

Wyniki testów przy różnych parametrach

Parametry, które można modyfikować:

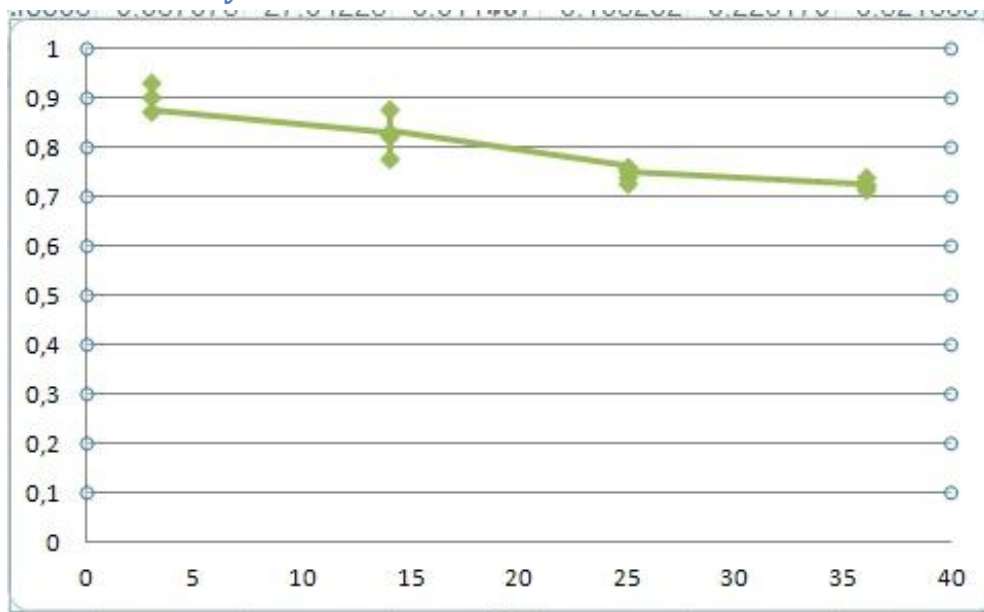
- Sąsiedzi – maksymalna liczba sąsiadów. Z tego przedziału jest losowana wartość atrybutu członków populacji startowej.
- Klastry – maksymalna wielkość Klastra. Z tego przedziału jest losowana wartość atrybutu członków populacji startowej.
- Medoid – wielkość wektora medoidów.
- Populacja – wielkość populacji ewolucji.
- Kroki - ilość kroków ewolucji.

Testy przeprowadzono na trzech bazach: Iris, Wine oraz Cancer (WPBC).

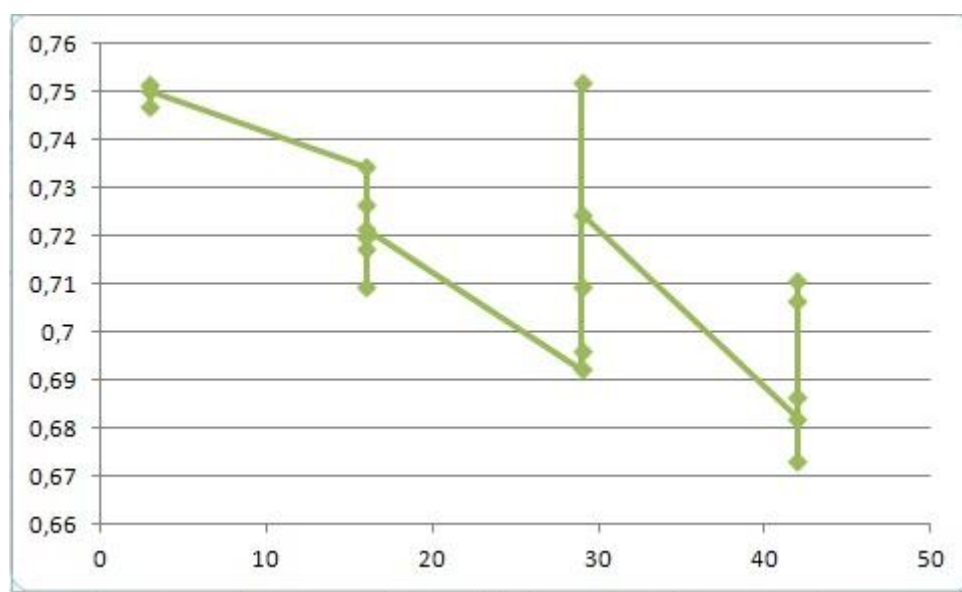
Wyniki testów oceniane są za pomocą trzech miar: Rand, BDI oraz DI

1 Rand

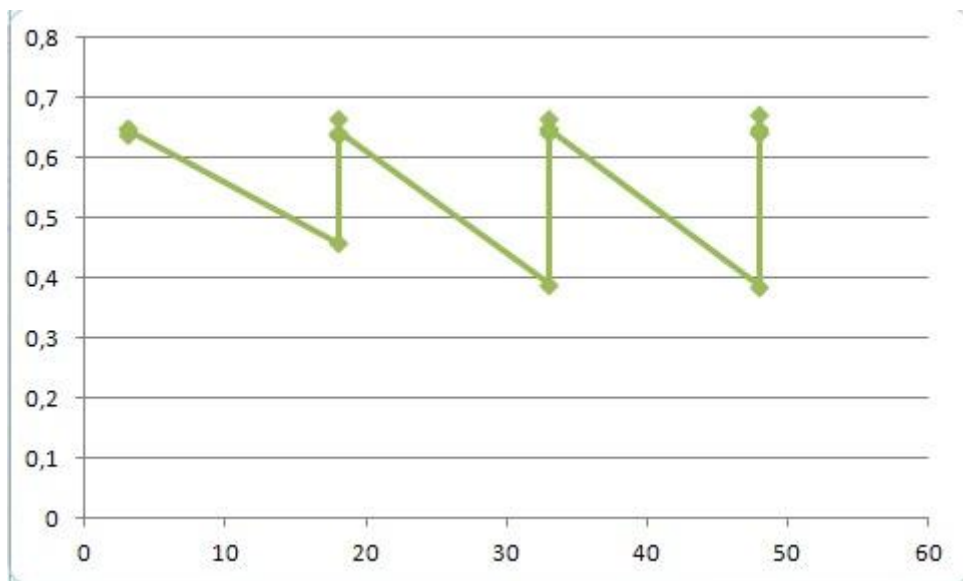
1.1 Rand vs Medoidy:



rys 1 – [Iris] Rand (rzędna) vs Medoidy (odcięta)



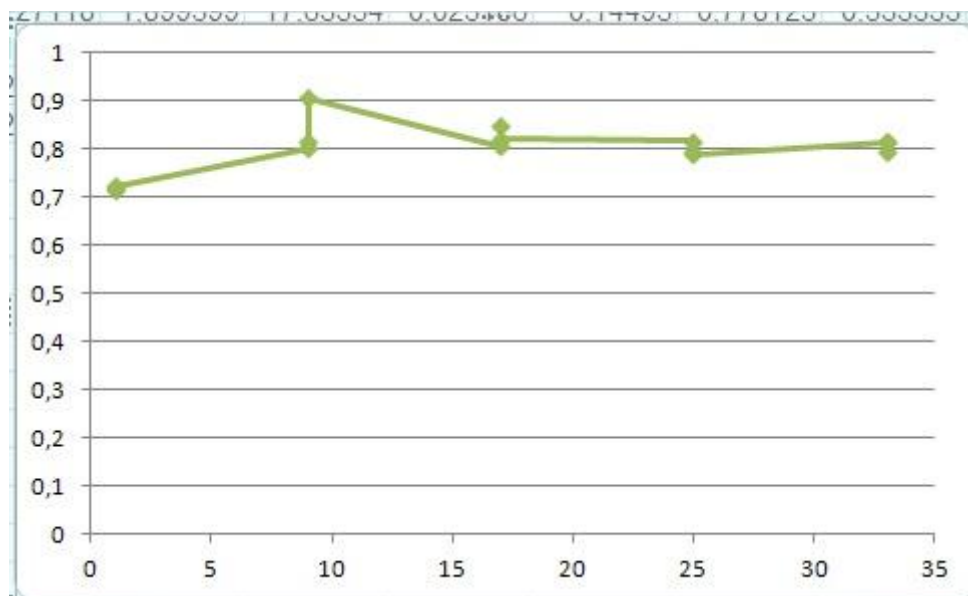
rys 2 - [Wine] Rand (rzędna) vs Medoidy (odcięta)



