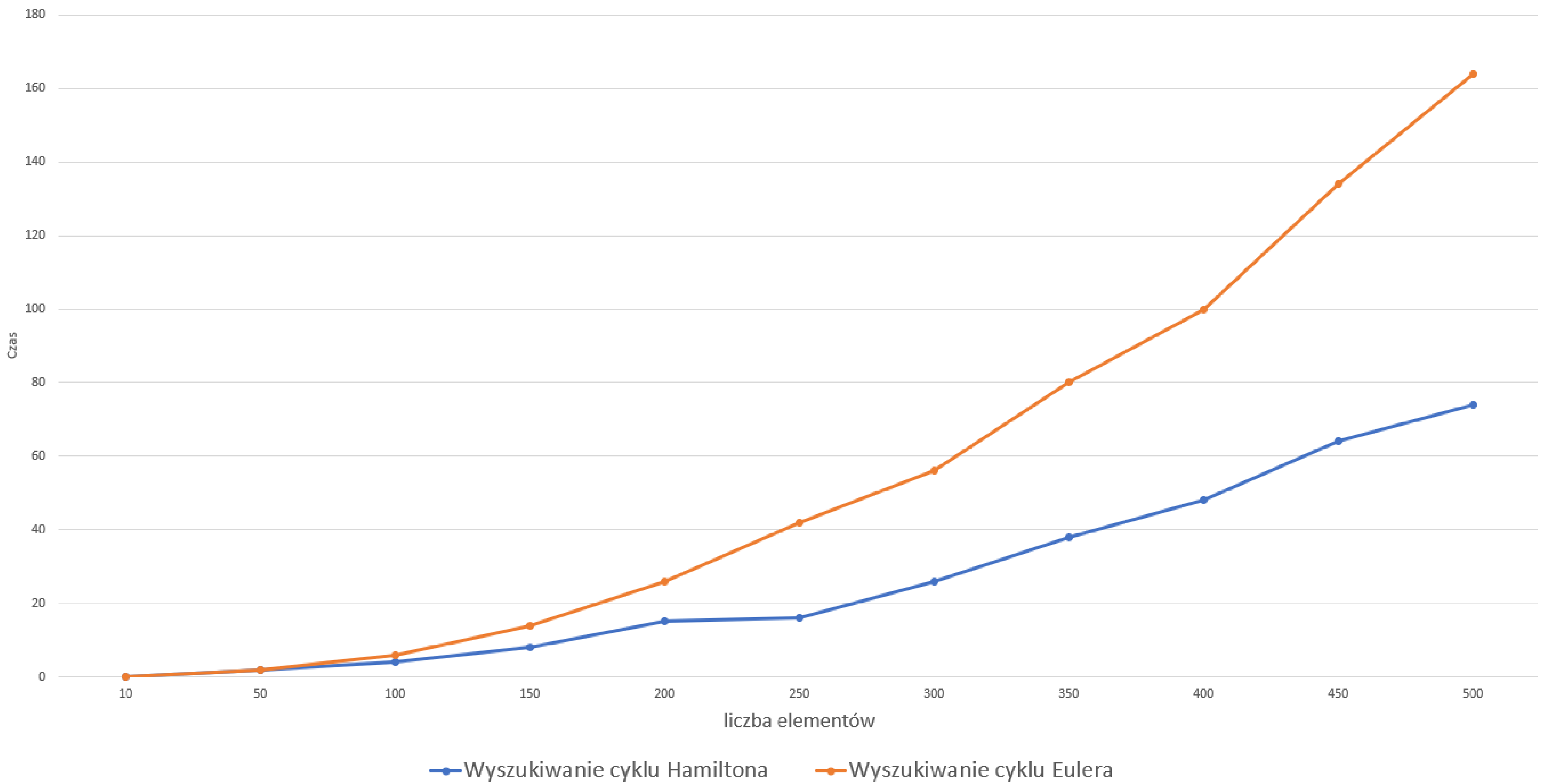
2) Zależność czasu obliczeń od ilości wierzchołków w grafie przy stałym nasyceniu. Wszystkie czasy wykonywania przedstawione są w 1/10000s ze względu na ograniczenia sprzętowe(niektóre algorytmy dla dużej liczby wierzchołków wymagają zbyt wiele miejsca w pamięci, nawet po zwiększeniu rozmiaru stosu do maksimum). Z powodu małego prawdopodobieństwa wygenerowania grafu Eulerowskiego przy dużej liczbie wierzchołków zaimplementowany algorytm sprowadza się w praktyce do problemu decyzyjnego i sprawdza tylko czy taki cykl występuje. Każdy punkt na wykresie reprezentuje średnią z 3 operacji.

2.1) Czas wykonywania algorytmów w grafie nieskierowanym w zależności od liczby wierzchołków przy stałym nasyceniu s = 50%.



