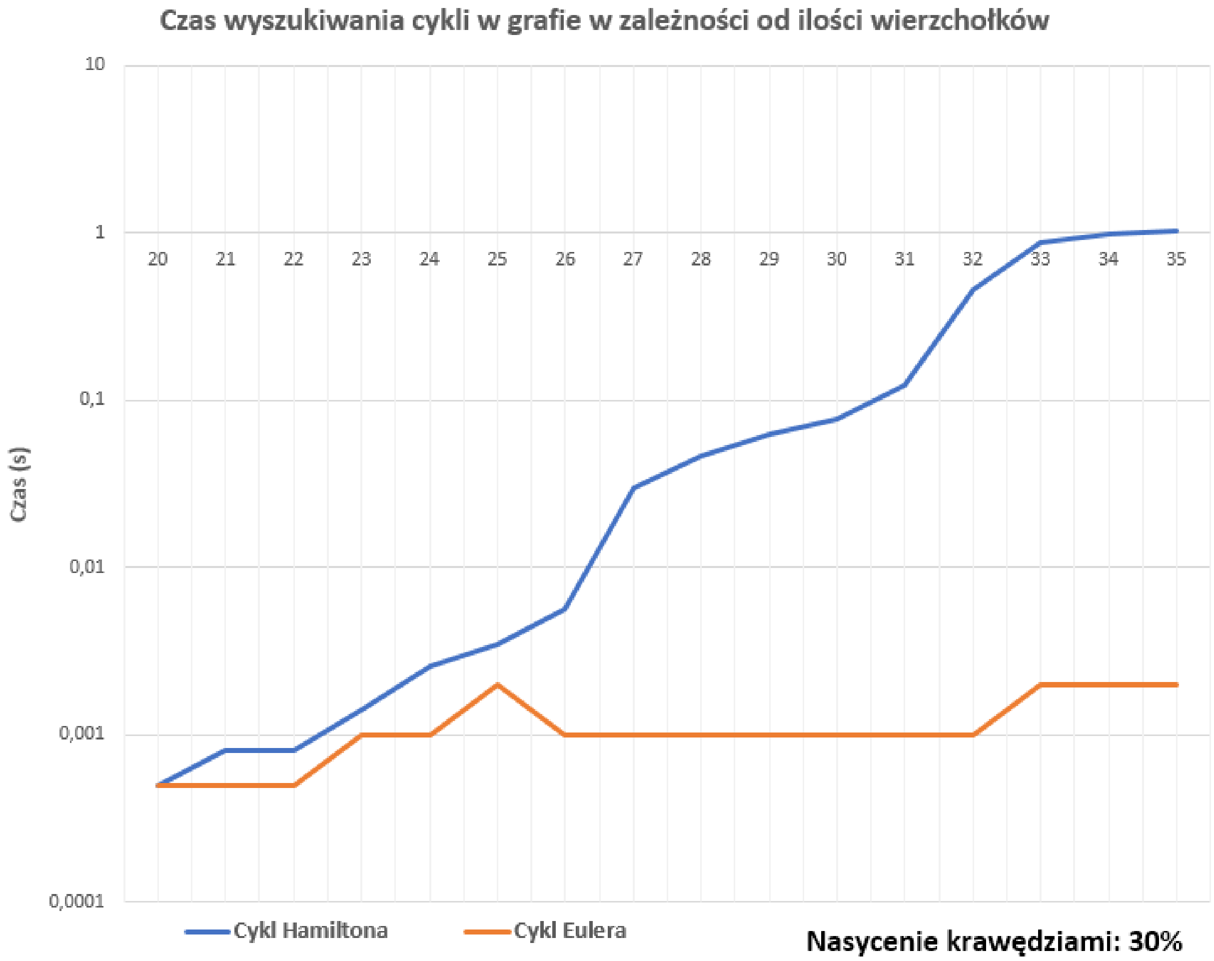
Zadanie 4. – Algorytmy z powracaniem – cykle Eulera i Hamiltona

Grupa I1, Jan Techner (132332), Sebastian Maciejewski (132275)

Wykresy zostały zrealizowane w skali logarytmicznej, z powodu konieczności porównania na jednym wykresie danych ze stosunkowo szerokiego przedziału. Na osi oY oznaczono czas działania algorytmu w sekundach, zaś na osi oX ilość wierzchołków grafu, w którym poszukiwane są cykle.