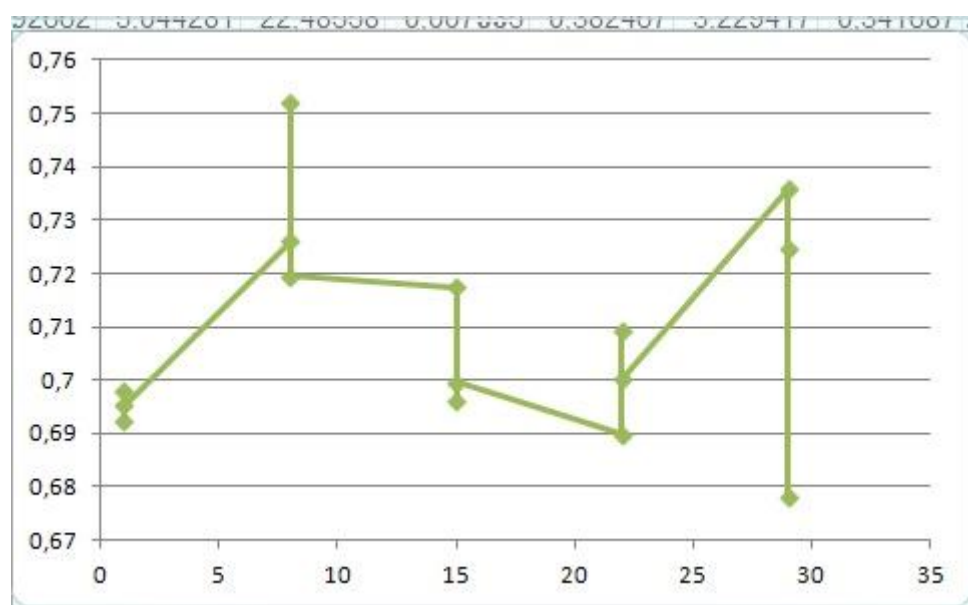
rys 3 - [Cancer] Rand (rzędna) vs Medoidy (odcięta)

W przypadku **Iris** Wynik ewolucji wyraźnie maleje wraz z zwiększeniem medoidów. W przypadku **Wine**, Rand także maleje ale jest także bardzo szeroko rozłożony zależnie od maksymalnej sąsiadów. W przypadku **Cancer** ilość medoidów ma natomiast bardzo mały wpływ na wynik.

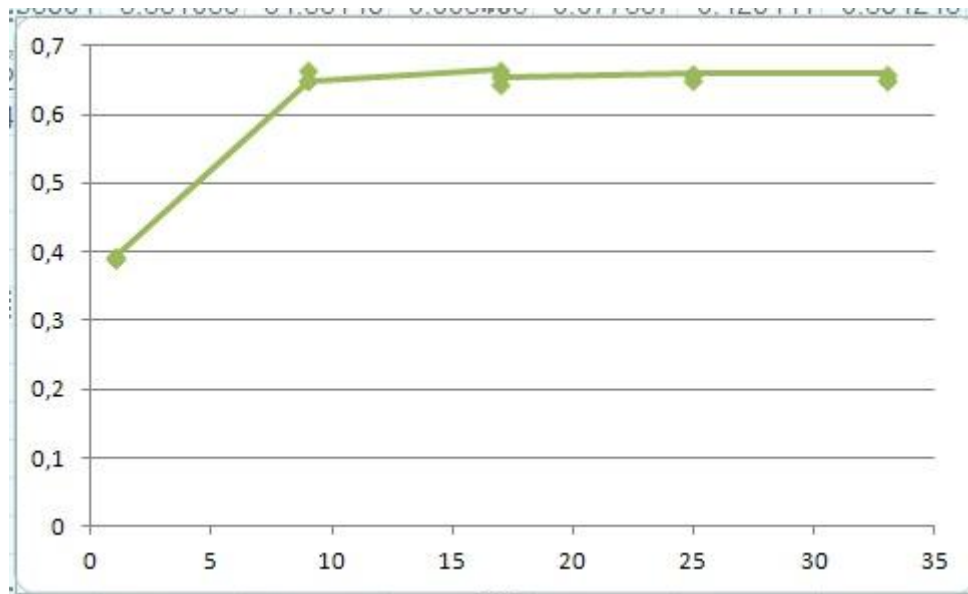
1.2 Rand vs Klaster



rys 4 - [Iris] Rand vs Klaster



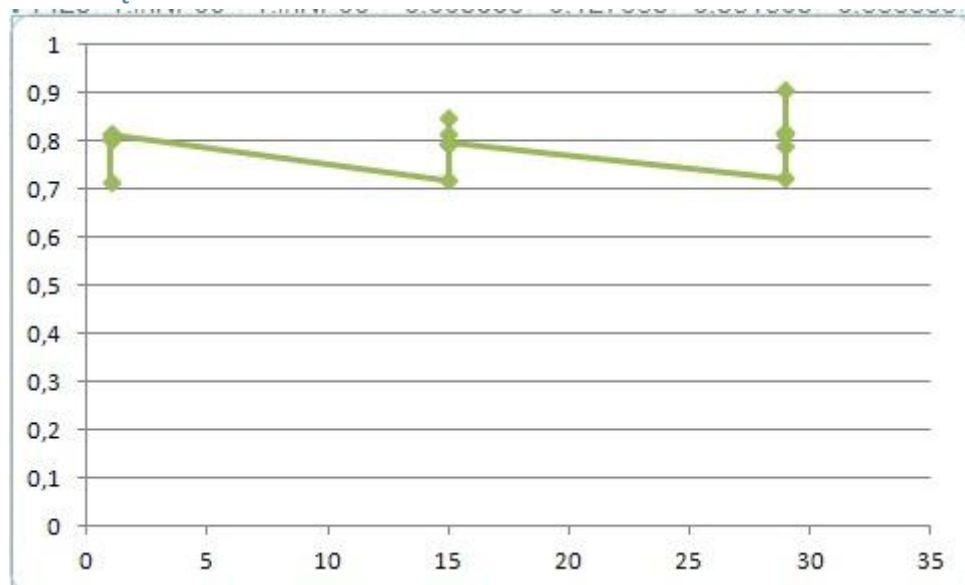
rys 5 - [Wine] Rand vs Klaster



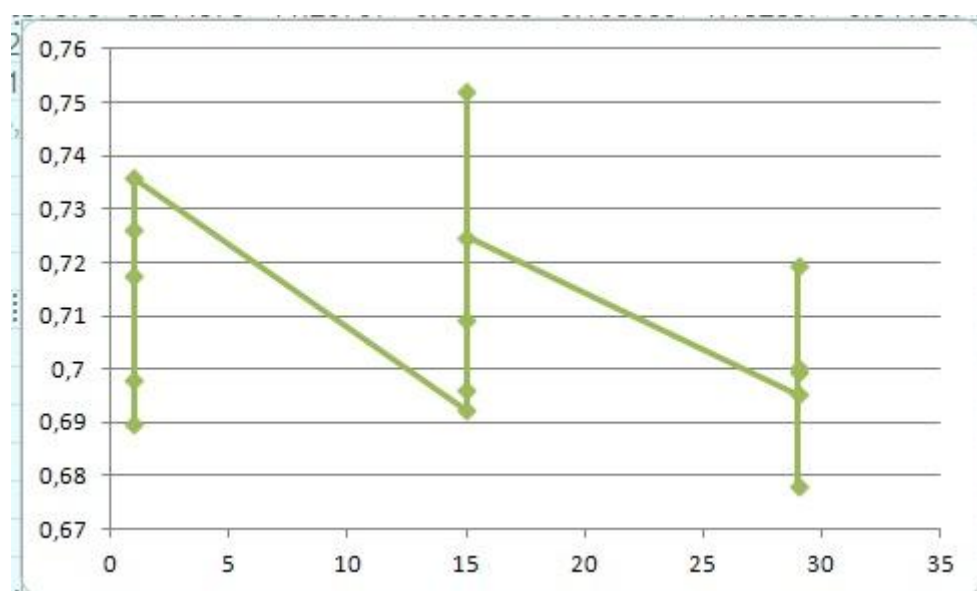
rys 6 - [Cancer] Rand vs Klaster

Choć dla każdej bazy maksymalna wielkość klastra ma widoczny wpływ. To we wszystkich trzech przypadkach zauważalna jest znaczna poprawa wyniku w okolicy $1/3$ aktualnego rozmiaru wektoru medoidów.