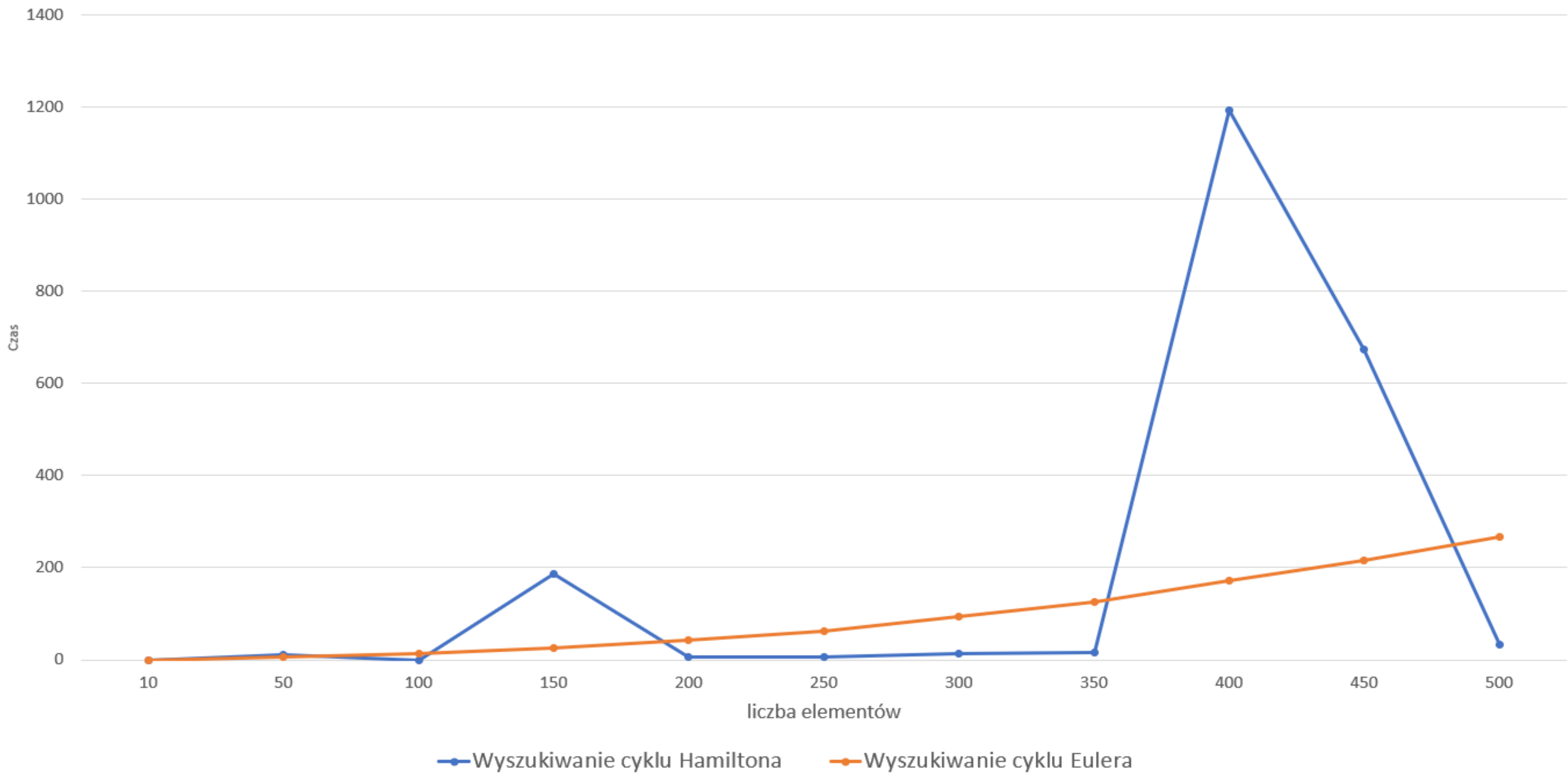
Niestety przy nasyceniu s = 50% czas wykonywania zależy od struktury wygenerowanego grafu i w niekorzystnym przypadku czas ten może być bardzo długi w stosunku do innych – nawet większych wartości n. Sytuacja poprawia się kiedy zwiększymy s.

2.2) Czas wykonywania algorytmów w grafie nieskierowanym w zależności od liczby wierzchołków przy stałym nasyceniu s = 90%.



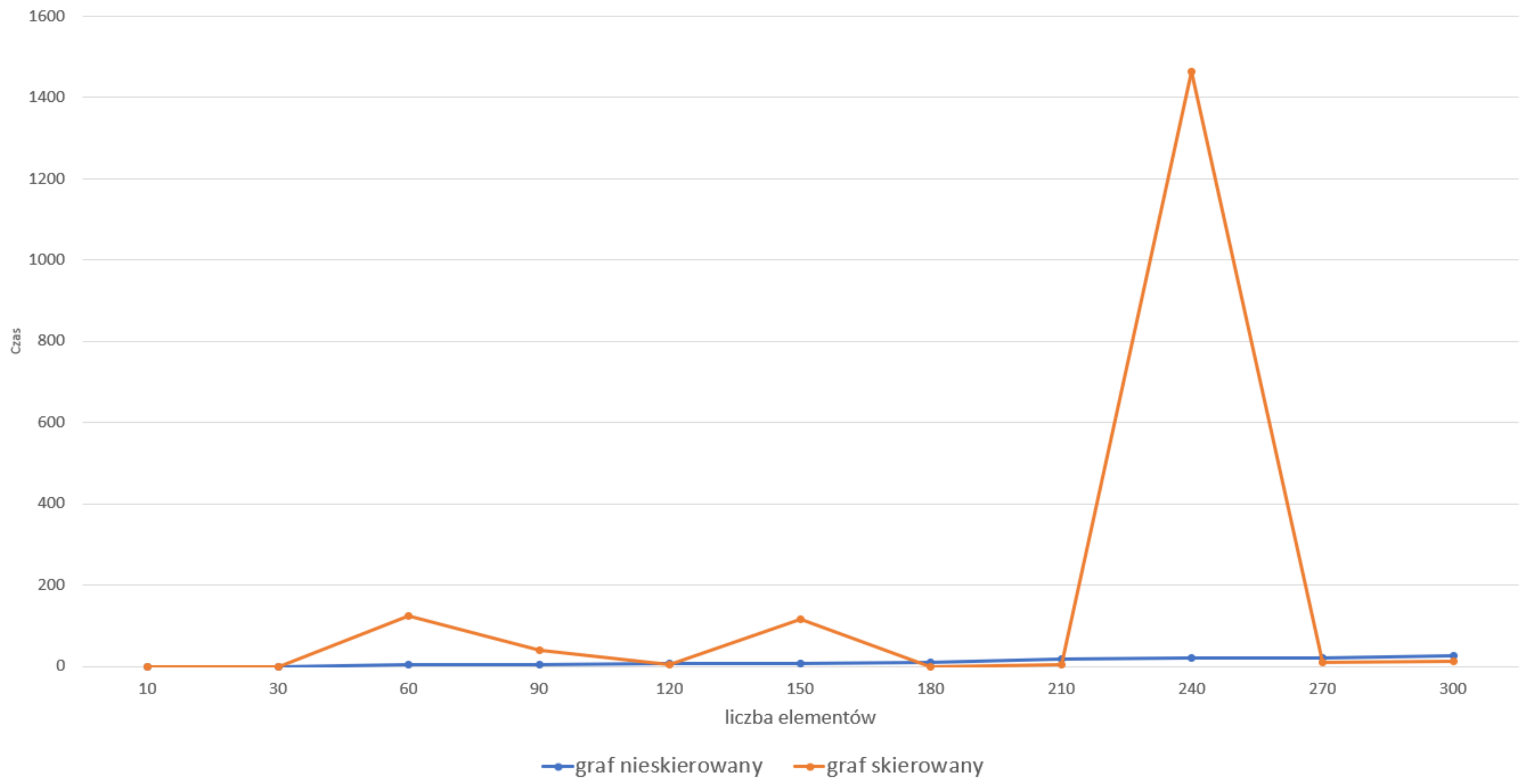
Przy większej ilości krawędzi łatwiej znaleźć cykle Hamiltona, dlatego powyższy wykres lepiej oddaje jego czas wykonywania, niż graf w punkcie 2.1.

2.3) Czas wykonywania algorytmów w grafie skierowanym w zależności od liczby wierzchołków przy stałym nasyceniu s = 80%.



