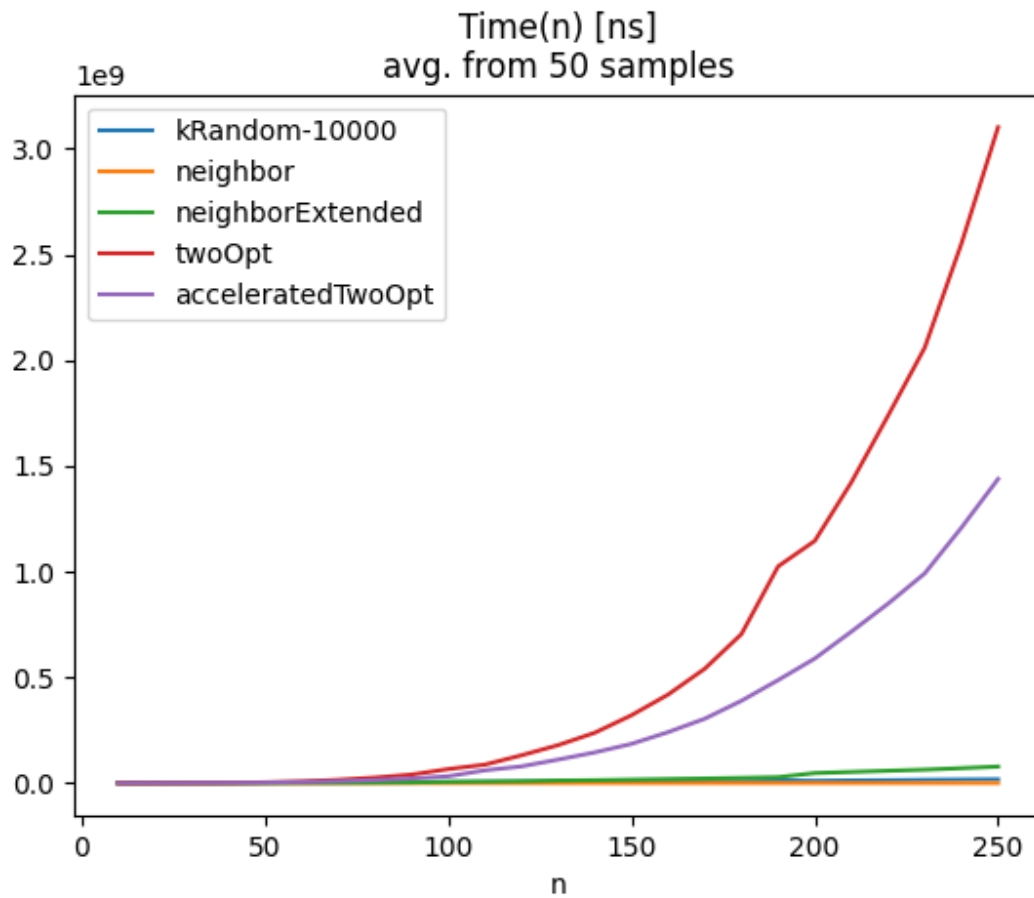


Algorytmy Metaheurystyczne

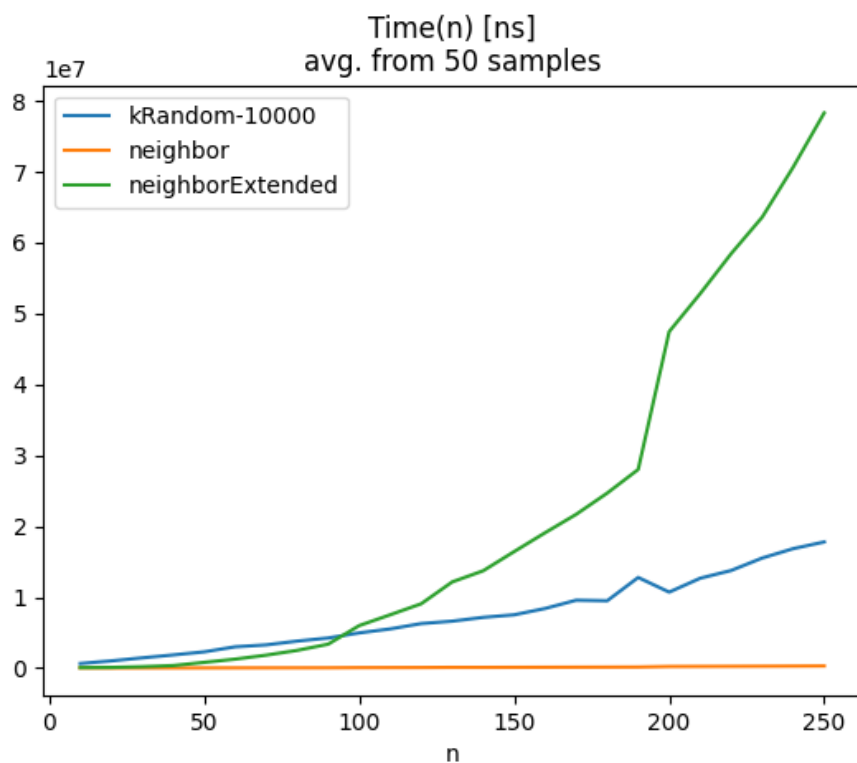
Lista 1 - Sprawozdanie

I - Czas działania / złożoność obliczeniowa

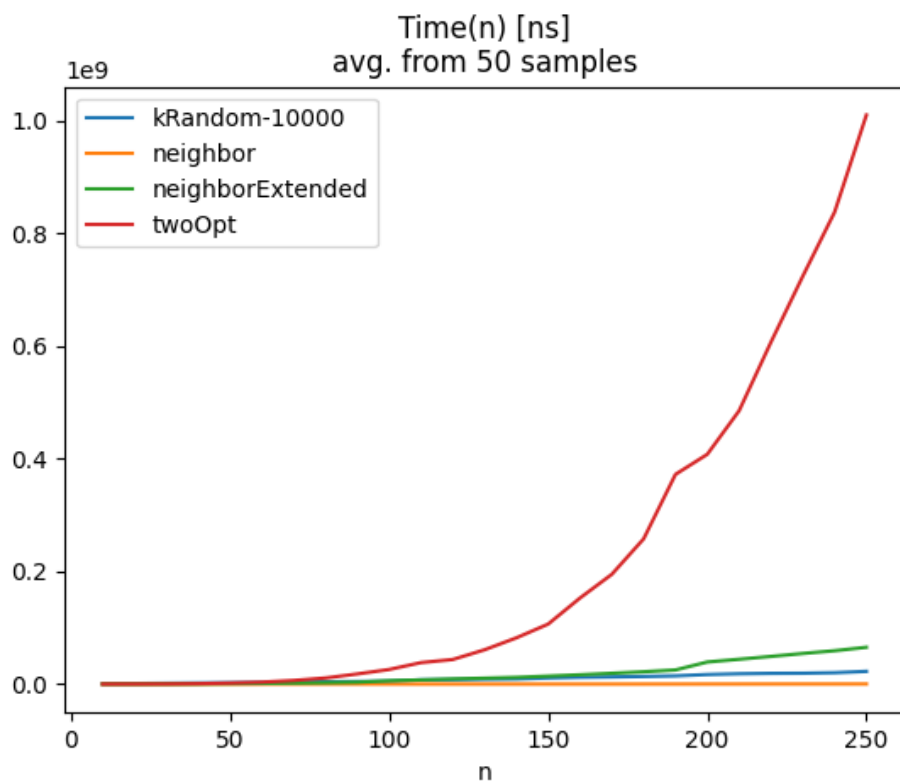
Dla danych generowanych losowo (symetryczne):