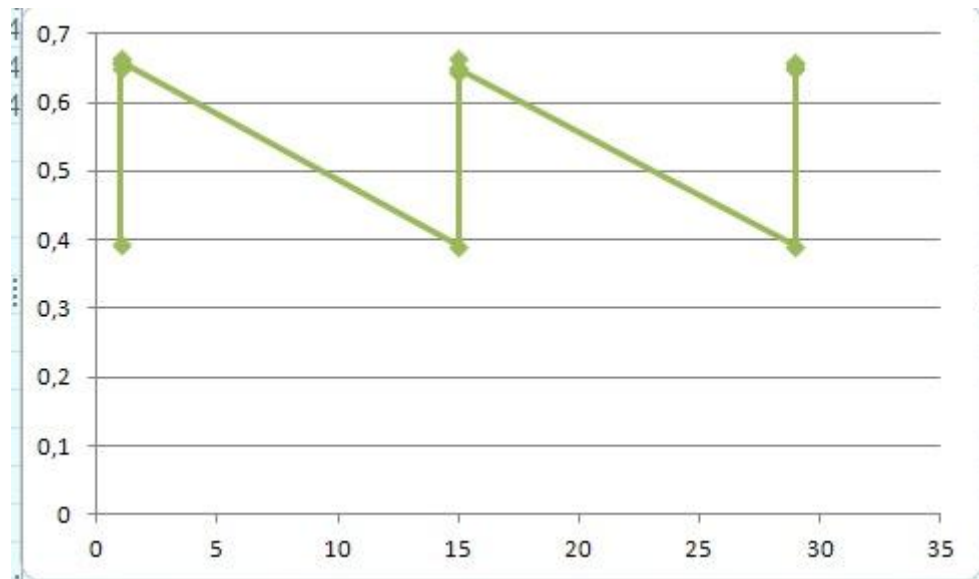
1.3 Rand vs Sąsiedzi



rys 7 [Iris] Rand vs Sąsiedzi



rys 8 - [Wine] Rand vs Sąsiedzi

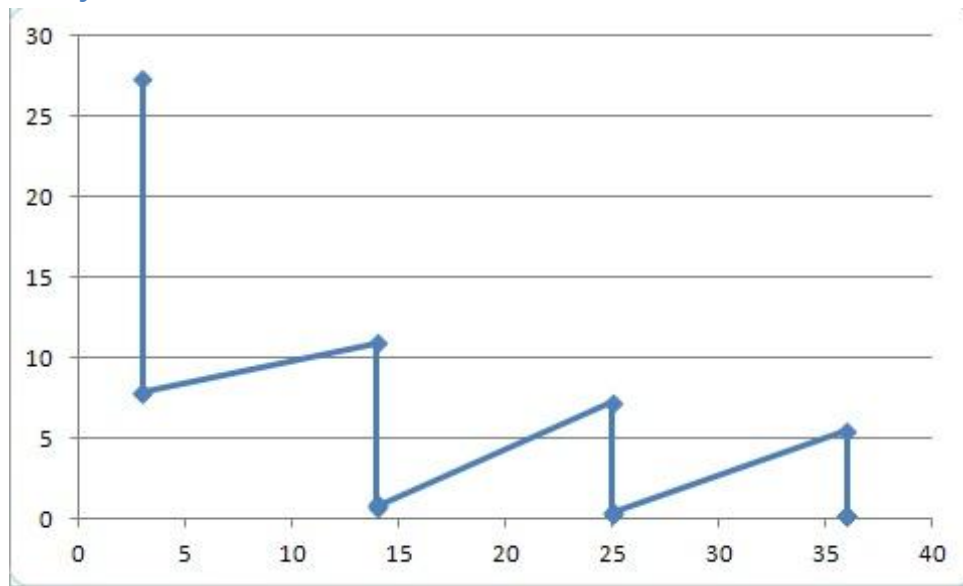


rys 9 - [Cancer] Rand vs Sąsiedzi

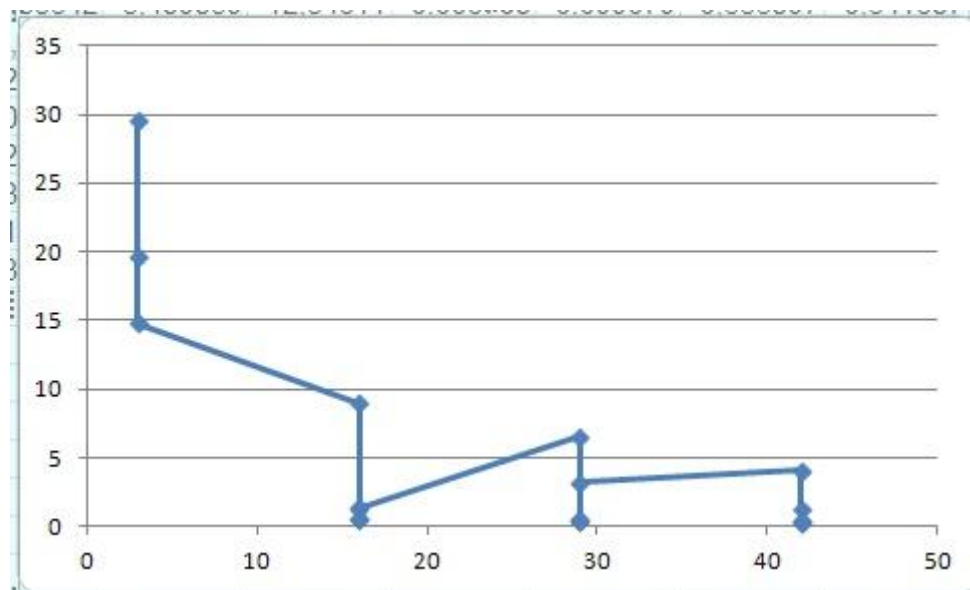
Na powyższych diagramach widać różny wpływ maksymalnej liczby sąsiadów na każdą bazę. W przypadku Iris, wyniki troszkę poprawiają się przy większej ilości sąsiadów. Dla Wine wyraźnie dzieje się tak w połowie. Natomiast dla Cancer nie ma to najwyraźniej żadnego znaczenia.

2 BDI:

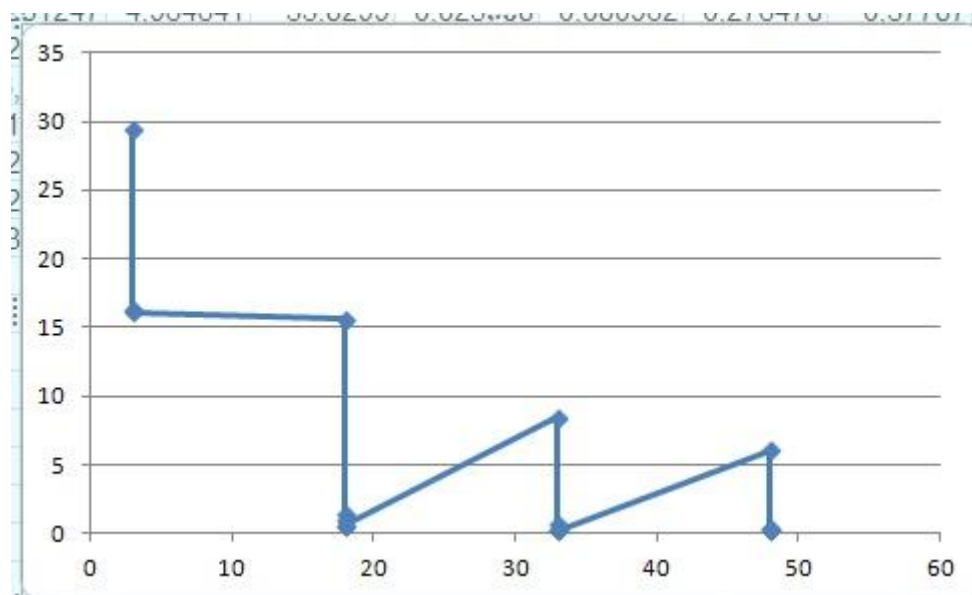
2.1 Medoidy vs BDI:



rys 10 - [Iris] BDI (rzędna) vs Medoidy (odcięta)