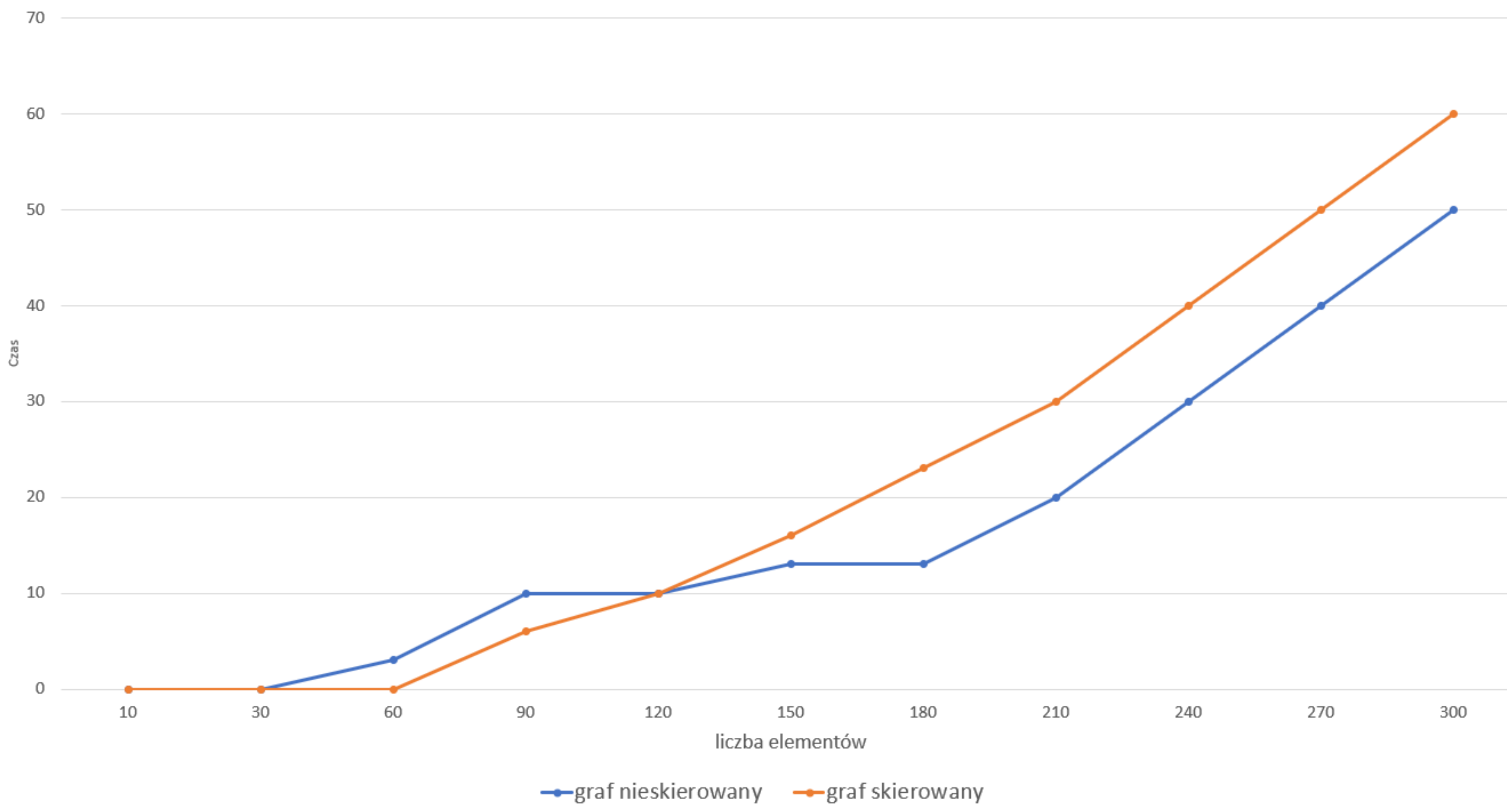
W grafie skierowanym poruszanie się po grafie zostaje ograniczone(ze względu na to, że można poruszać się tylko w jednym kierunku), dlatego nawet dla dużego zagęszczenia czas wykonywania zależy od struktury wygenerowanego losowo grafu, przez co jest nieregularny i ciężki do przewidzenia. Przeszukiwanie cyklu Eulera ponownie sprowadza się do rozwiązania problemu decyzyjnego.

2.4) Czas wykonywania algorytmu przeszukiwania cyklu Hamiltona w zależności od ilości n wierzchołków przy stałym nasyceniu s = 80%



Podobnie jak w poprzednich punktach zauważyć można, że dla grafu skierowanego nawet przy dużym zagęszczeniu krawędzi czas wykonywania mocno zleży od struktury grafu, np. niekorzystny przypadek przy 240 elementach.