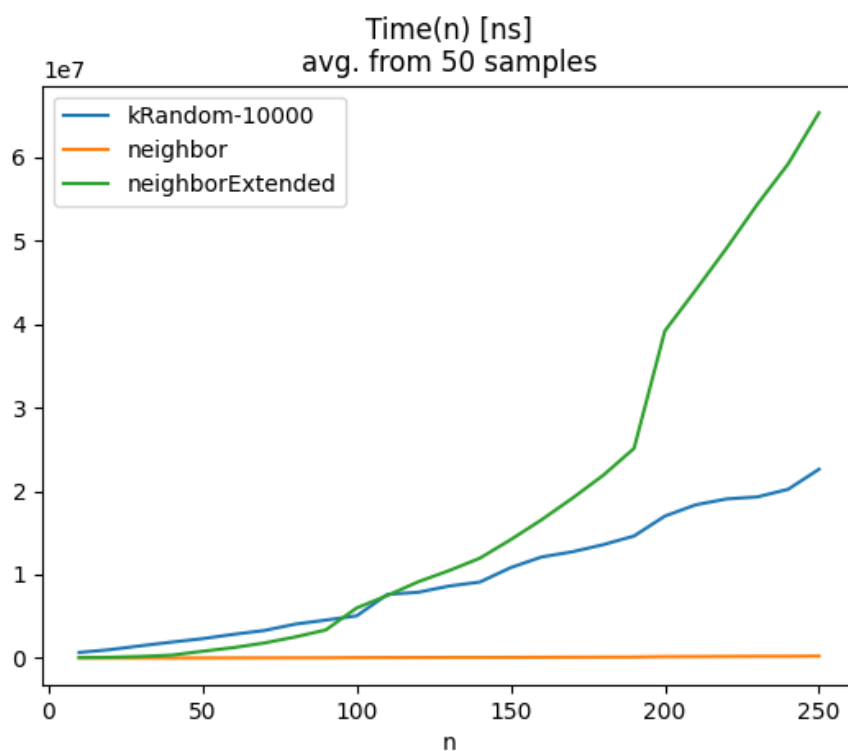
Bez TwoOpt'ów:



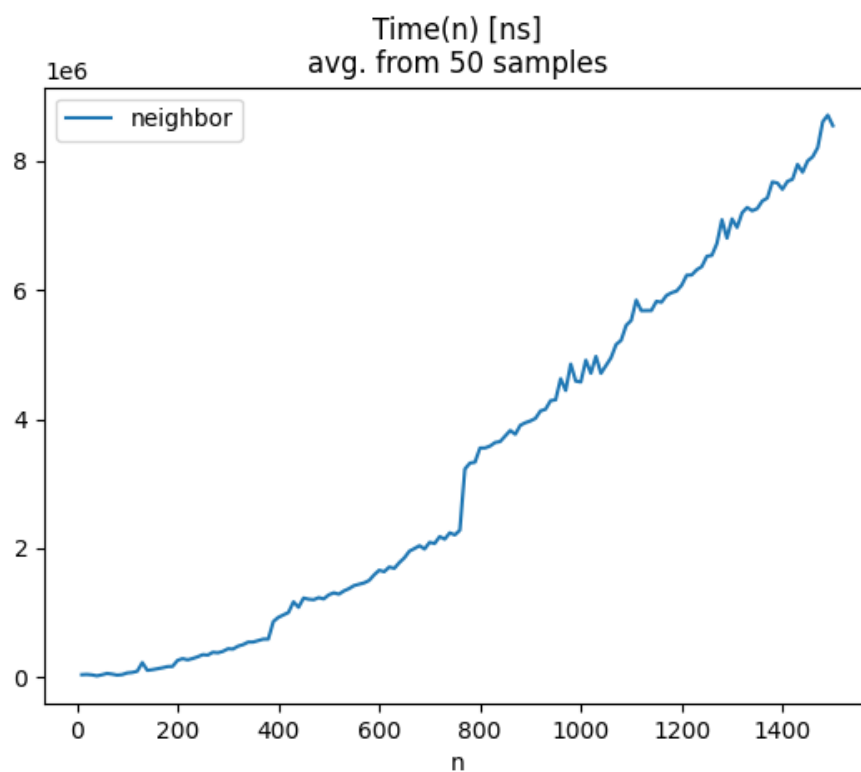
Dla danych generowanych losowo (asymetryczne):



Bez TwoOpt'a:



Dodatkowe testy algorytmu Nearest Neighbor dla losowych danych (asymetrycznych):



Wnioski:

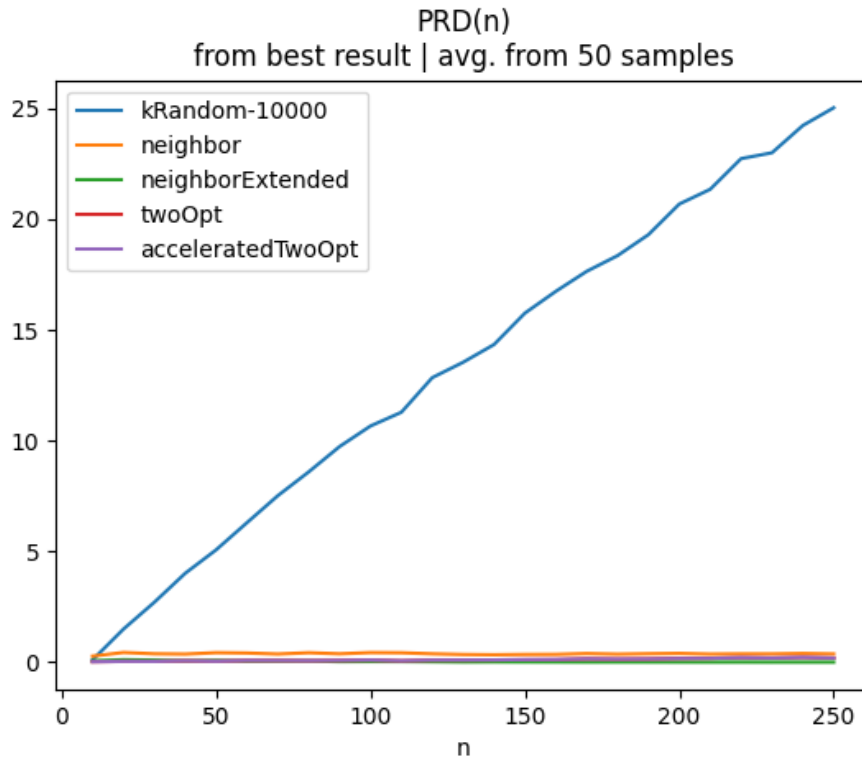
- **Java nie jest najlepszym wyborem do testowania czasu działania algorytmu ze względu na: Garbage Collector, Just In Time Compiler, dynamiczne rozszerzanie pamięci maszyny wirtualnej.**
- **K-Random:** teoretyczna złożoność obliczeniowa wynosi $O(k * n)$ - gdzie k jest liczbą losowań, a koszt generowania losowej permutacji oraz obliczania jej wartości jest liniowy zależny od n . Pokrywa się to z danymi z wykresu.
- **Nearest Neighbour:** teoretyczna złożoność obliczeniowa wynosi $O(n^2)$ - dla każdej z pośród n wartości z instancji szukamy najbliższego sąsiada z pośród pozostałych nie będących jeszcze w permutacji (na początku będziemy mieli $n-1$ takich wartości, a później co wywołanie pętli o jeden mniej). Teoretyczna złożoność w miarę koreluje z danymi z wykresu.
- **Extended Nearest Neighbour:** teoretyczna złożoność obliczeniowa wynosi $O(n^3)$ – wywołujemy algorytm Nearest Neighbour dla każdej z n wartości jako wartości początkowej więc mamy złożoność $n * O(n^2) = O(n^3)$. Pokrywa się to z danymi z wykresu.
- **2 Opt:** Złożoność algorytmu zależy od danych wejściowych. Każda iteracja algorytmu ma złożoność $O(n^3)$, natomiast iteracje są wykonywane do momentu brak poprawy długości drogi. Z wykresów widać, że jest to algorytm dużo wolniejszy od pozostałych.
- **Akcelerowany 2 Opt:** Złożoność algorytmu ponownie zależy od danych wejściowych. Natomiast każda iteracja ma złożoność $O(n^2)$ - ze względu na poprawę sposobu liczenia długości drogi. Iteracje są wykonywane do momentu braku poprawy długości drogi. Z wykresów widać, że jest zauważalnie szybszy od zwykłego 2 Opt'a.
- Zmiana typu problemu - tzn. to czy był on symetryczny, czy nie (nadal losowy) nie miało znaczącego wpływu na asymptotyczny czas działania algorytmów **K-Random, Nearest Neighbour, Extended Nearest Neighbour**.
- Natomiast algorytm 2 Opt reagował na to czy problem był symetryczny, czy też nie - w przypadku problemów symetrycznych wykonywał się zauważalnie (3-krotnie) dłużej.