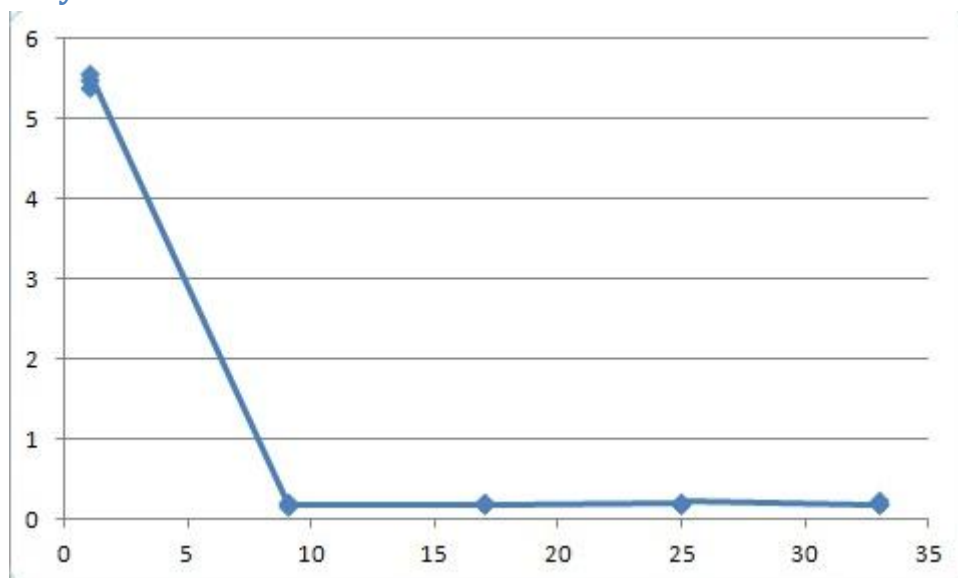
rys 11 - [Wine] BDI vs Medoidy



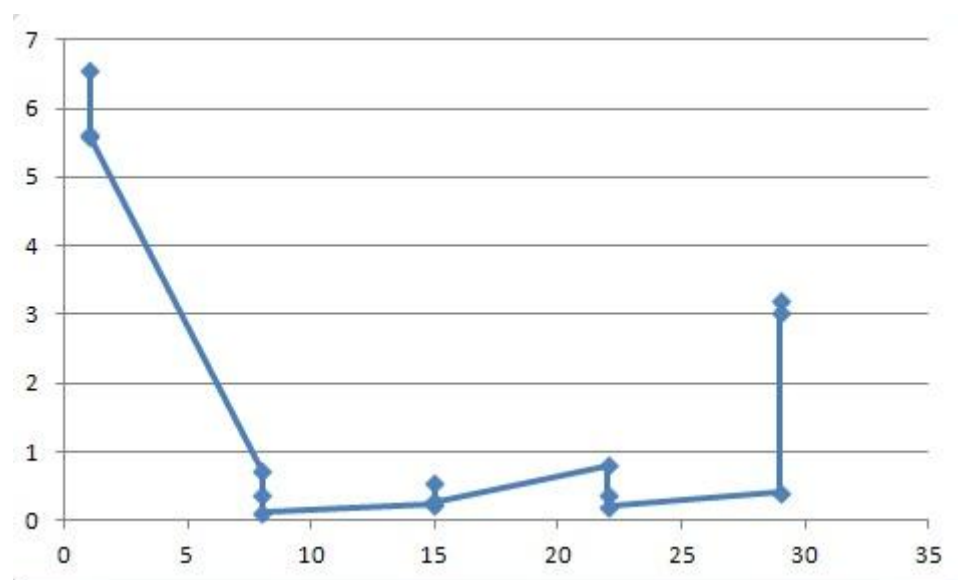
rys 12 - [Cancer] BDI vs Medoidy

We wszystkich trzech przypadkach wyraźnie widać, że zwiększanie ilości medoidów poprawia wynik (im mniejszy tym lepszy).

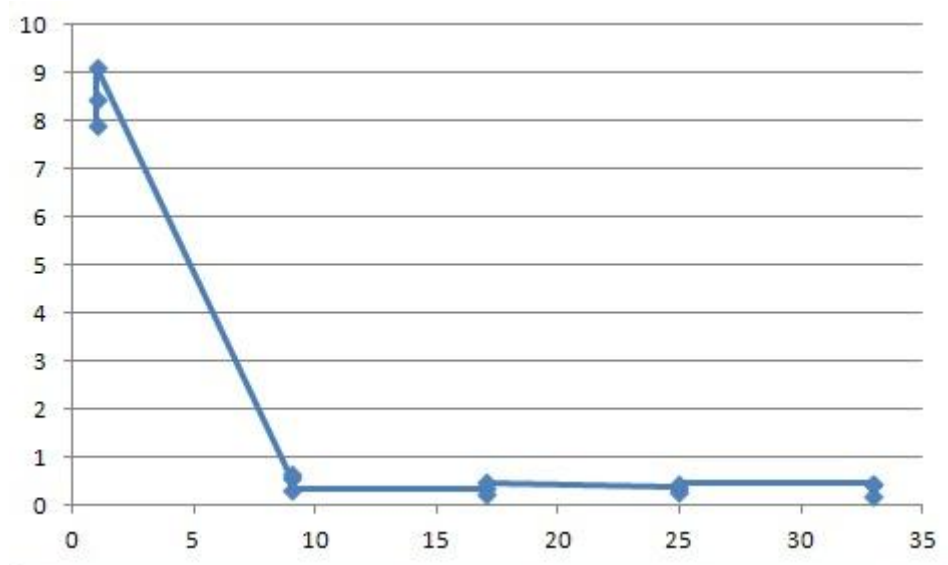
2.2 Klastry vs BDI:



rys 13 - [Iris] BDI (rzędna) vs Klastry (odcięta)



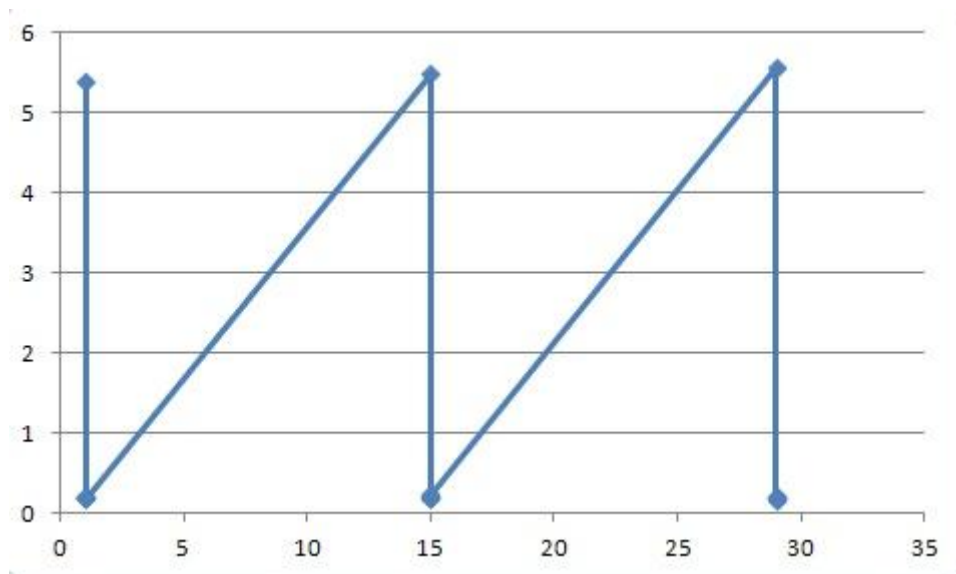
rys 14 - [Wine] BDI (rzędna) vs Klastry (odcięta)



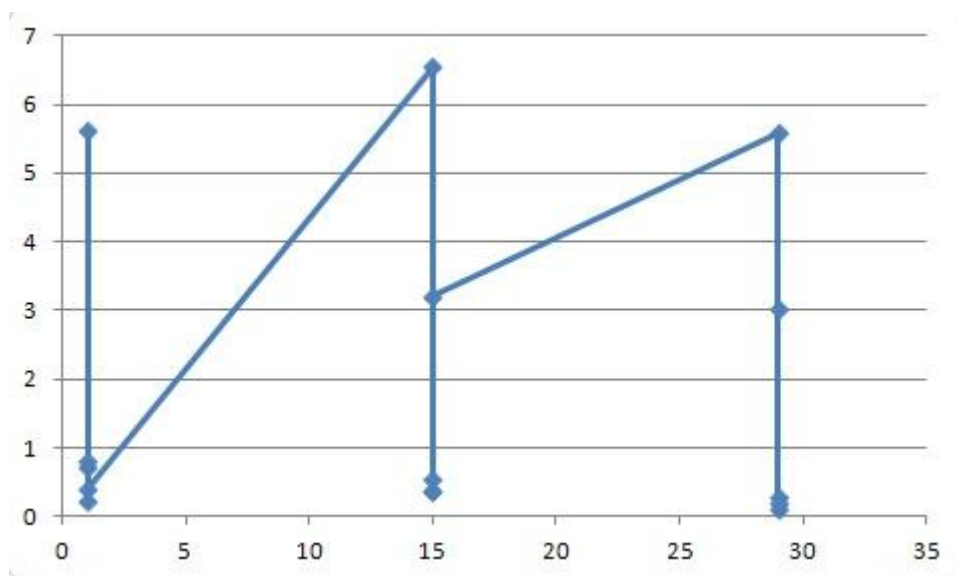
rys 15 - [Cancer] BDI (rzędna) vs Klastry (odcięta)

We wszystkich przypadkach najwyraźniej widać, że większy maksymalny rozmiar klastra wpływa pozytywnie na wynik. Wynik ten bardzo mocno poprawia się w 1/3 wartości klastra, a w przypadku wina jest tam nawet najlepszy.