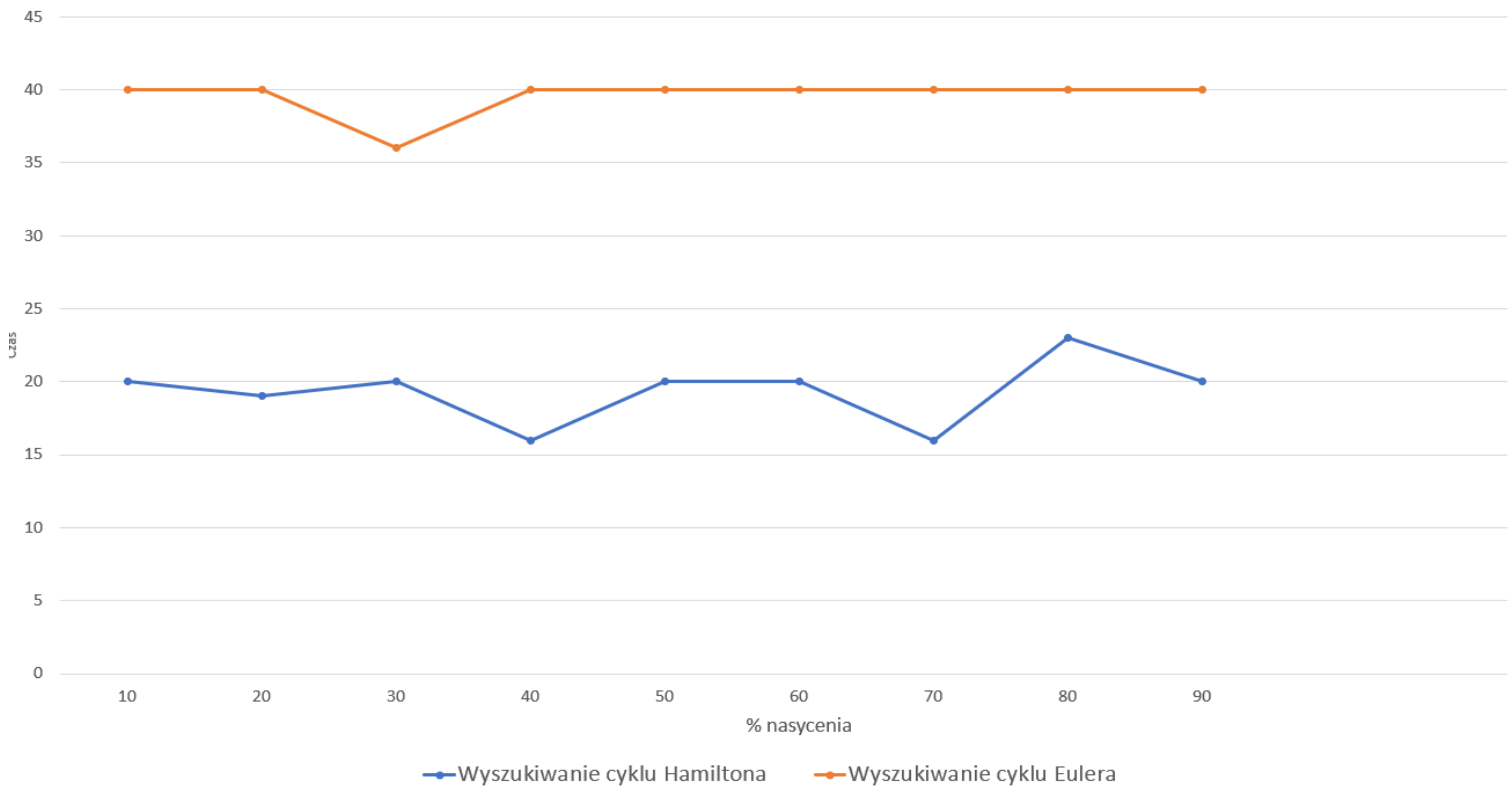
2.5) Czas wykonywania algorytmu przeszukiwania cyklu Eulera w zależności od ilości n wierzchołków przy stałym nasyceniu s = 50%



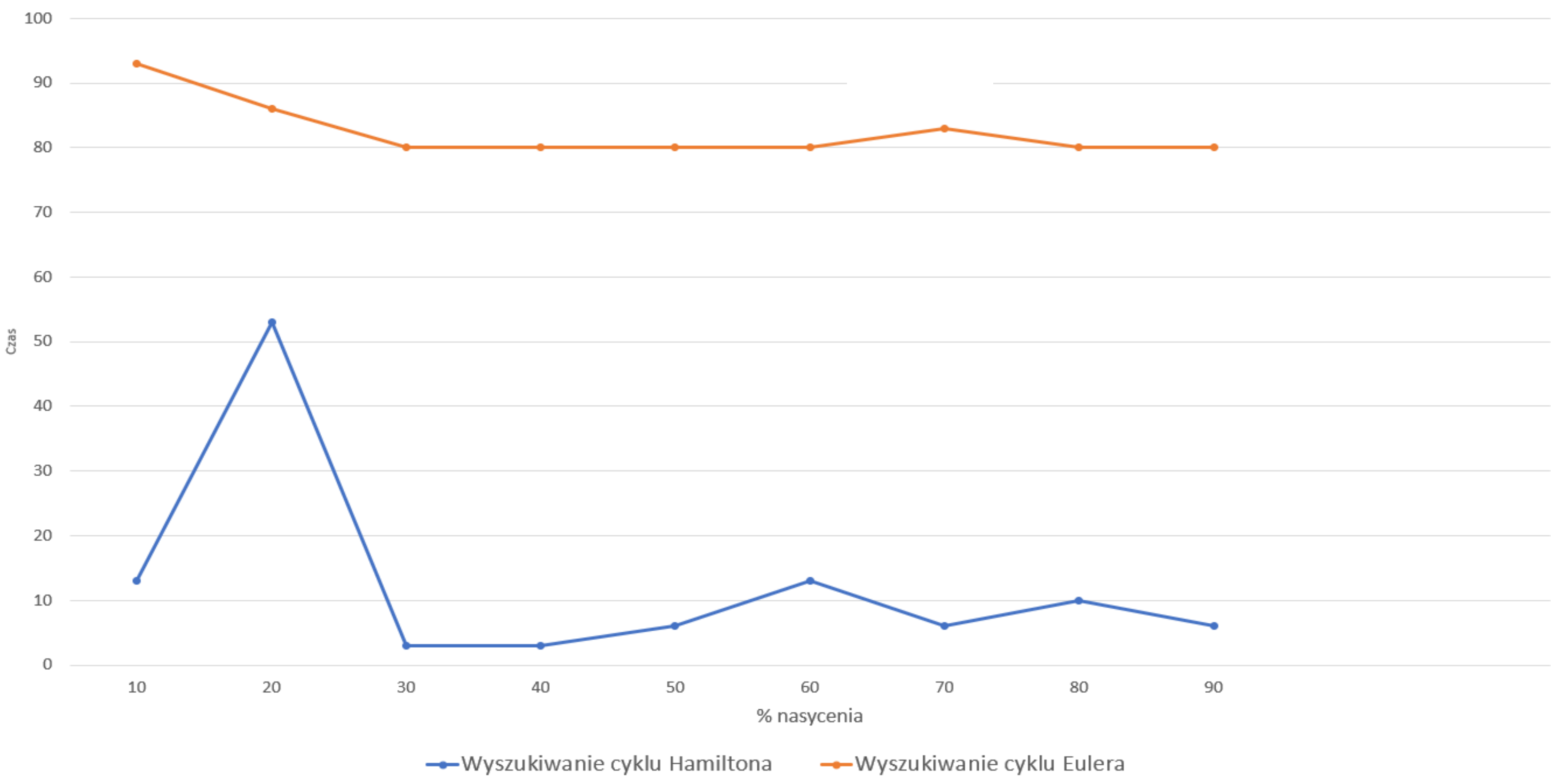
Mimo, że działanie algorytmu sprowadza się do odpowiedzi na pytanie czy cykl Eulera istnieje, zauważyć można, że lista następników, na której oparte są grafy skierowane radzi sobie lepiej z przeszukiwaniem krawędzi, niż macierz grafu, która jest bazą grafów nieskierowanych.

3) Zależność czasu wykonywania algorytmów od ilości krawędzi przy stałej liczbie wierzchołków.