Pierwszym rozważanym problemem jest poszukiwanie cyklu Eulera oraz cyklu Hamiltona w grafie spójnym, nieskierowanym, będącym grafem Eulerowskim, o czym zapewnia parzysty stopień każdego z wierzchołków generowanych grafów. Pomiary wykonano dla dwóch wartości nasycenia grafu krawędziami – zacznijmy od nasycenia 30%.

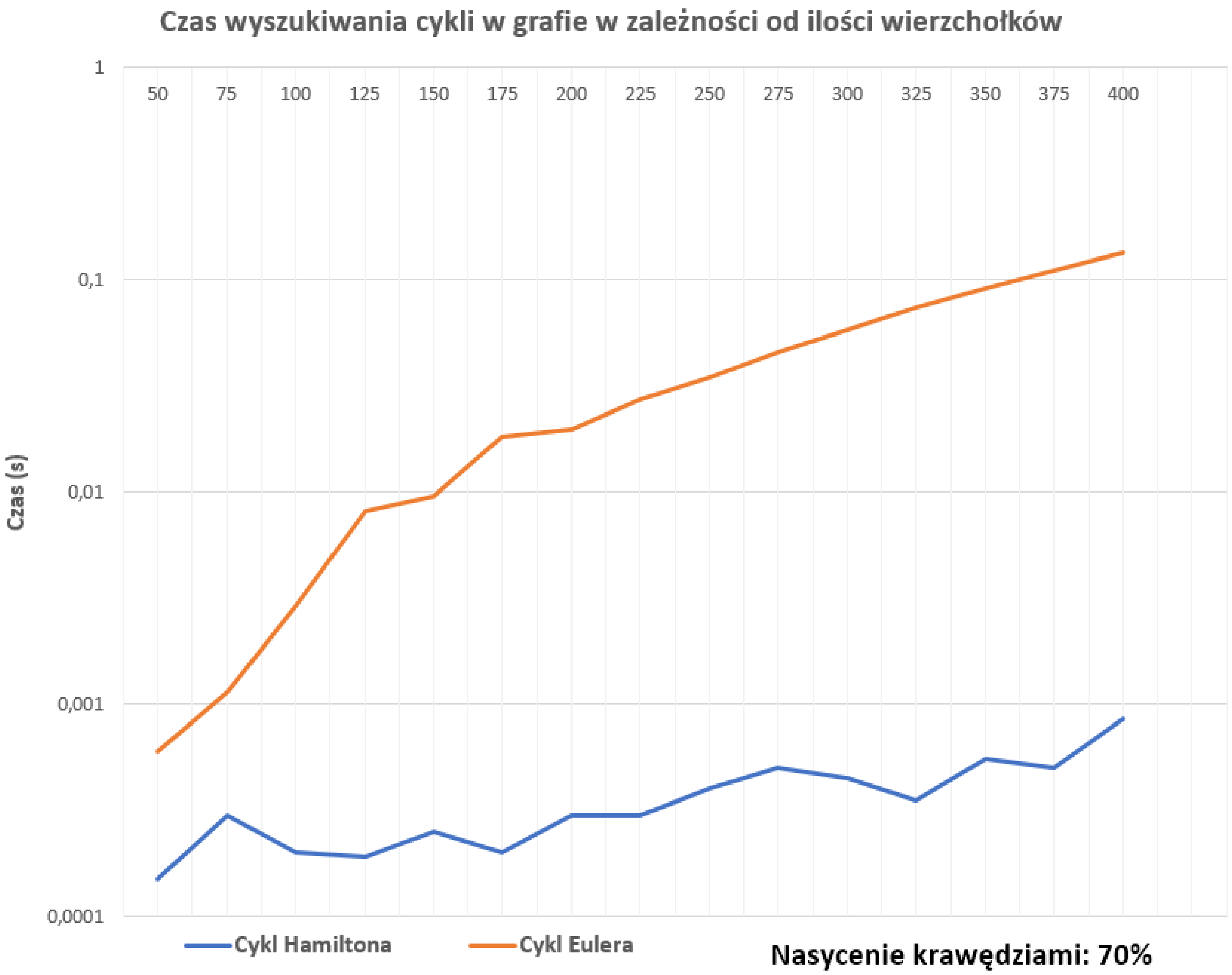


Jak widać z wykresu, wyszukiwanie cyklu Hamiltona dla takiego grafu zajmuje więcej czasu, a różnica na korzyść algorytmu znajdowania cyklu Eulera wzrasta wraz ze wzrostem liczby wierzchołków. Jest to spowodowane głównie tym, że algorytm wyszukiwania cyklu Hamiltona działa na zasadzie znajdowania pierwszej krawędzi łączącej dany wierzchołek z innym, co powoduje, że dla małej ilości krawędzi w grafie, algorytm ma znacznie więcej wierzchołków do przeszukania (aby znaleźć pierwszą krawędź). Dla tak małej ilości połączeń pomiędzy wierzchołkami doskonale widać istotną cechę tego algorytmu: jest wrażliwy na skrajnie niekorzystne przypadki, których prawdopodobieństwo wystąpienia jest w grafie o nasyceniu 30% znacznie wyższe.