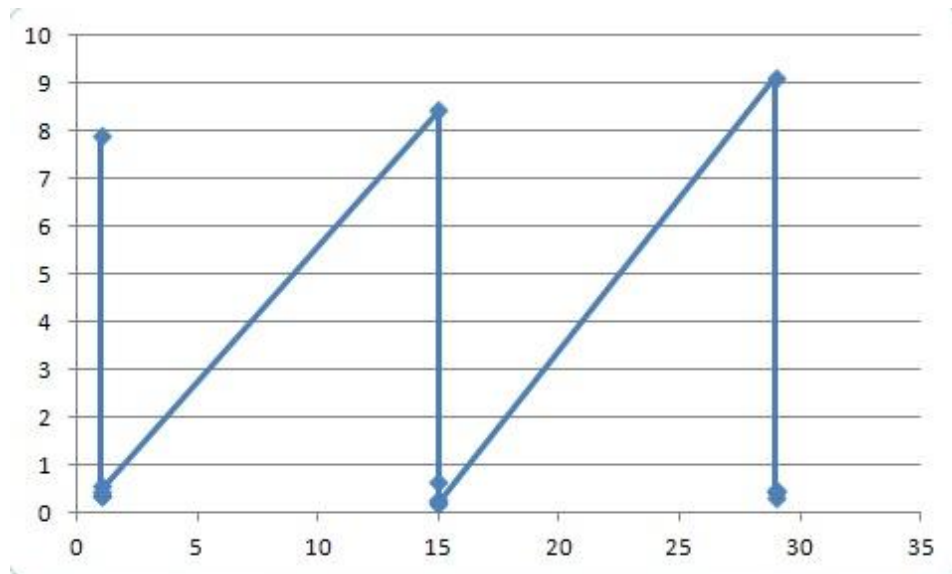
2.3 Sąsiedzi vs BDI



rys 16 - [Iris] BDI (rzędna) vs Sąsiedzi (odcięta)



rys 17 - [Wine] BDI (rzędna) vs Sąsiedzi (odcięta)

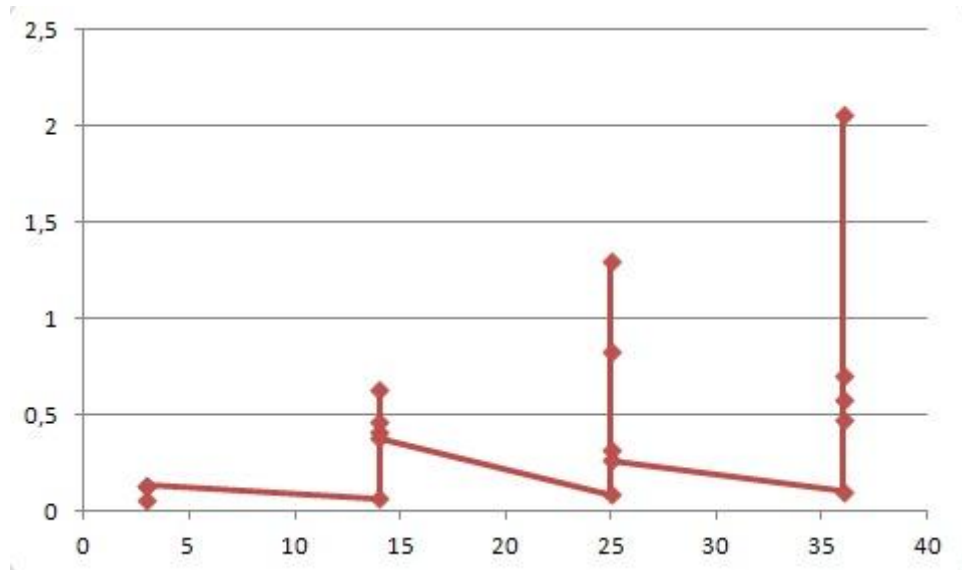


rys 18 - [Cancer] BDI (rzędna) vs Sąsiedzi (odcięta)

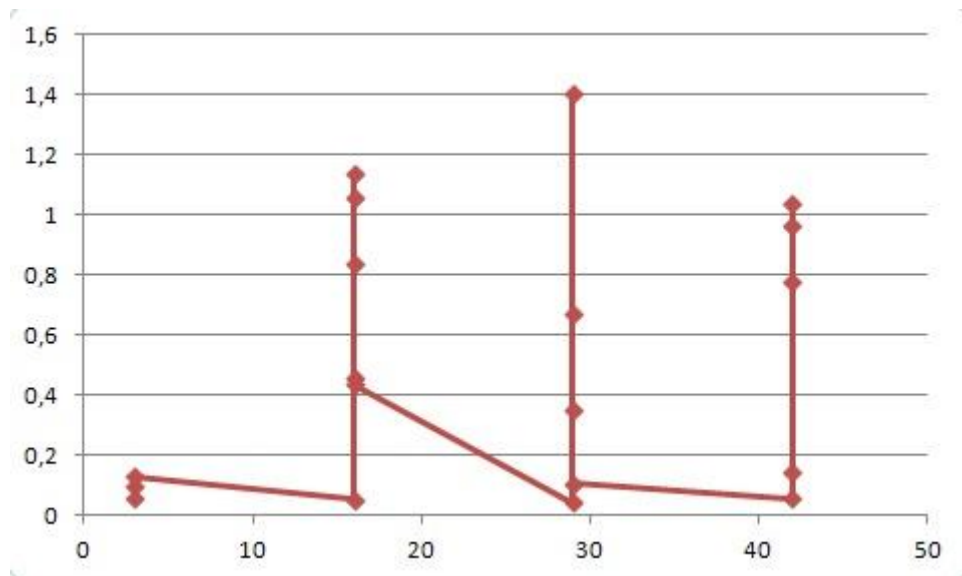
Maksymalna liczba sąsiadów ma minimalny wpływ na uzyskany wynik, a na pewno inny dla każdego przypadku. Dla Iris nie ma żadnego znaczenia. Dla Wine poprawia się nieznacznie dla wartości środkowej, a dla Cancer poprawia się wraz ze wzrostem.

3 DI

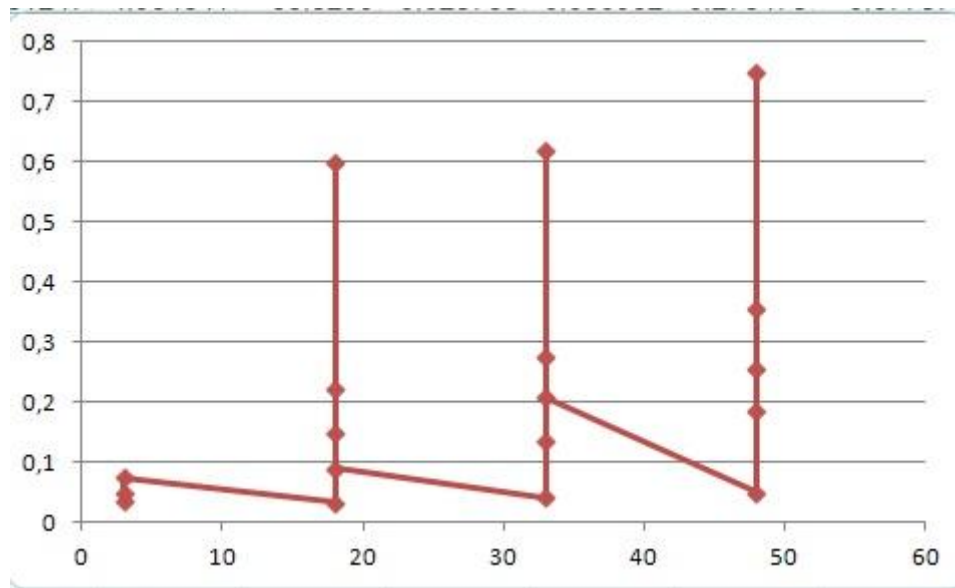
3.1 Medoidy vs DI



rys 19 - [Iris] DI (rzędna) vs Medoidy (odcięta)



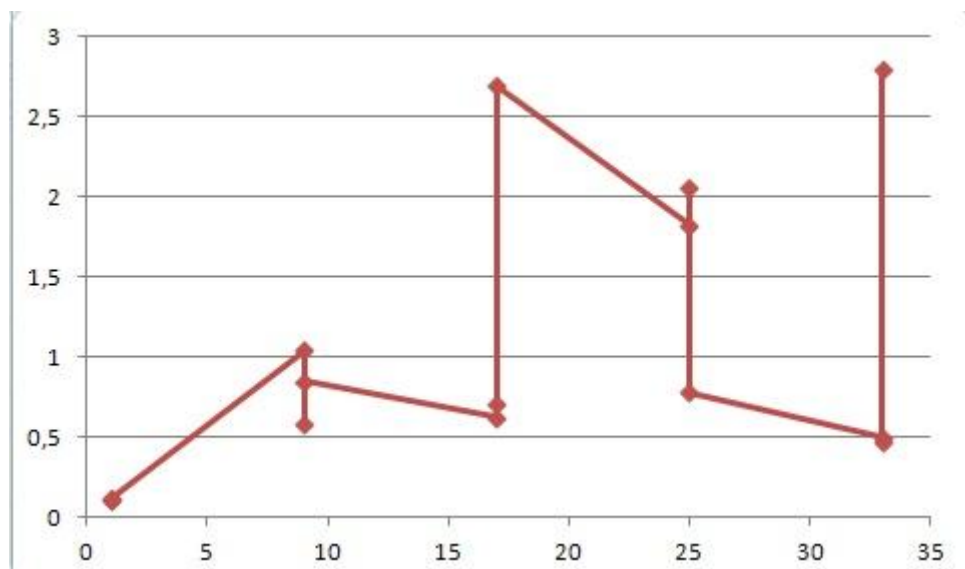
rys 20 - [Wine] DI (rzędna) vs Medoidy (odcięta)