3.1) Czas wykonywania algorytmów w grafie nieskierowanym w zależności od nasycenia krawędzi przy stałej liczbie wierzchołków n = 250.

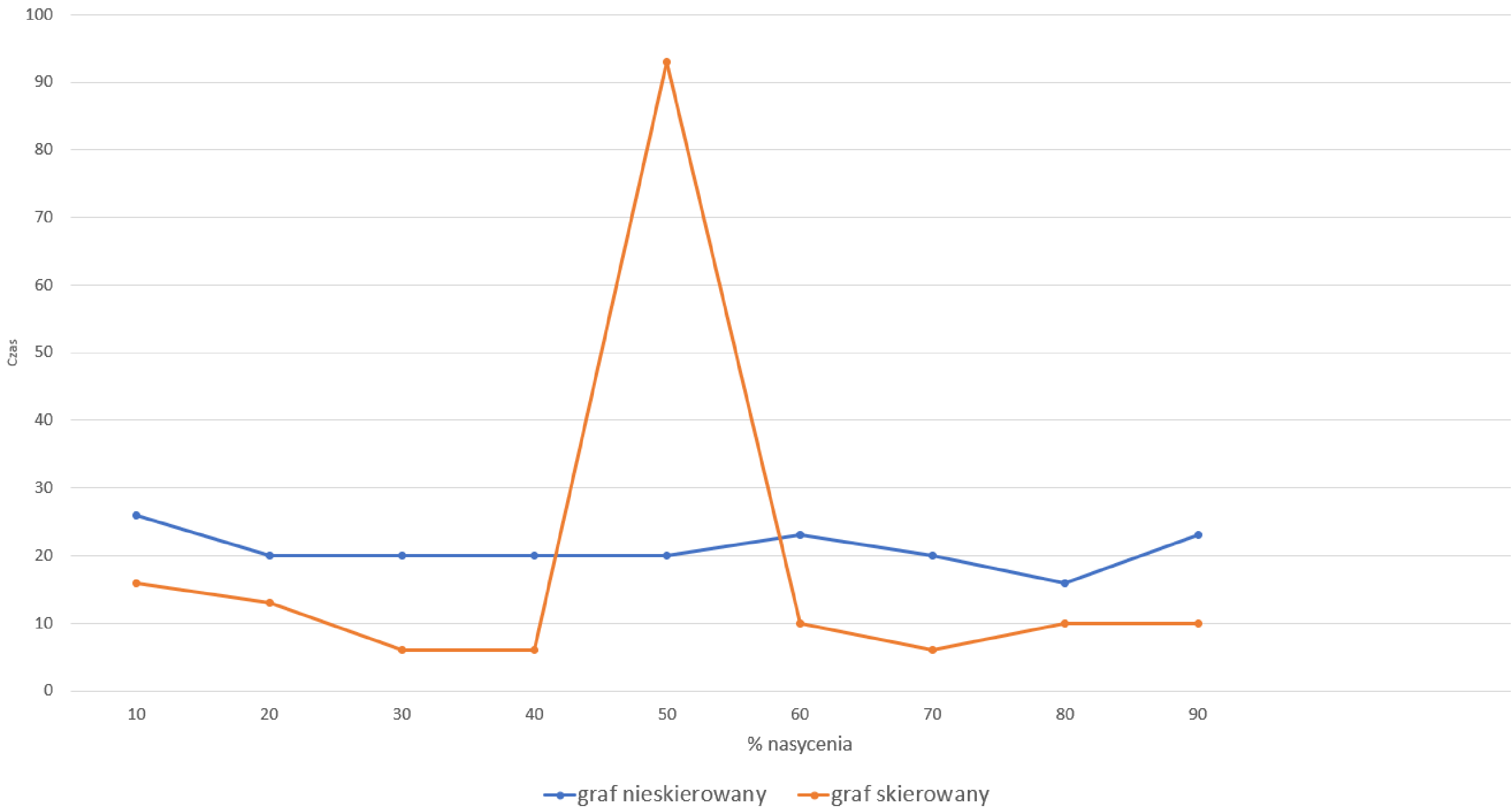


3.2) Czas wykonywania algorytmów w grafie skierowanym w zależności od nasycenia krawędzi przy stałej liczbie wierzchołków n=250.