Łatwo jest sobie uzmysłowić, że problem poszukiwania cyklu Hamiltona w grafie jest problemem skomplikowanym. Metoda, z której korzystaliśmy podczas przeszukiwania grafu jest w zasadzie metodą brute-force, która wprawdzie prowadzi do znalezienia cyklu, ale czas jej działania nie jest wielomianowy (mimo zastosowania przeszukiwania grafu w głąb (DFS)). Problem poszukiwania cyklu Hamiltona jest zatem problemem NP-zupełnym, ponieważ nie istnieje rozwiązanie tego problemu w czasie wielomianowym (pesymistyczna złożoność to aż O(n!)).

Nieco inaczej wygląda algorytm poszukiwania cyklu Eulera. Wykres ukazuje nam, że nie dość, iż czas działania algorytmu wyszukiwania cyklu Eulera jest krótszy, to jeszcze charakteryzuje go dosyć wysoka stabilność (nie ma gwałtownych wzrostów czasu potrzebnego do znalezienia cyklu dla kolejnych ilości wierzchołków). Dzieje się tak za sprawą prostoty algorytmu poszukiwania cyklu Eulera – przechodzi on po kolejnych krawędziach grafu usuwając je i dodając odwiedzone wierzchołki do stosu obrazującego cykl. Jego zasada działania również wykorzystuje DFS. Problem poszukiwania cyklu Eulera w grafie okazuję się być problemem P, czyli takim, który można rozwiązać w czasie wielomianowym (złożoność $O(n^2)$), co stanowi główną różnicę pomiędzy algorytmami poszukiwania cykli Eulera i Hamiltona.

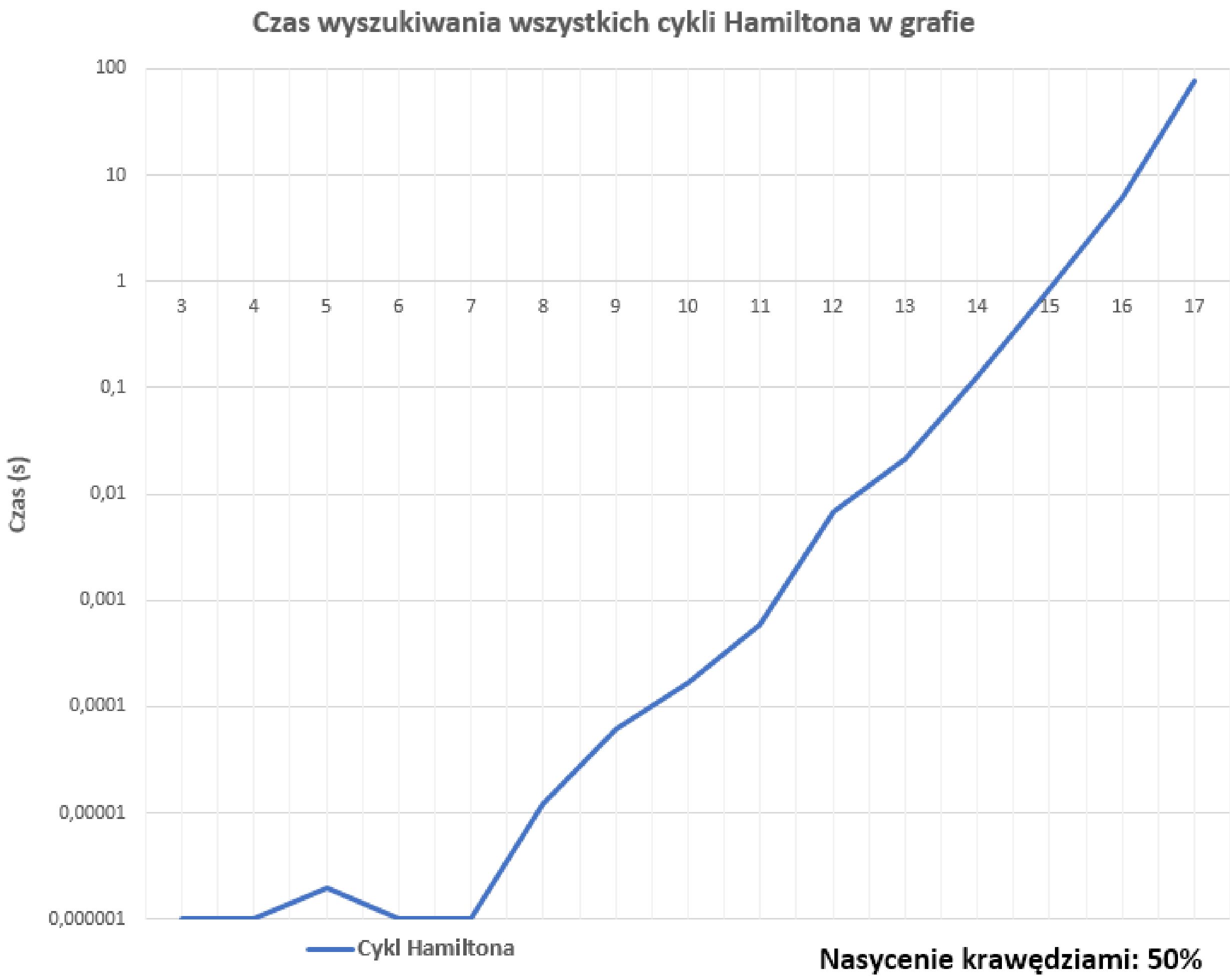
Spójrzmy teraz na drugi wykres – dla grafu o nasyceniu krawędziami 70%.