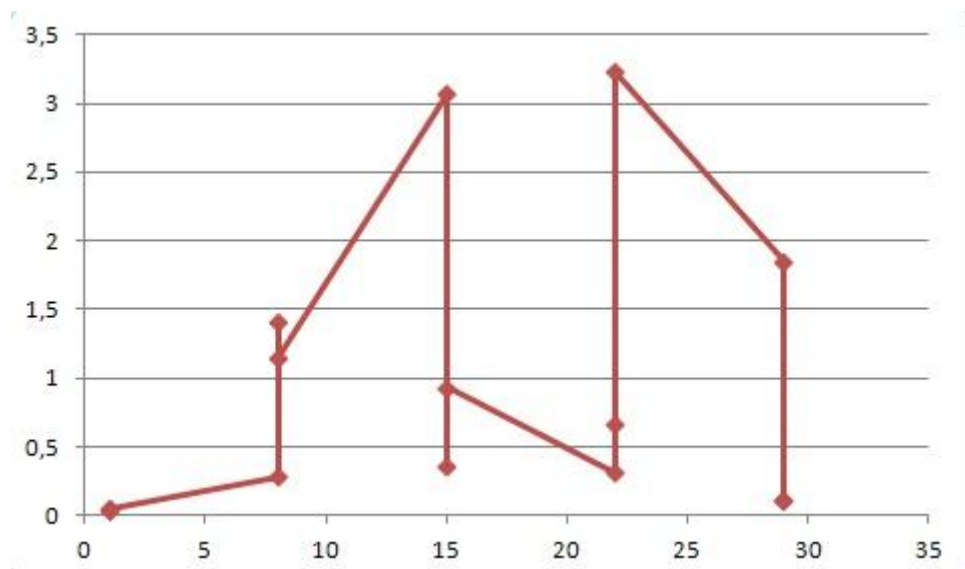
rys 21 - [Cancer] DI (rzędna) vs Medoidy (odcięta)

Widok poprawia się tutaj wraz ze wzrostem ilości medoidów. Poza **Wine**, gdzie najlepszy wynik uzyskany jest w 3/4. Dla **Iris** i **Cancer** najlepszy wynik uzyskiwany jest dla największej liczby medoidów.

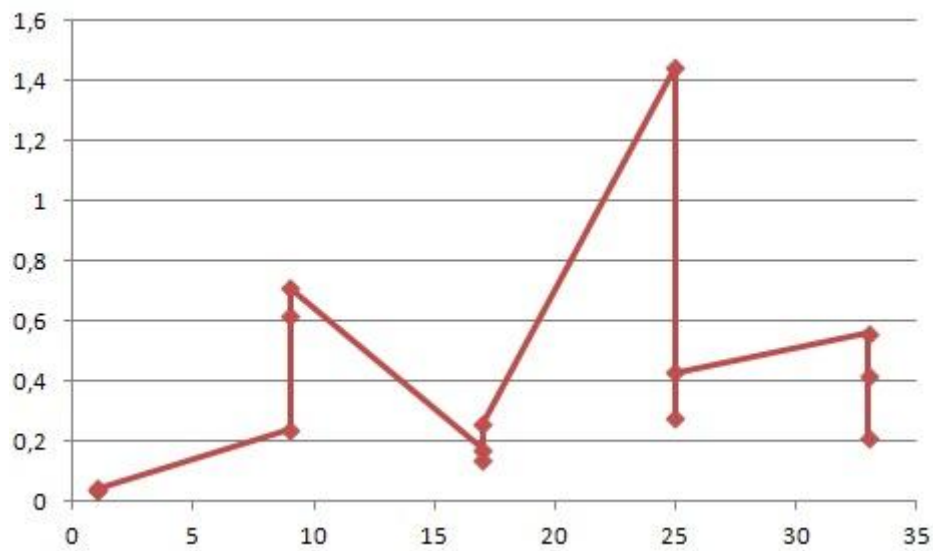
3.2 Klaster vs DI



rys 22 - [Iris] DI (rzędna) vs Klaster (odcięta)



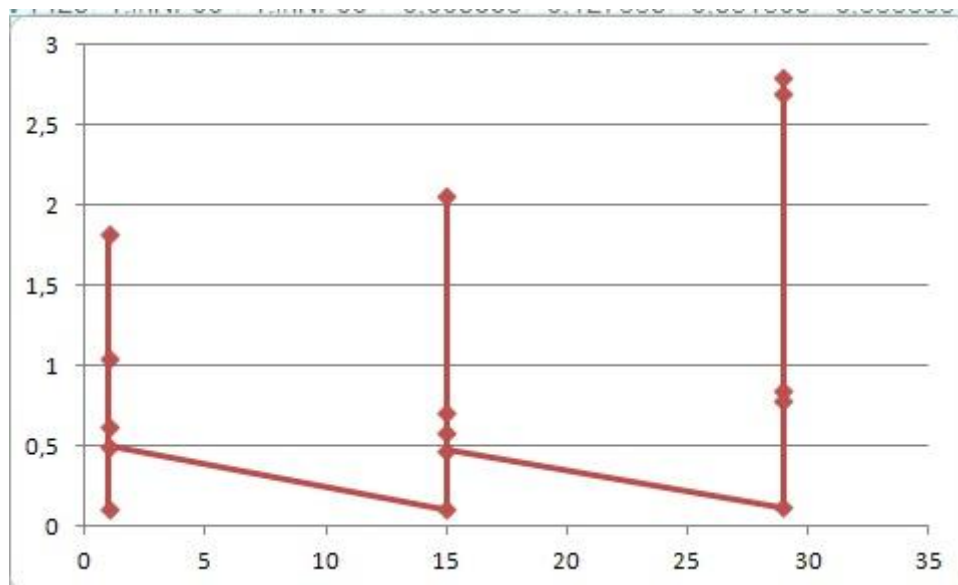
rys 23 - [Wine] DI (rzędna) vs Klaster (odcięta)



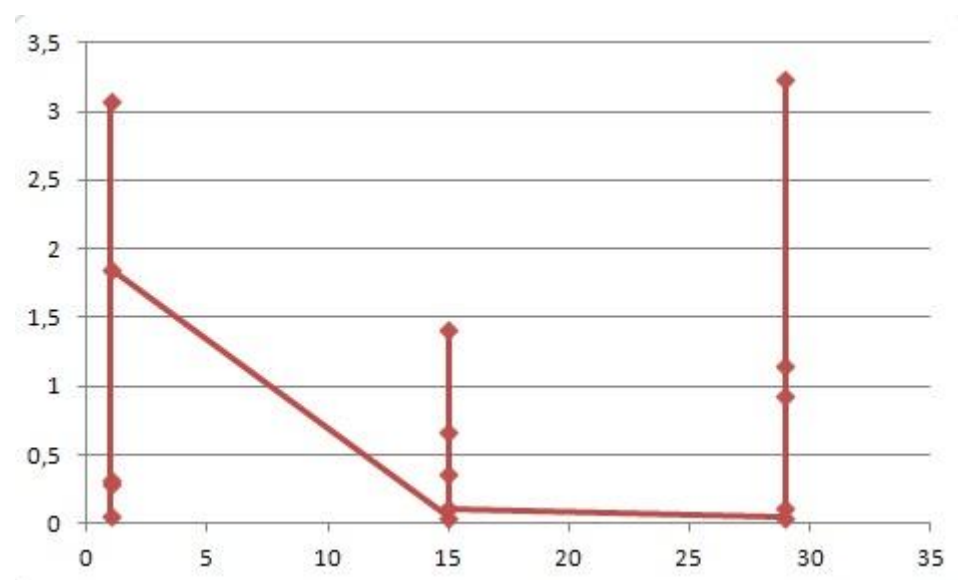
rys 24 - [Cancer] DI (rzędna) vs Klaster (odcięta)

Maksymalna wielość klastra ma widoczny różny wpływ zależnie od bazy. Mimo to dosyć znaczący. W przypadku Iris, najlepszy wynik uzyskiwany jest w połowie i przy klastrze wielkości wektora medoidów. W przypadku Wine, najlepszy wynik uzyskiwany jest tylko w połowie, dla Cancer w 3/4.

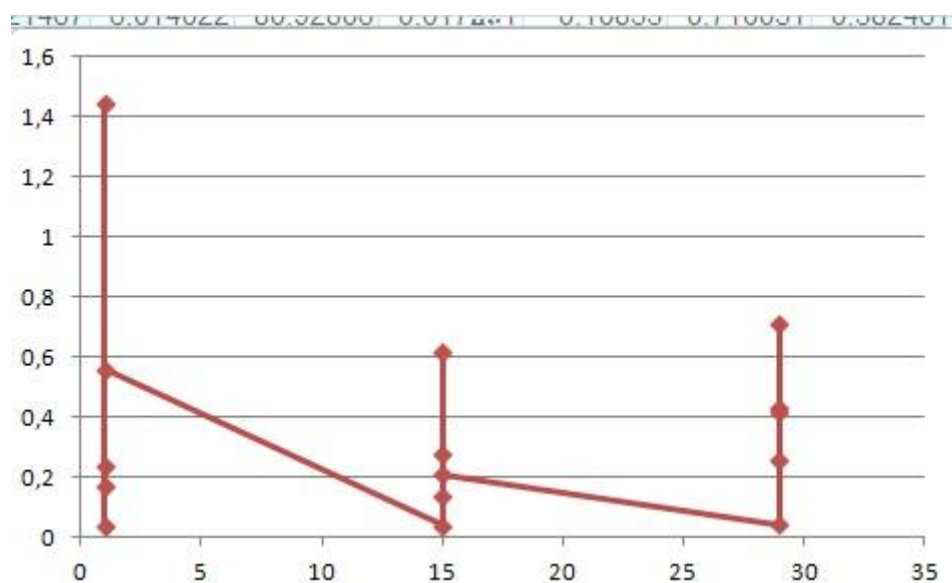
3.3 Sąsiedzi vs DI



rys 25 - [Iris] DI (rzędna) vs Sąsiedzi (odcięta)



rys 26 - [Wine] DI (rzędna) vs Sąsiedzi (odcięta)