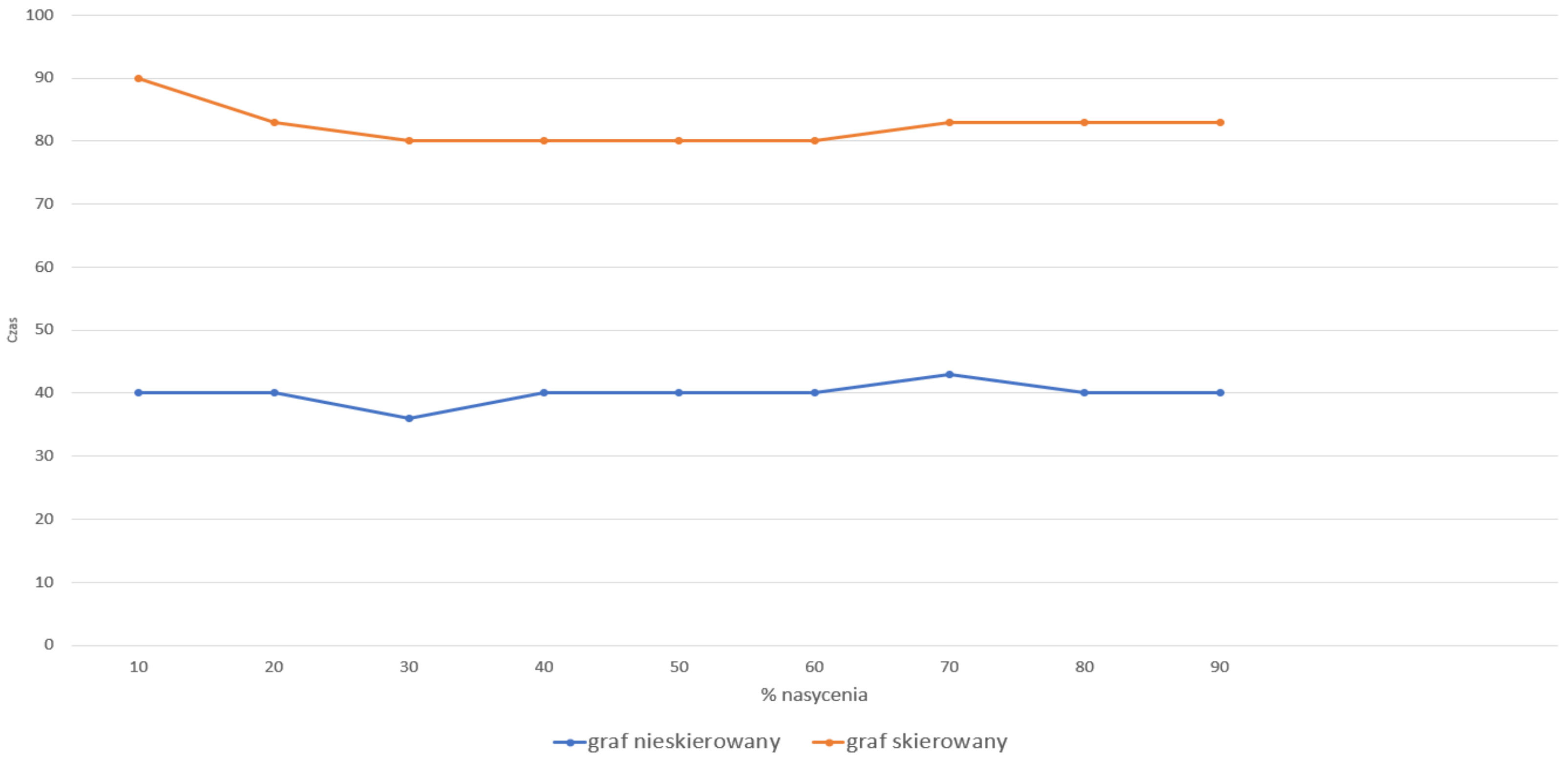


Czas wykonywania algorytmów przy stałej liczbie wierzchołków zdaje się być podobny dla każdej wartości nasycenia krawędziami. Wraz ze wzrostem nasycenia s w przypadku poszukiwania cyklu Hamiltona maleje prawdopodobieństwo tego, że pojawi się niekorzystny przypadek(np. dla nasycenia 20 w punkcie 3.2) i czas wykonywania będzie znacznie większy. Sytuacja jest odwrotna w przypadku cyklu Eulera – czym więcej krawędzi, tym więcej może być opcji, które algorytm musiał będzie przetworzyć.

3.3) Czas wyszukiwania cyklu Hamiltona w grafie skierowanym i nieskierowanym w zależności od nasycenia, przy stałej liczbie wierzchołków n = 250.



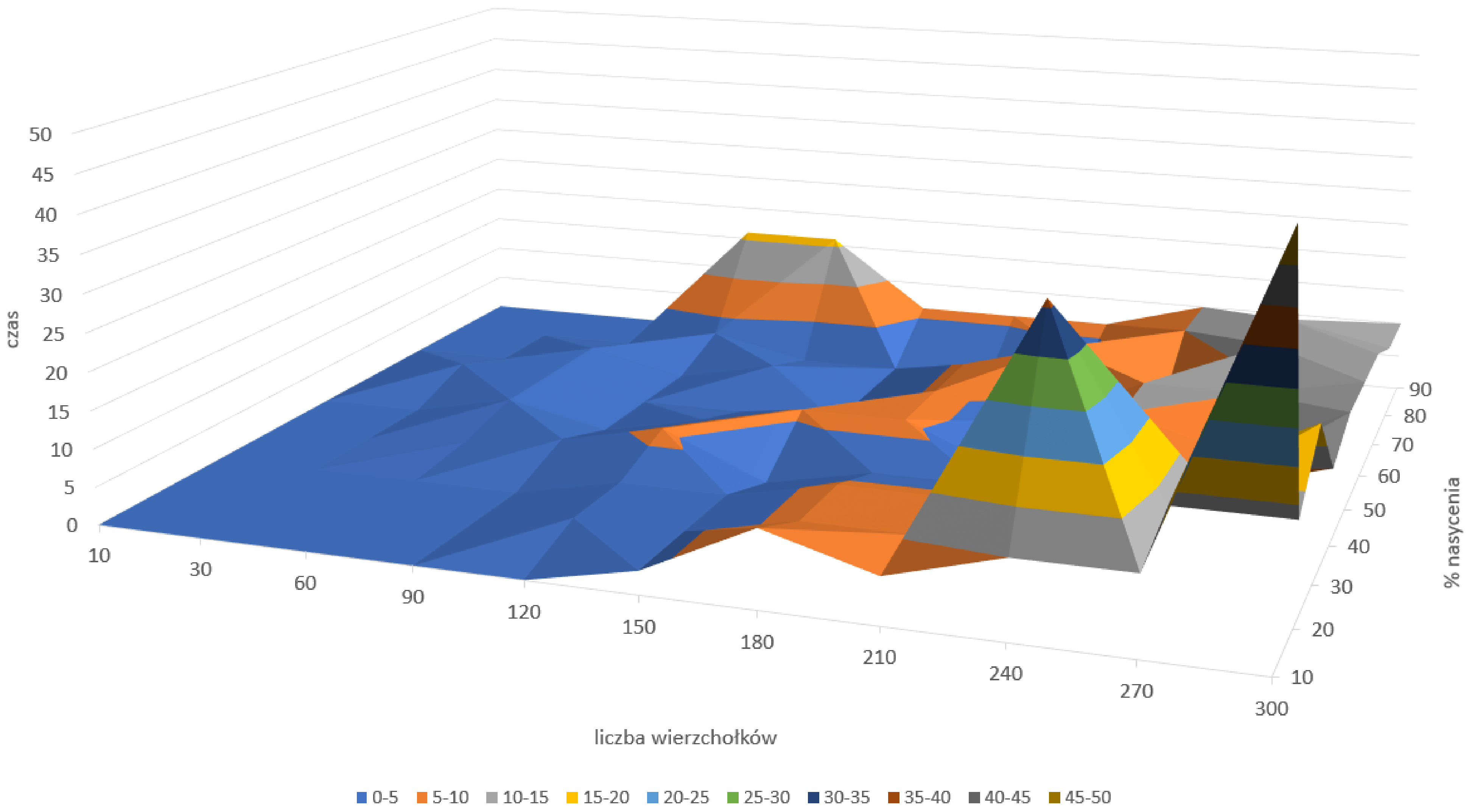
3.4) Czas wyszukiwania cyklu Eulera w grafie skierowanym i nieskierowanym w zależności od nasycenia, przy stałej liczbie wierzchołków n = 250.