II - Odchylenie od najlepszego wyniku

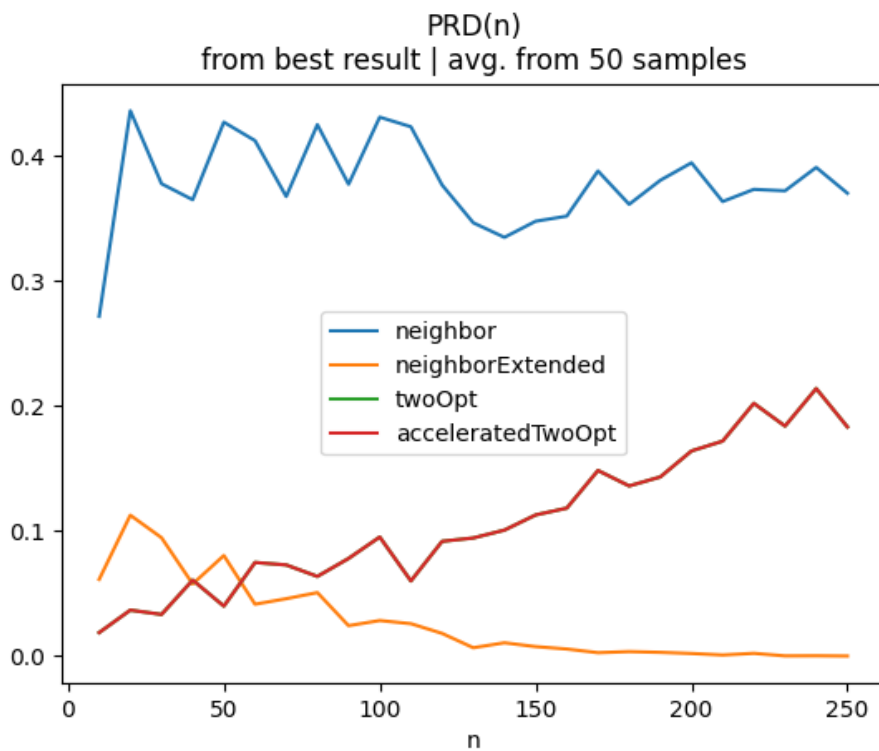
Za najlepszy wynik jest wybierana trasa o najmniejszej długości wyliczona przez algorytmy.

Uwaga: wyniki 2 Opt'a i 2 Opt'a Akcelerowanego dla problemów symetrycznych pokrywają się.

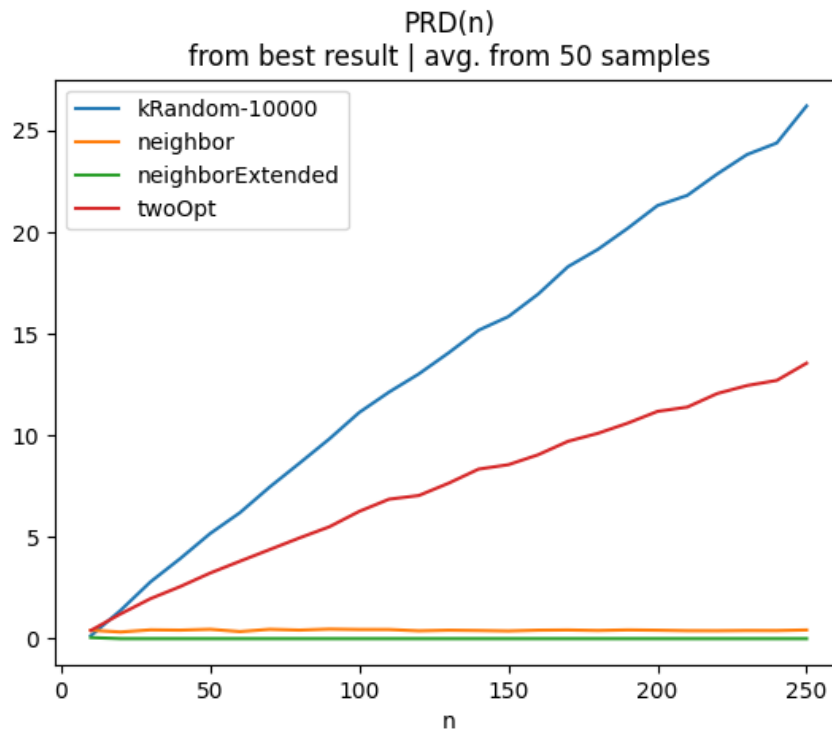
Dla danych generowanych losowo (symetryczne):



Bez K-Random:



Dla danych generowanych losowo (asymetryczne):

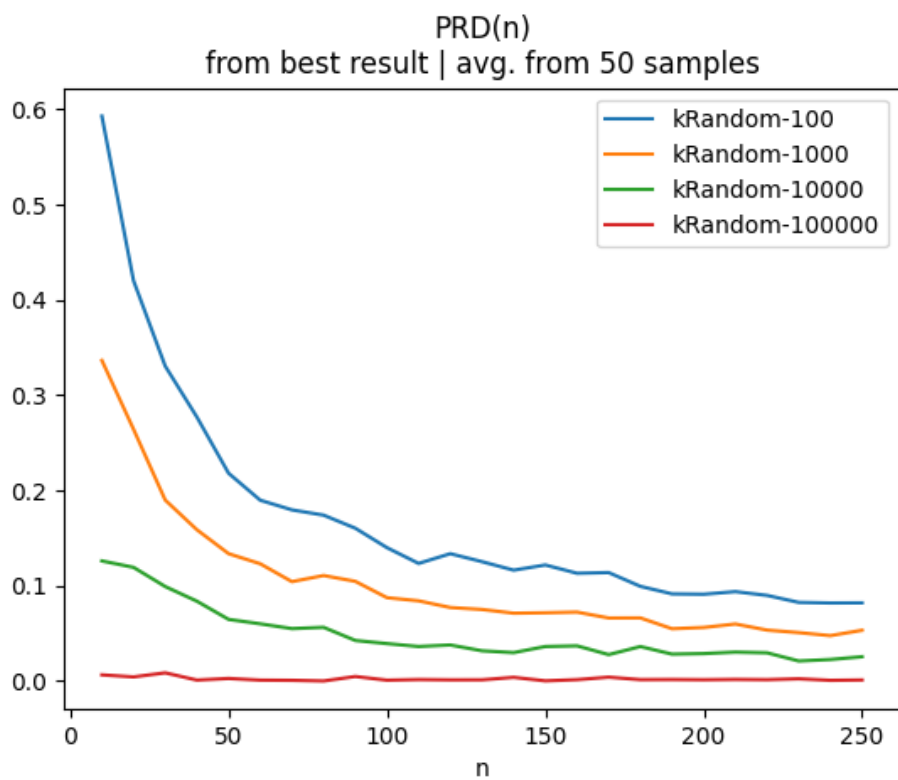
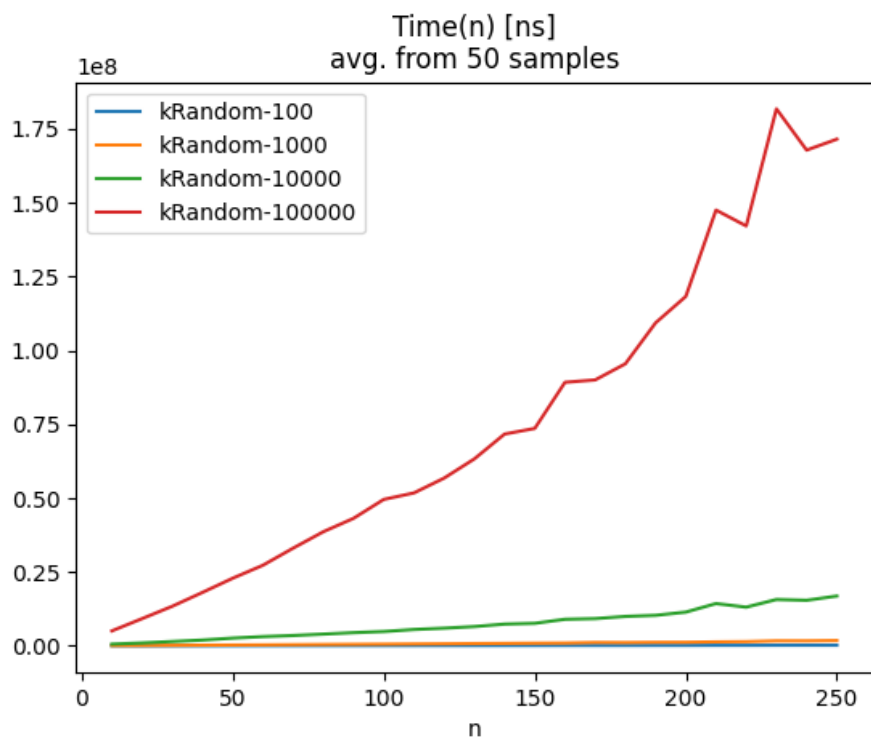