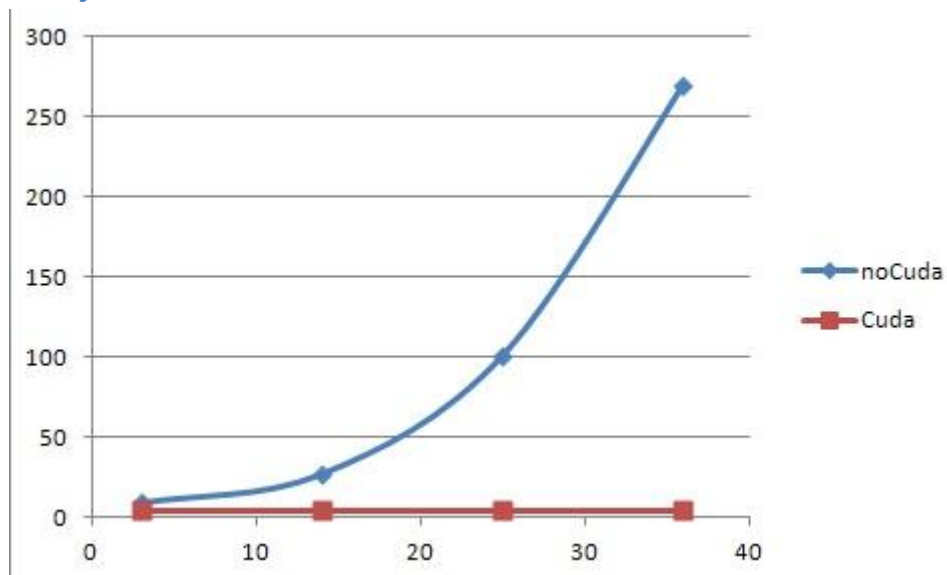
rys 27 - [Cancer] DI (rzędna) vs Sąsiedzi (odcięta)

Podobnie jak maksymalny rozmiar klastra, maksymalna ilość sąsiadów ma różny wpływ zależnie od bazy.

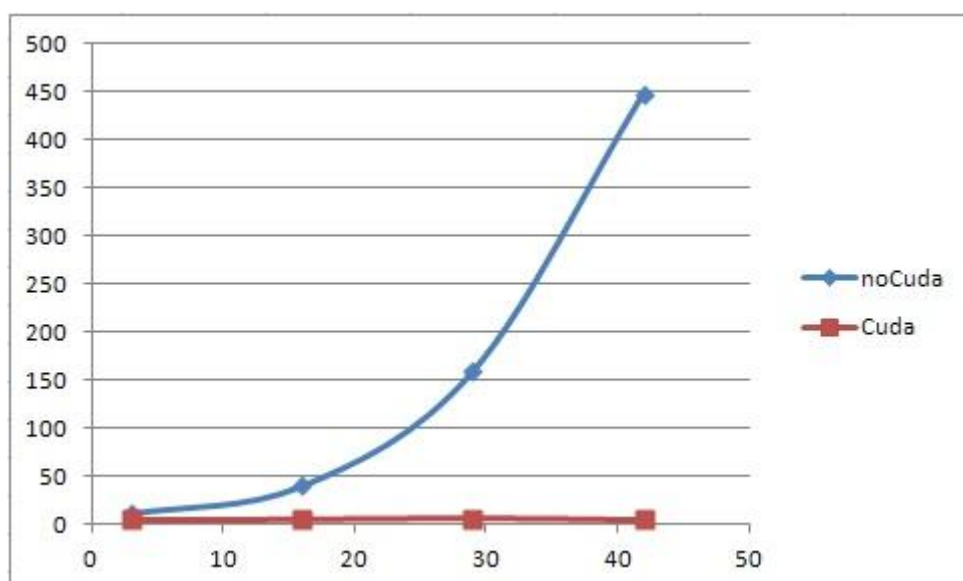
4 Wyniki testów porównawczych czasowych

Pomiarów dokonano w sekundach (time()) pobierając próbki przed i po ewolucji. Tutaj mamy podaną średnią z pięciu testów dla każdego parametru.

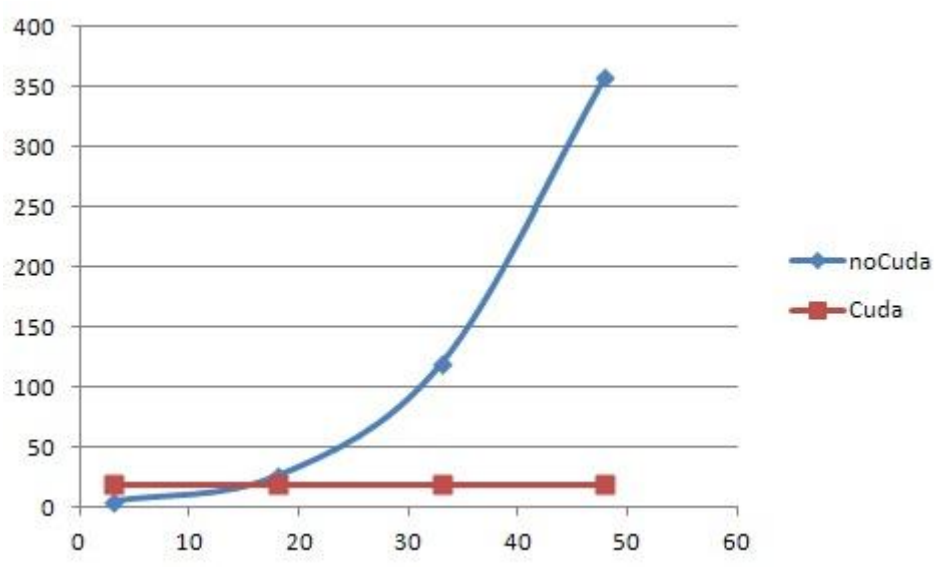
4.1 Medoidy



rys 28 - [Iris] Czas [s] vs Medoidy



rys 29 - [Wine] Czas [s] vs Medoidy

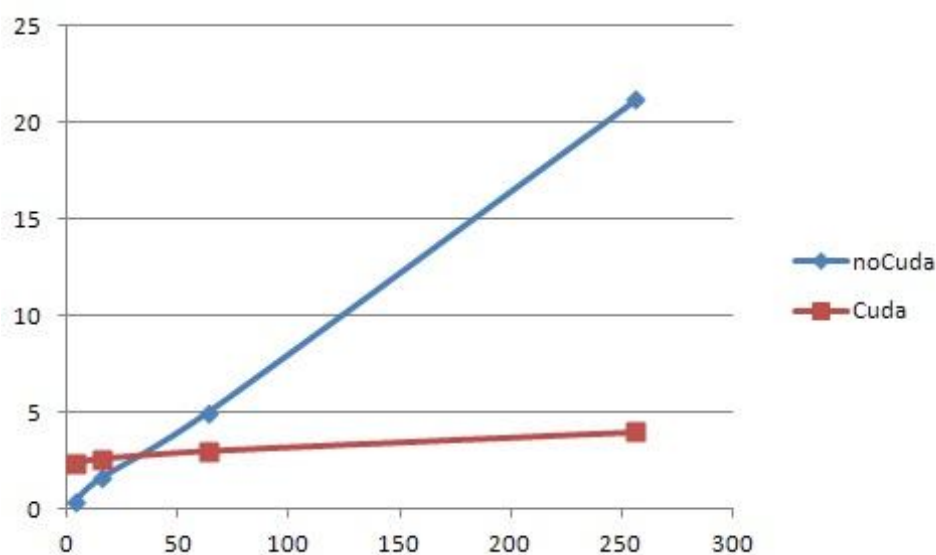


rys 30 - [Cancer] Czas [s] vs Medoidy

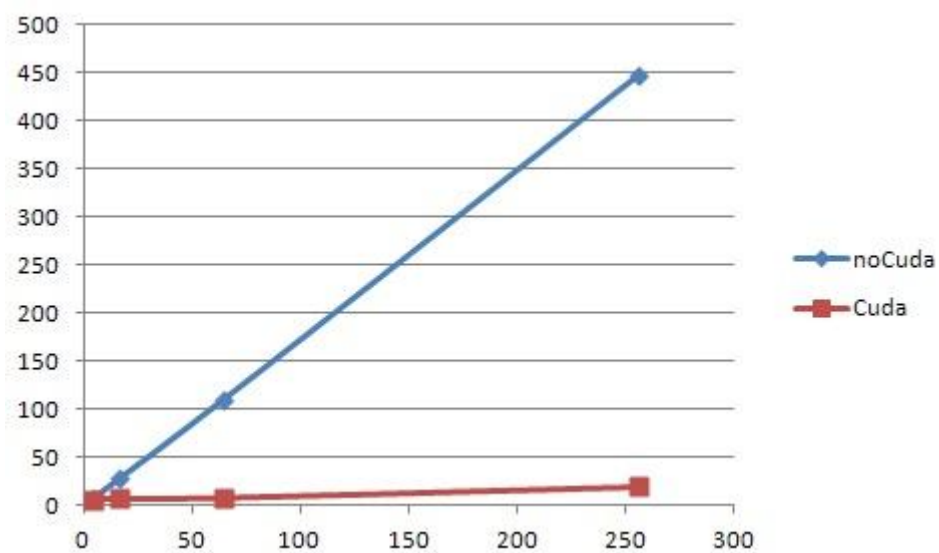
Jak widać, w przypadku algorytmu wykonywanego na procesorze, czas wykonania, wraz z wzrostem ilości medoidów, rośnie niemalże wykładniczo. Gdy natomiast, w przypadku programu wykonującego się na karcie graficznej, czas ten się nie zmienia.