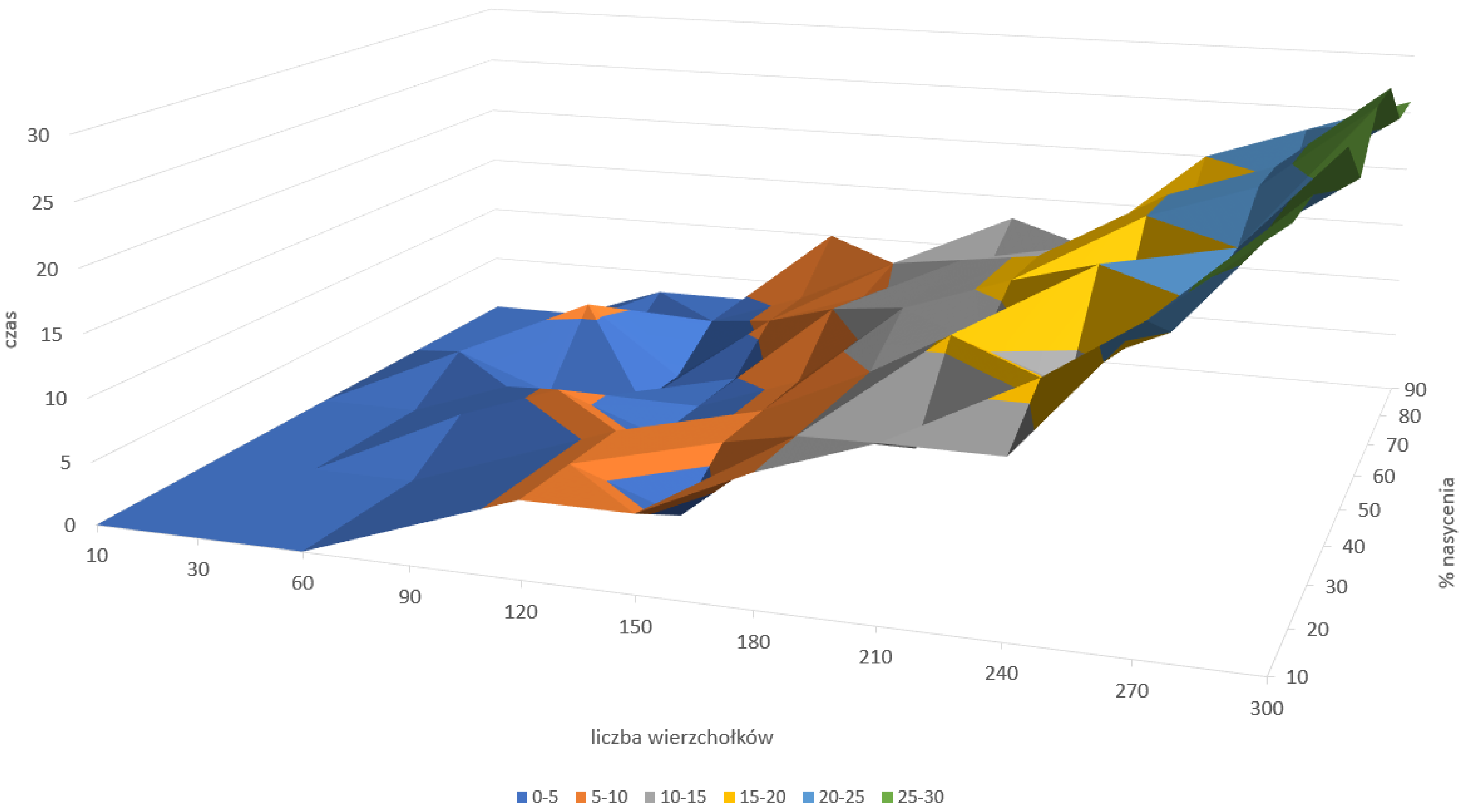
Na powyższych dwóch wykresach zauważyć można, że algorytm wyszukiwania Cyklu Hamiltona lepiej radzi sobie kiedy graf jest skierowany, a algorytm wyszukiwania cyklu Eulera – w przeciwnym przypadku, czyli kiedy graf jest nieskierowany. Wynika to bezpośrednio ze sposobu reprezentacji maszynowej grafów. (Anomalia w punkcie 3.3 wynika z wyjątkowo niekorzystnej dla algorytmu struktury jednego z wygenerowanych grafów.)

4) Zależność czasu obliczeń od liczby wierzchołków n w grafie i nasycenia s.

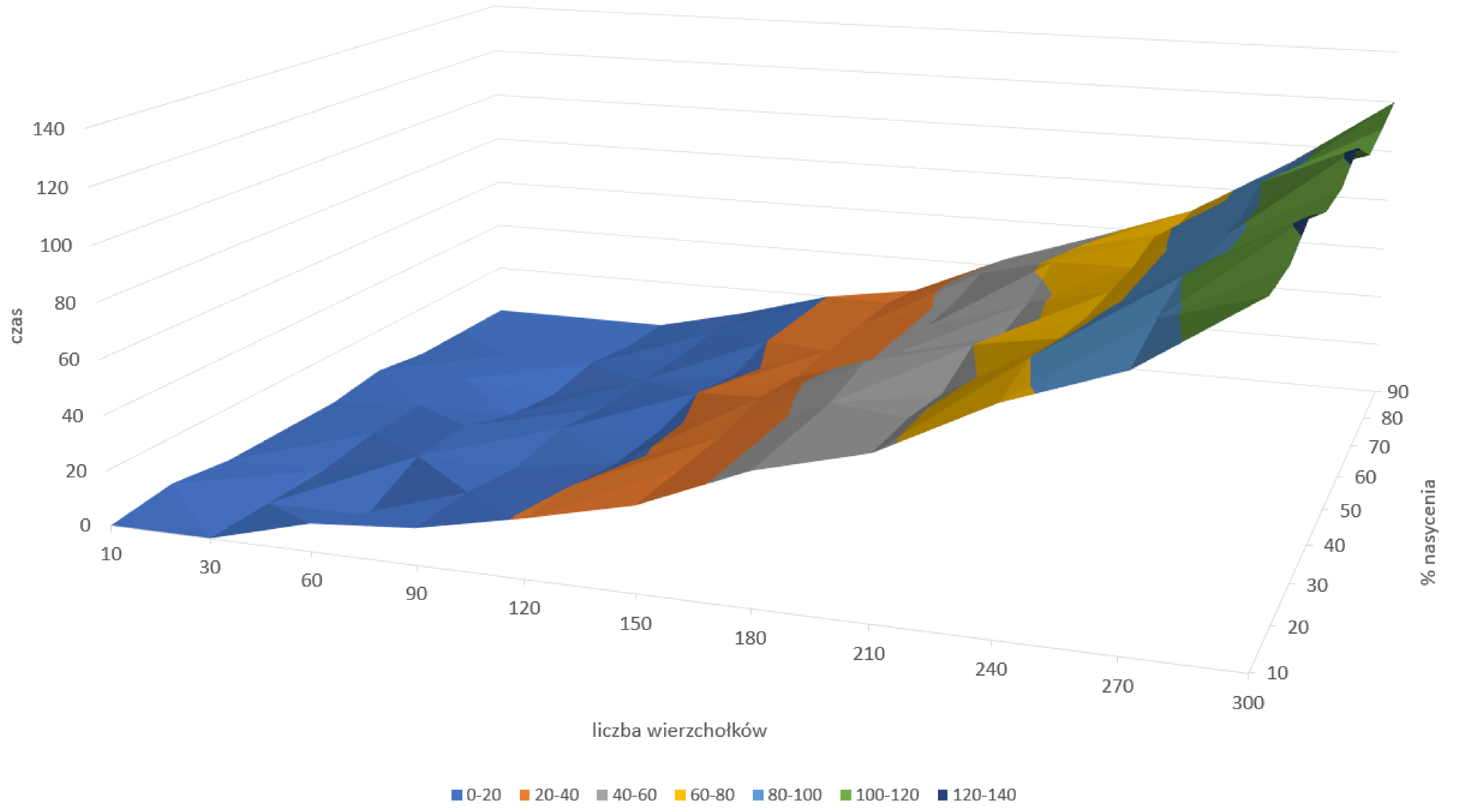
4.1) Zależność czasu poszukiwania cyklu Hamiltona w grafie skierowanym od liczby wierzchołków n i nasycenia s.