Wnioski:

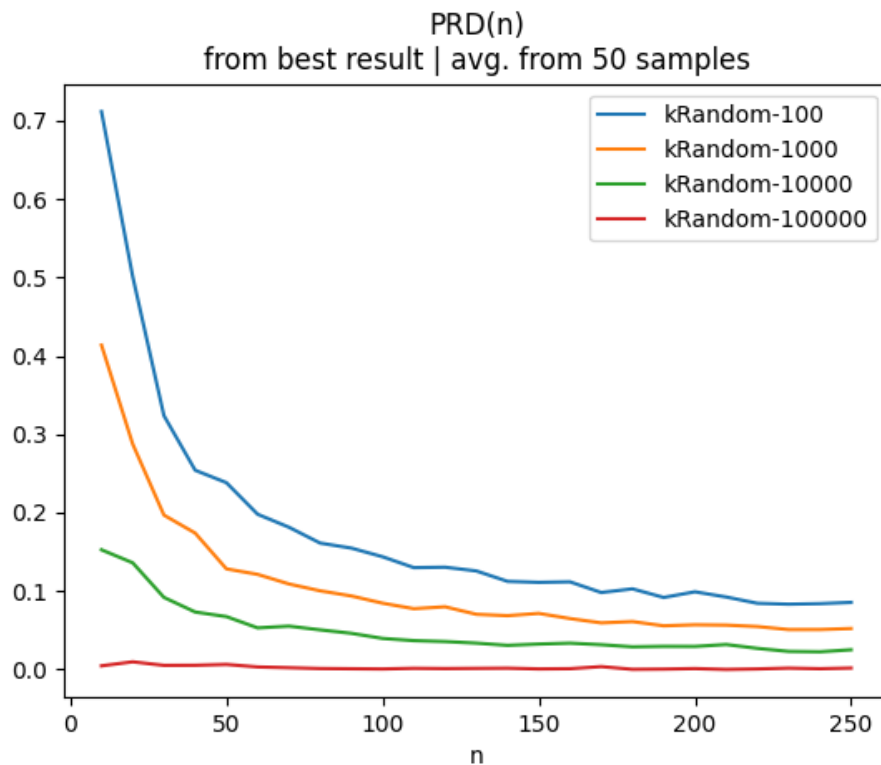
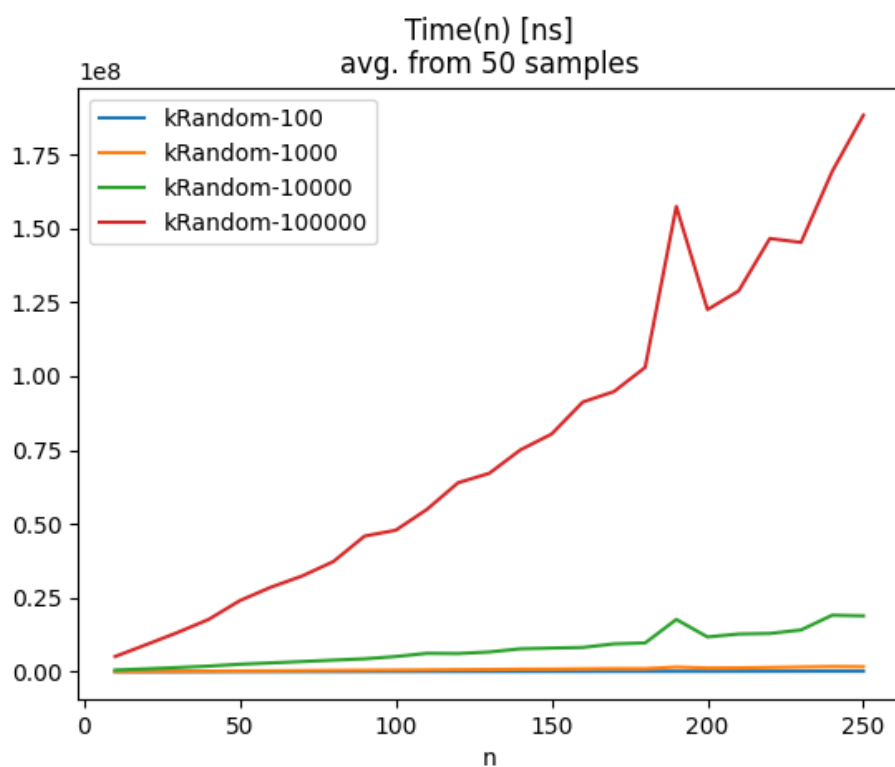
- **K-Random:** dla dużych problemów dawał wartości daleko drogi dużo większe od pozostałych algorytmów. W próbie, dla której stworzony został powyższy wykres zostało zastosowane $k = 10000$
- **Nearest Neighbour:** dla losowych problemów wyniki są całkiem zadowalające (w porównaniu do pozostałych algorytmów), jednakże łatwo pokazać problem, gdzie ta metoda daje wyniki bardzo dalekie od optymalnego. Dodatkowym problemem są różne wyniki algorytmu w zależności od punktu rozpoczęcia.
- **Extended Nearest Neighbour:** wywołuje algorytm Nearest Neighbour dla każdego możliwego punktu początkowego tym samym eliminując problem zależności od punktu początkowego. Wyniki tego algorytmu są średnio najlepsze (porównując do pozostałych algorytmów) dla większych problemów (wielkości > 60).
- **2 Opt:** radzi sobie zdecydowanie lepiej w przypadku problemów symetrycznych niż asymetrycznych - może to być powodem dlaczego czas działania tego algorytmu w przypadku problemów symetrycznych jest większy.

III - Porównanie różnych wartości k dla algorytmu k-Random

Dla danych generowanych losowo (symetryczne):



Dla danych generowanych losowo (asymetryczne):



Wnioski:

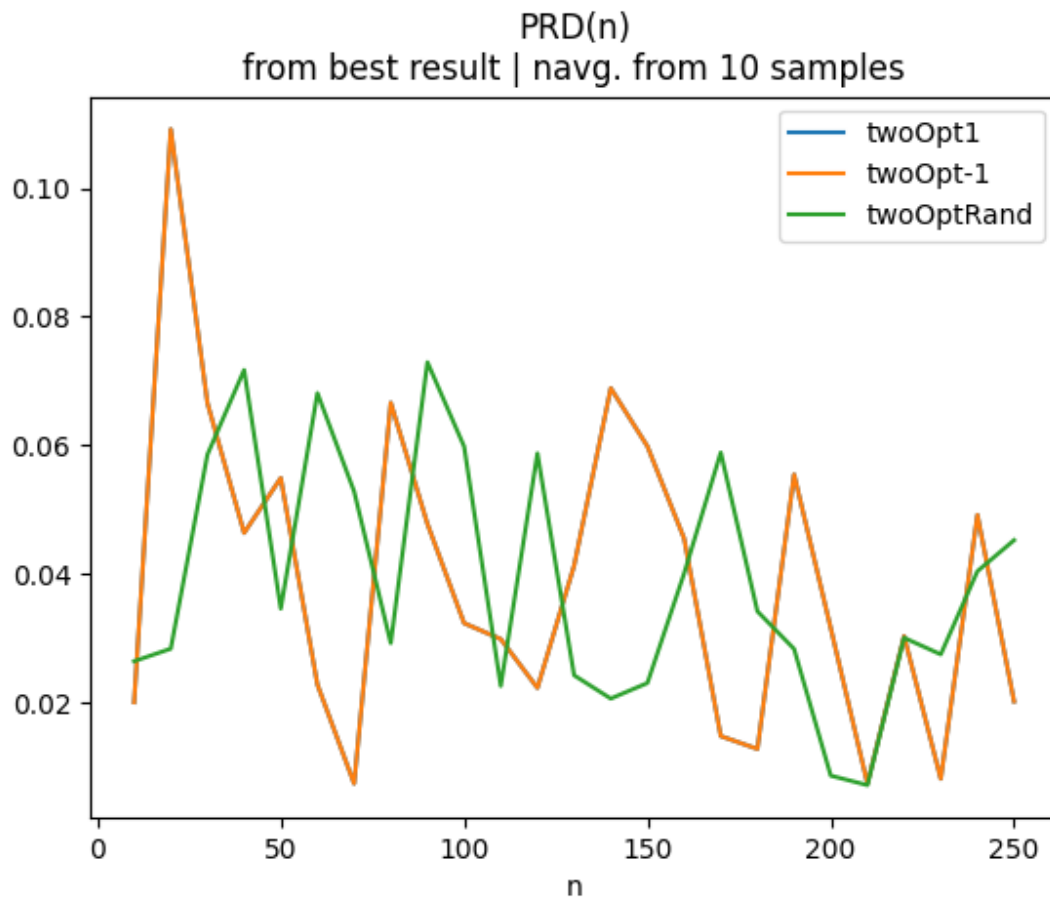
- Zwiększanie k co prawda daje średnio lepsze rezultaty, natomiast znacząco wydłuża czas działania algorytmu.
- Warto zauważyć że dla problemów $n \geq 200$ i algorytmów kRandom-10000 oraz kRandom-100000 ten pierwszy był średnio gorszy o około 5%, natomiast jego czas wykonania był ponad wielokrotnie szybszy.

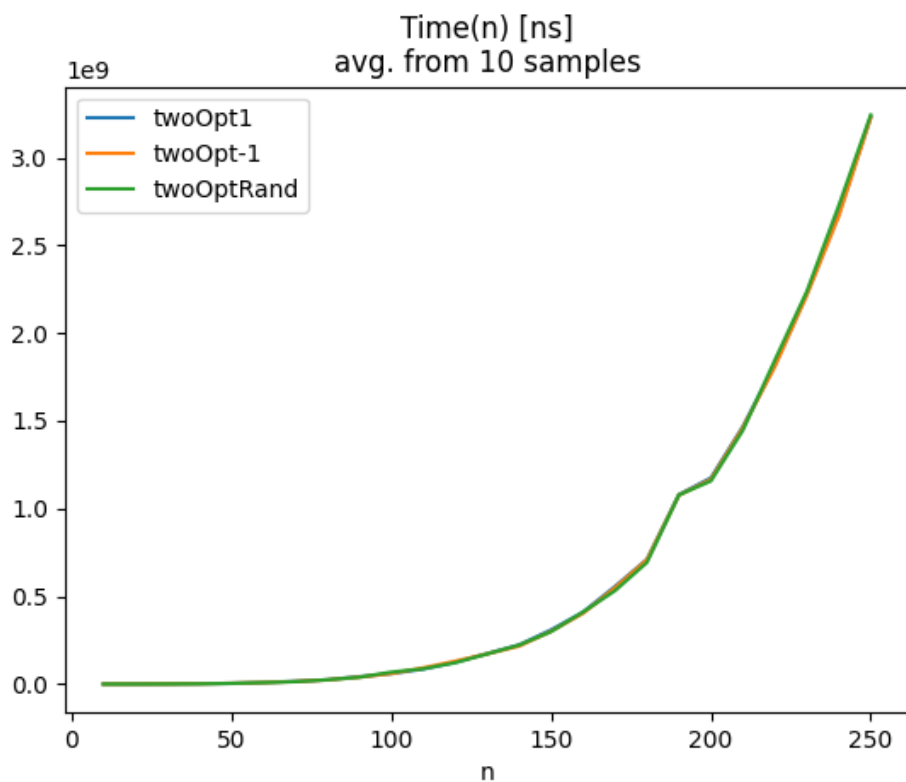
IV - Porównanie wyboru losowej permutacji, a permutacji (1 2 3 ... n) dla algorytmu 2 Opt

Oznaczenia:

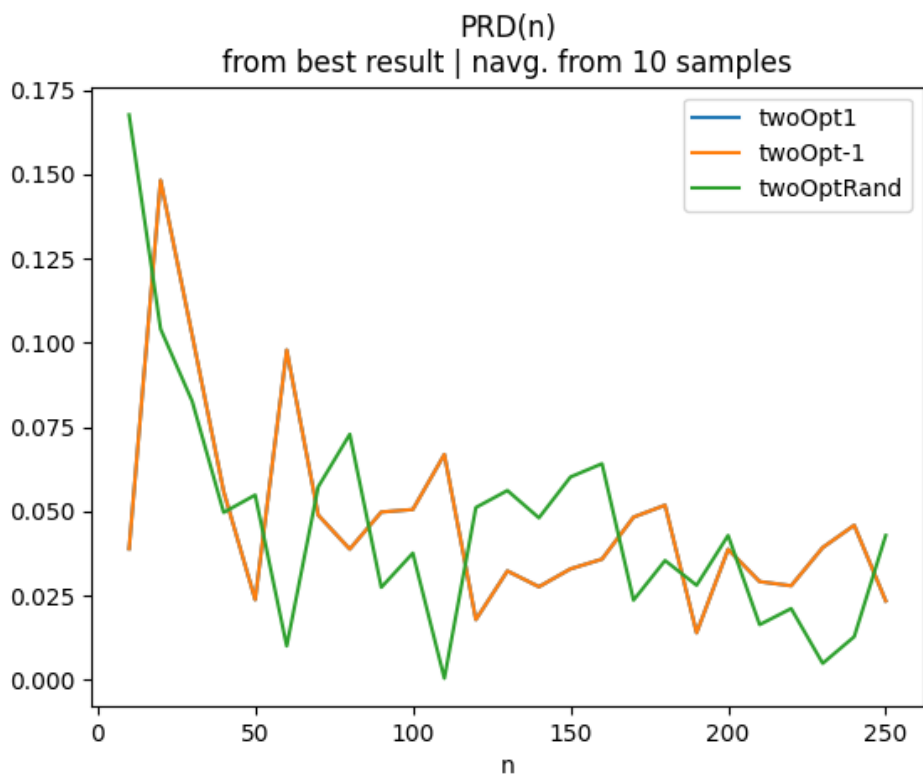
- twoOpt1 - algorytm 2 Opt o rozwiązaniu startowym (1 2 3 ... n)
- twoOpt-1 - algorytm 2 Opt o rozwiązaniu startowym (n $n-1$ $n-2$... 1)
- twoOptRand - algorytm 2 Opt o losowym rozwiązaniu startowym