Tutaj jest właśnie widoczna główna optymalizacja programu wykonywanego w technologii CUDA. Wektor medoidów jest dzielony na bloki po 16 wątków każdy i każdy wątek przelicza pojedynczy medoid.

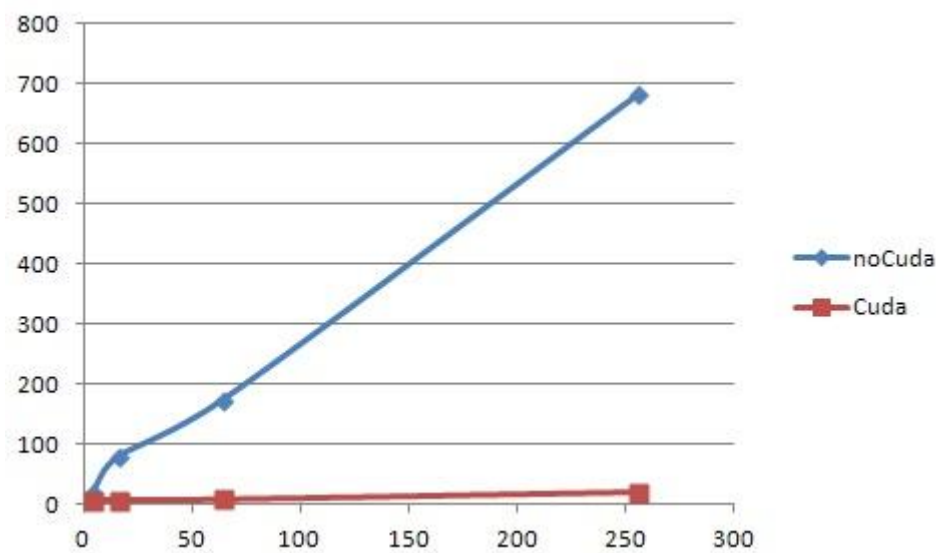
4.2 Populacja:



rys 31 - [Iris] Czas[s] vs Populacja



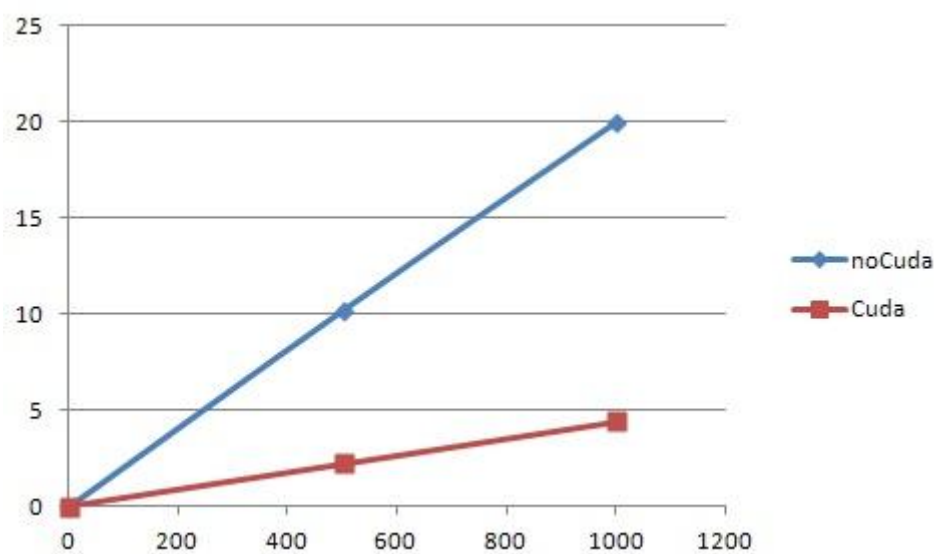
rys 32 - [Wine] Czas[s] vs Populacja



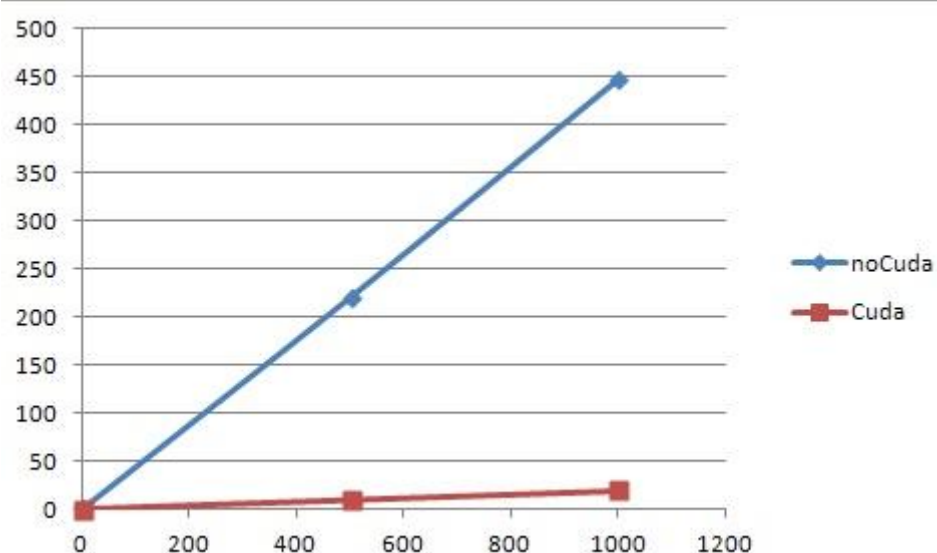
rys 33 - [Cancer] Czas[s] vs Populacja

W przypadku populacji, widać wzrost bardziej liniowy. Oczywiście w przypadku CUDA wzrost ten także istnieje, ale jest minimalny.

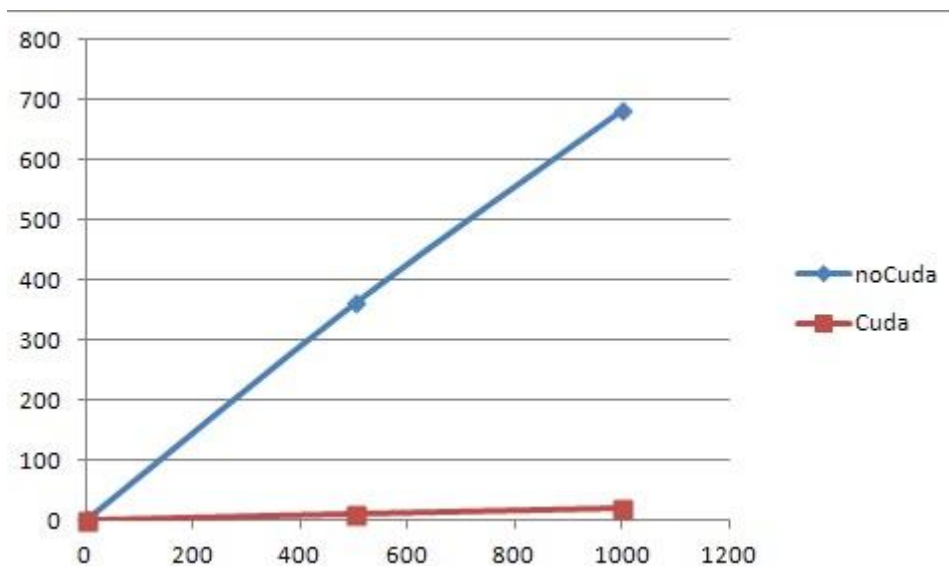
4.3 Kroki



rys 34 - [Iris] Czas[s] vs Kroki



rys 35 - [Wine] Czas[s] vs Kroki



rys 36 - [Cancer] Czas[s] vs Kroki

Czas względem kroki ewolucji jest w obu przypadkach liniowy. Widoczny tu jest jednak wpływ poprzednich optymalizacji i w porównaniu do kodu wykonywanego na procesorze, kod wykonywany przy użyciu CUDA ma o wiele mniejszy wzrost.