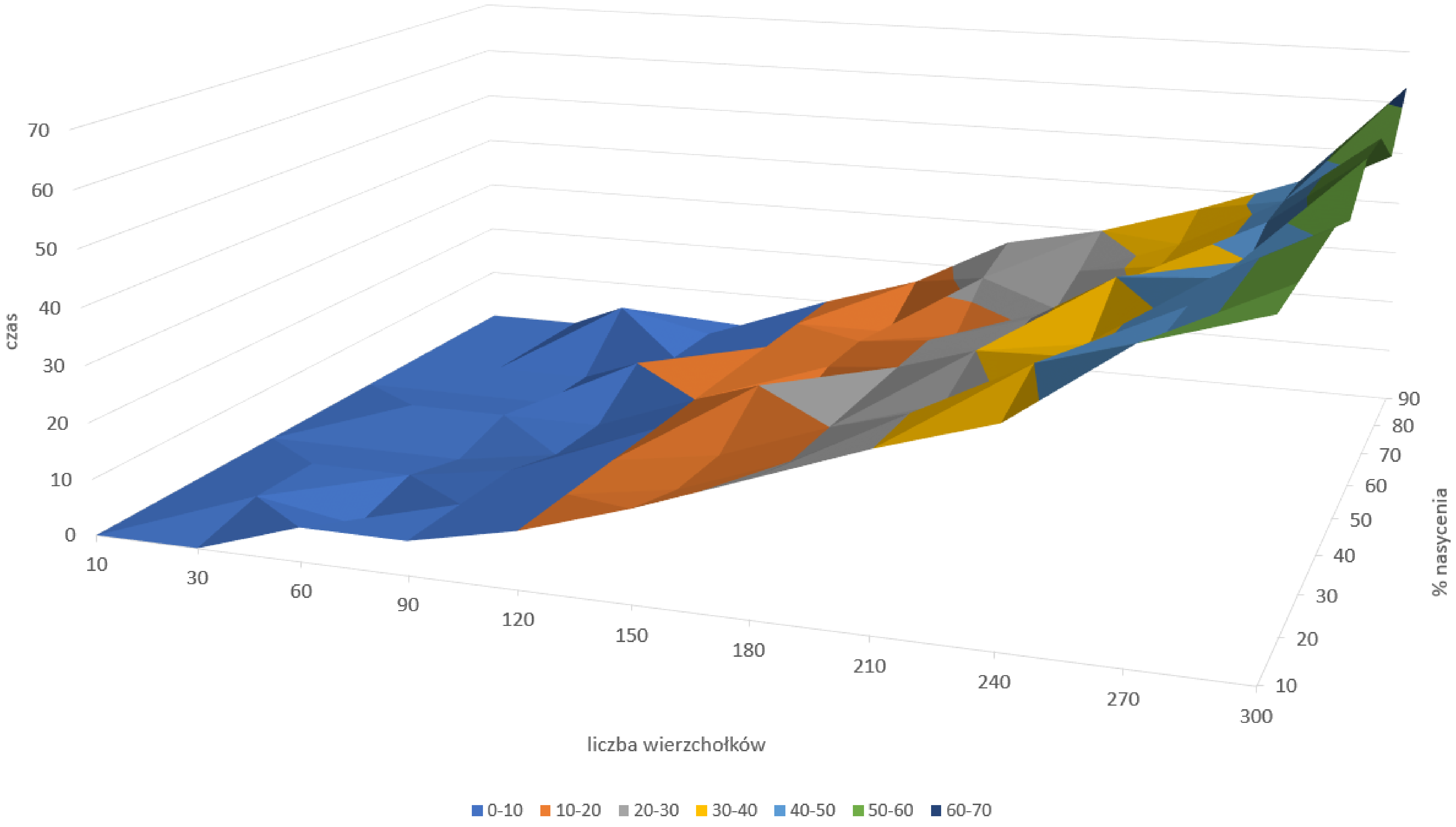
4.2) Zależność czasu poszukiwania cyklu Hamiltona w grafie nieskierowanym od liczby wierzchołków n i nasycenia s.



4.3) Zależność czasu poszukiwania cyklu Eulera w grafie skierowanym od liczby wierzchołków n i nasycenia s.



4.4) Zależność czasu poszukiwania cyklu Eulera w grafie nieskierowanym od liczby wierzchołków n i nasycenia s.



5) Podsumowanie.

5.1) Algorytm szukający ścieżki Hamiltona w grafie ma złożoność wykładniczą, a klasa złożoności samego problemu to silnie NP-trudne.

5.2) Algorytm szukający ścieżki Hamiltona w grafie ma złożoność wielomianową, a problem należy do klasy problemów P.

5.3) W przypadku poszukiwania cyklu Hamiltona najlepszym wyborem sposobu reprezentacji grafu jest macierz sąsiedztwa, ze względu na łatwość w przechodzeniu wierzchołków grafu. W przypadku algorytmu poszukującego cyklu Eulera, którego zadaniem jest przechodzenie krawędzi, optymalną opcją będzie reprezentacja w postaci listy następników dla grafu skierowanego, i listy krawędzi dla grafu nieskierowanego.