Już na pierwszy rzut oka widać, że wykres dla grafu o nasyceniu 70% znacznie różni się od swojego odpowiednika dla grafu o mniejszym nasyceniu. Szczególnie istotną różnicą jest znacznie szybsze działanie algorytmu wyszukiwania cyklu Hamiltona (nawet dla grafu o 400 wierzchołkach!). Różnica jest do tego stopnia znacząca, że algorytm ten działa nawet szybciej niż algorytm wyszukiwania cyklu Eulera, co wygląda co najmniej dziwnie biorąc pod uwagę klasy złożoności tych problemów.

Jednak za tym szokującym rezultatem stoi coś, o czym powinniśmy pamiętać – duża ilość krawędzi działa na korzyść algorytmu wyszukiwania cyklu Hamiltona, a na niekorzyść tego, który wyszukuje cykl Eulera. Algorytm wyszukiwania cyklu Hamiltona błyskawicznie znajdzie krawędź do najbliższego wierzchołka, można powiedzieć, że ma wiele możliwych dróg dojścia do każdego z wierzchołków, co skutkuje błyskawicznym działaniem tego algorytmu. W myśl tego, co pisaliśmy wcześniej: algorytm znajdowania cyklu Eulera, przy większym nasyceniu krawędziami musi przejść znacznie więcej krawędzi, co istotnie wydłuża jego czas działania. Stąd taki a nie inny układ krzywych na wykresie.

Zajmijmy się teraz drugim problemem – problemem znalezienia wszystkich cykli Hamiltona w danym grafie. Zależność czasu, jaki zajmuje znalezienie wszystkich cykli, od ilości wierzchołków grafu o nasyceniu 50% przedstawiono na poniższym wykresie.



Od razu rzuca się w oczy drastyczna tendencja wzrostowa, która charakteryzuje ten wykres. Drobne fluktuacje dla początkowych ilości wierzchołków są wynikiem dużej losowości generowanych grafów. Różnica pomiędzy pierwszym i ostatnim punktem osi oX wynosi na osi oY aż siedem rzędów wielkości. Jest to pośrednio wynikiem tego, o czym pisaliśmy już wcześniej – problem wyszukiwania cykli Hamiltona jest bardzo skomplikowany, jednak w tym przypadku daje się to mocniej we znaki, ponieważ interesują nas wszystkie (!) cykle Hamiltona istniejące w grafie. Zadanie wyszukania wszystkich cykli w grafie jest przypadkiem, który doskonale ukazuje dlaczego poszukiwanie cykli Hamiltona jest problemem NP-trudnym. Ciężko jest oszacować ilość zagnieżdżeń rekurencji, których musi dokonać algorytm w naszej implementacji (DFS) ale wykres jest sam w sobie doskonałym obrazem tego, jak bardzo niedoskonałe są znane nam algorytmy, gdy zestawić je z trudnymi problemami.