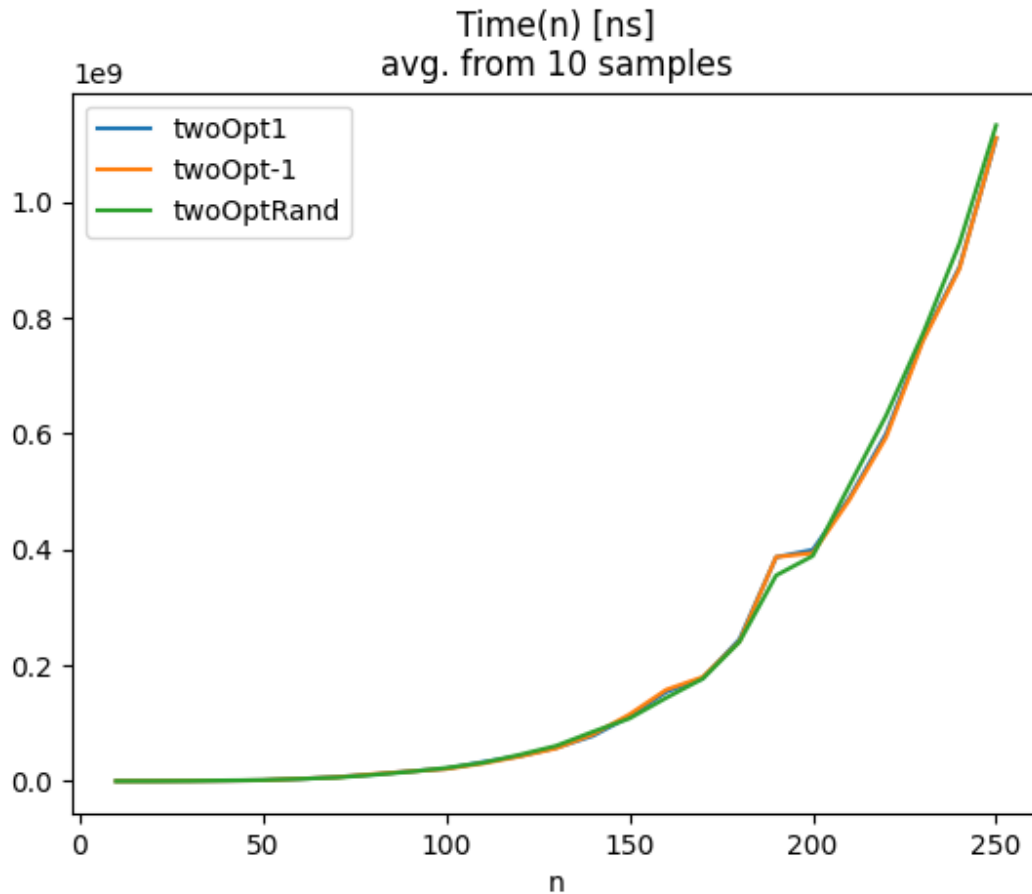
Dla danych generowanych losowo (symetryczne):





Dla danych generowanych losowo (asymetryczne):





Wnioski:

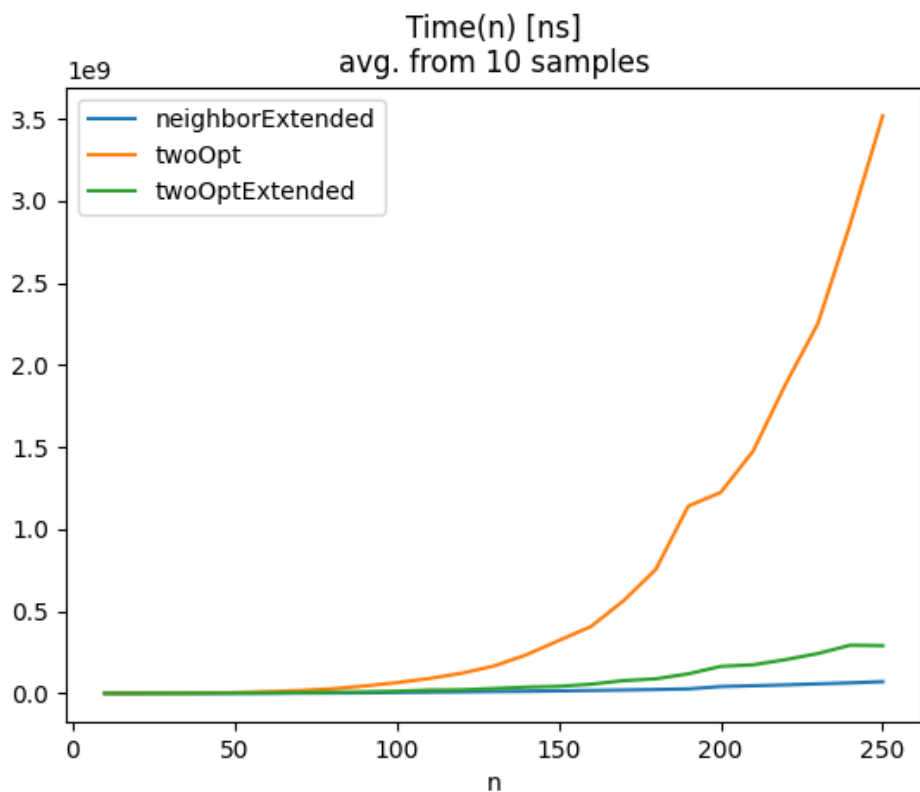
- W średnim przypadku wybór początkowej permutacji dla algorytmu 2 Opt nie ma znaczenia (o ile jest ona losowa - nie zastosowaliśmy innego algorytmu do jej znalezienia). Zatem w implementacji możemy wychodzić od permutacji (1 2 3 ... n).

V - Zastosowanie 2 Opt'a do poprawy wyników algorytmu Neighbor Extended

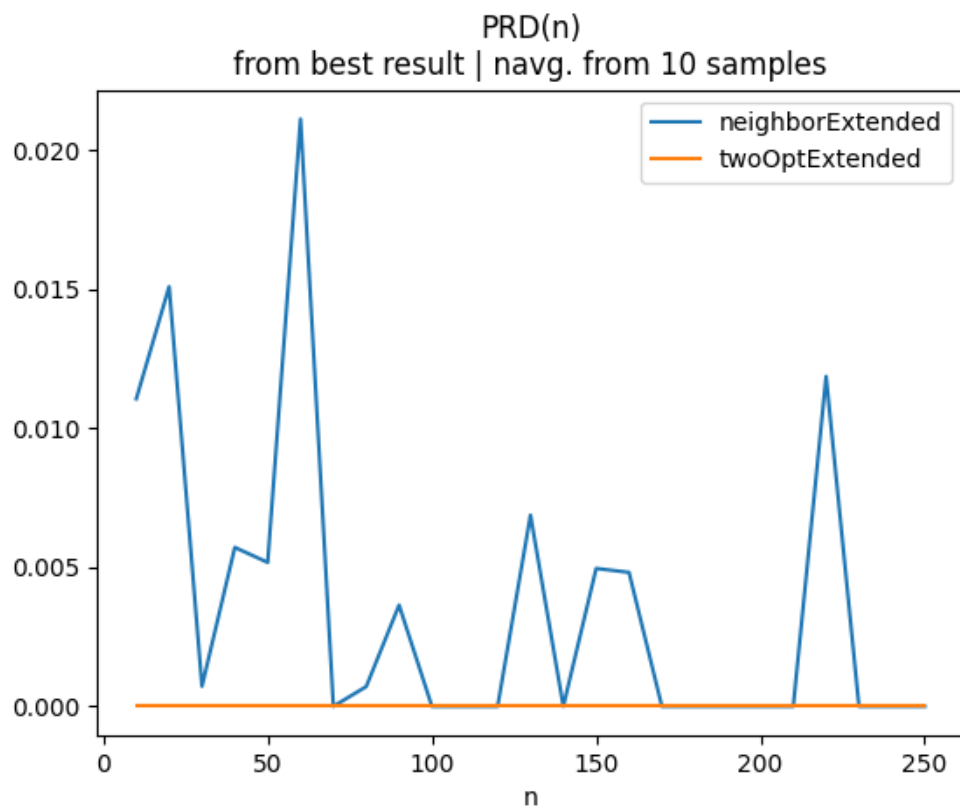
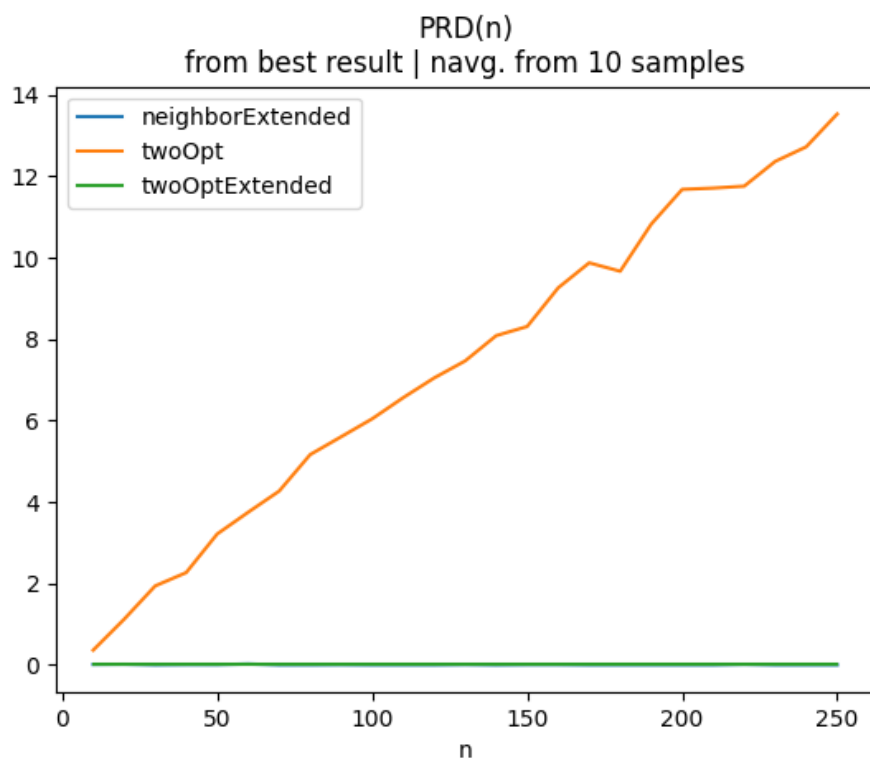
Oznaczenie:

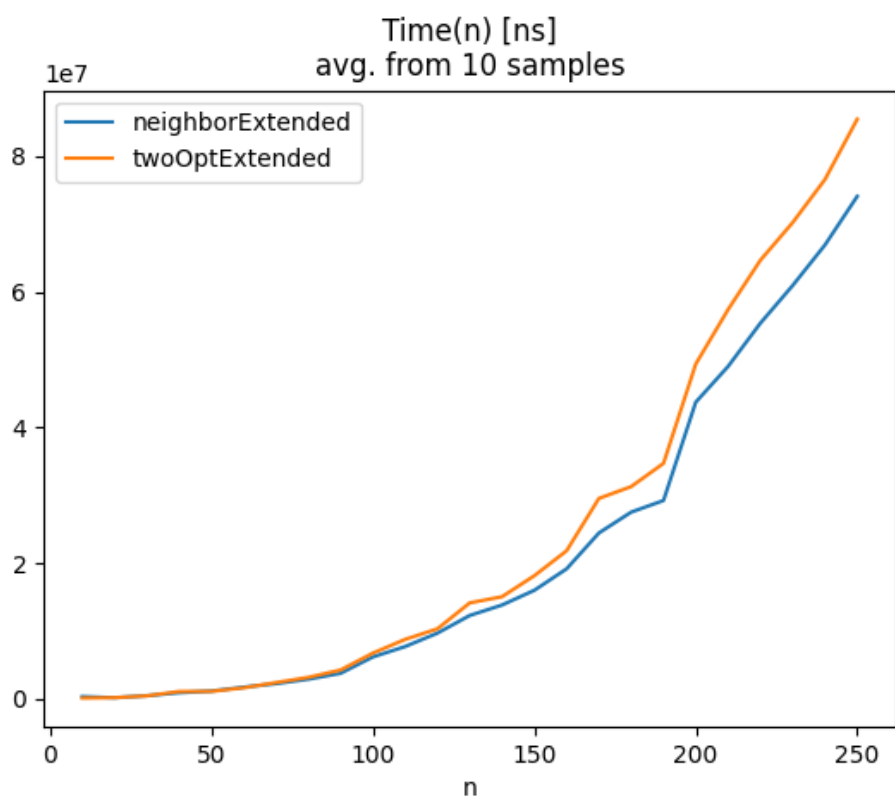
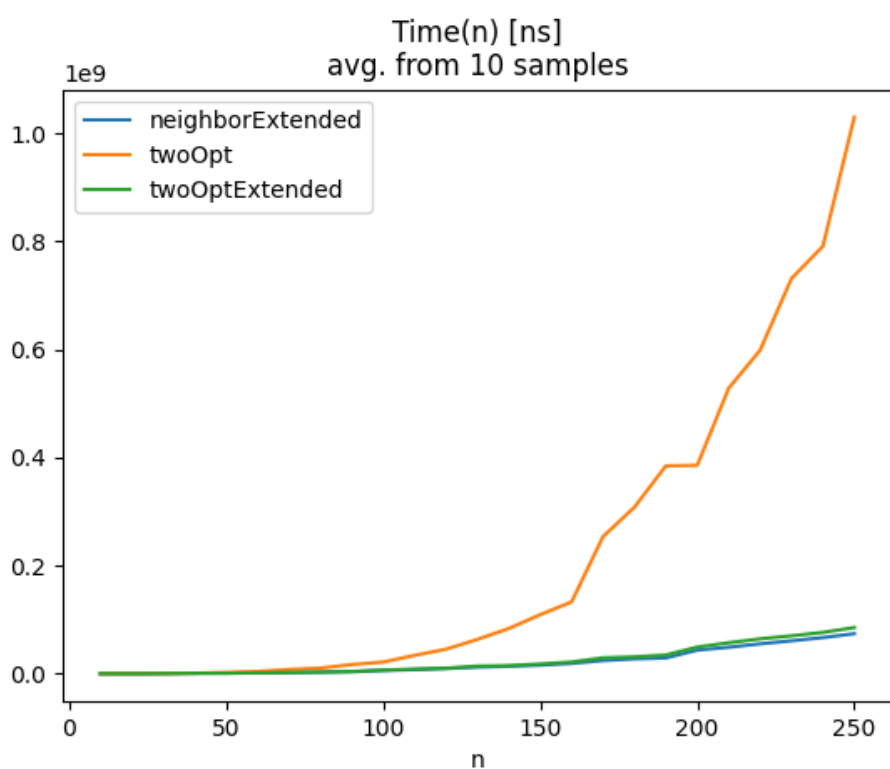
twoOptExtended - Algorytm wykorzystuje Neighbor Extended do wyszukiwania permutacji, a następnie poprawia ją algorytmem 2 Opt (używa wyniku Neighbor Extended jako początkowej permutacji dla 2 Opt'a).

Dla danych generowanych losowo (symetryczne):



Dla danych generowanych losowo (asymetryczne):





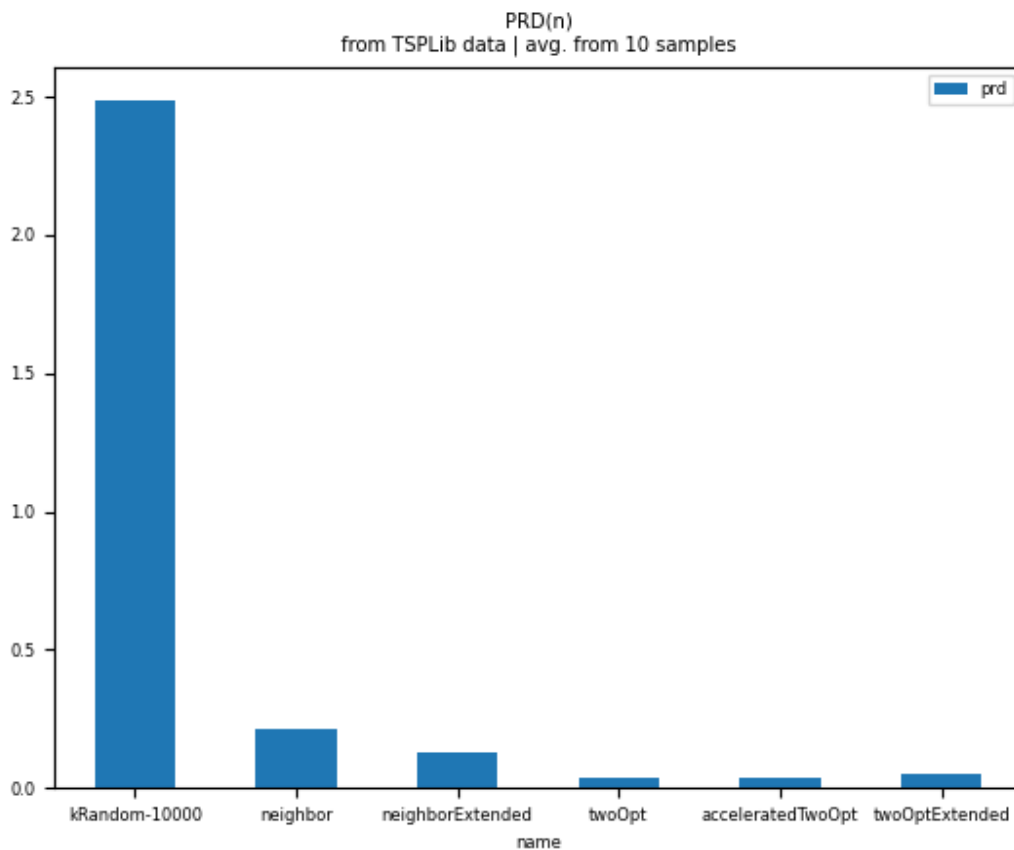
Wnioski:

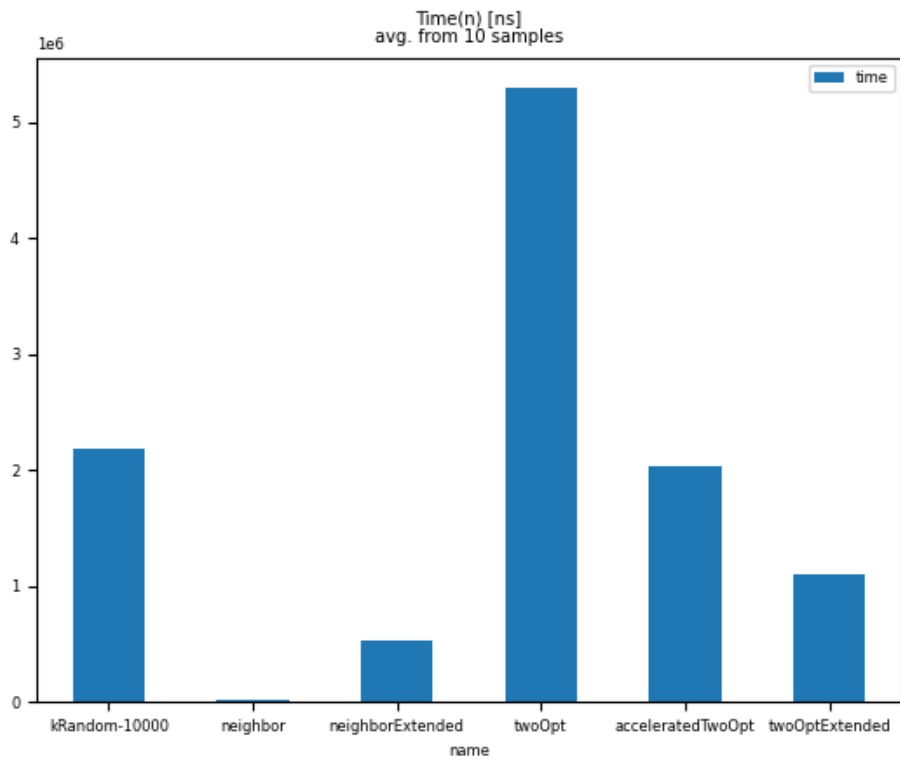
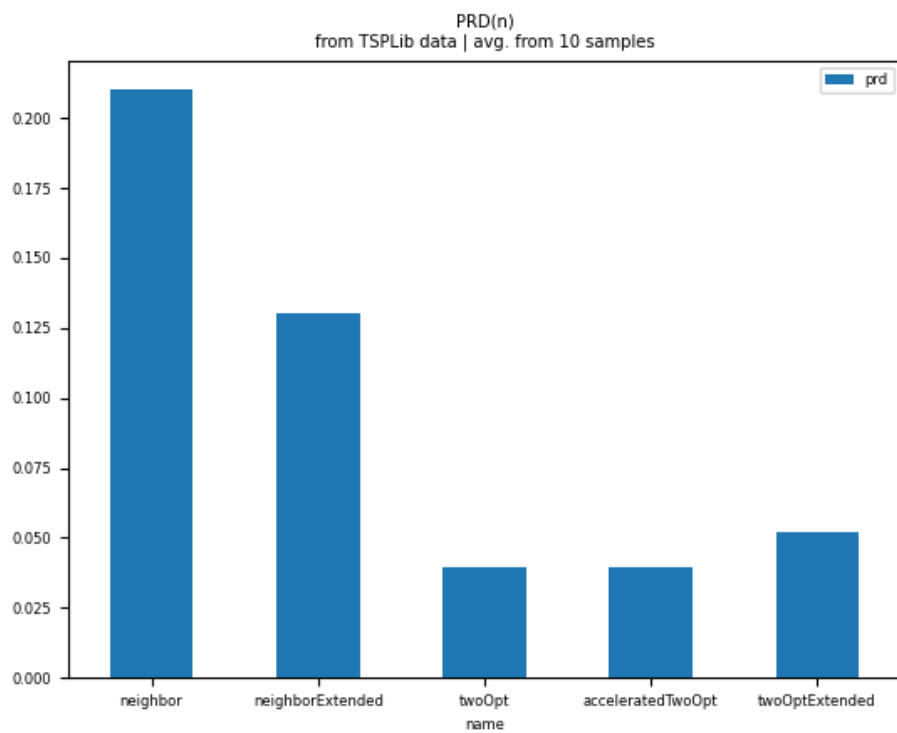
- Algorytm twoOptExtended daje znaczącą poprawę wyników dla przypadków symetrycznych w porównaniu do wyników algorytmów składowych.
- Dla przypadków asymetrycznych bardzo rzadko otrzymamy poprawę wyników otrzymanych z algorytmu Neighbor Extended.
- Algorytm twoOptExtended jest średnio znacząco lepszy od 2 Opt'a ze względu na czas działania.
- W przypadku problemów asymetrycznych jego czas działania i wyniki są zbliżone do Neighbor Extended.

VI - TSPLIB (Symetryczne)

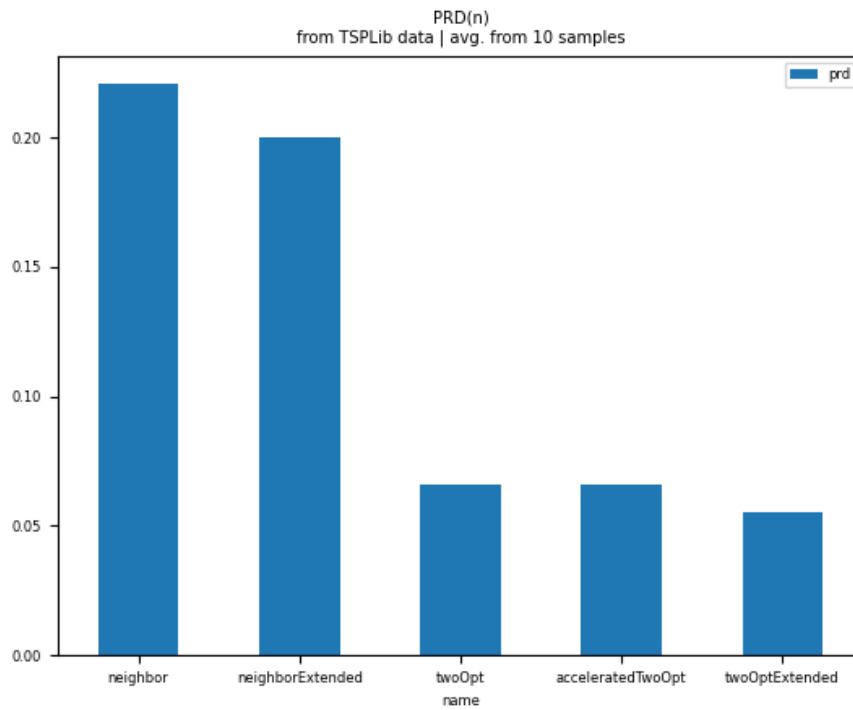
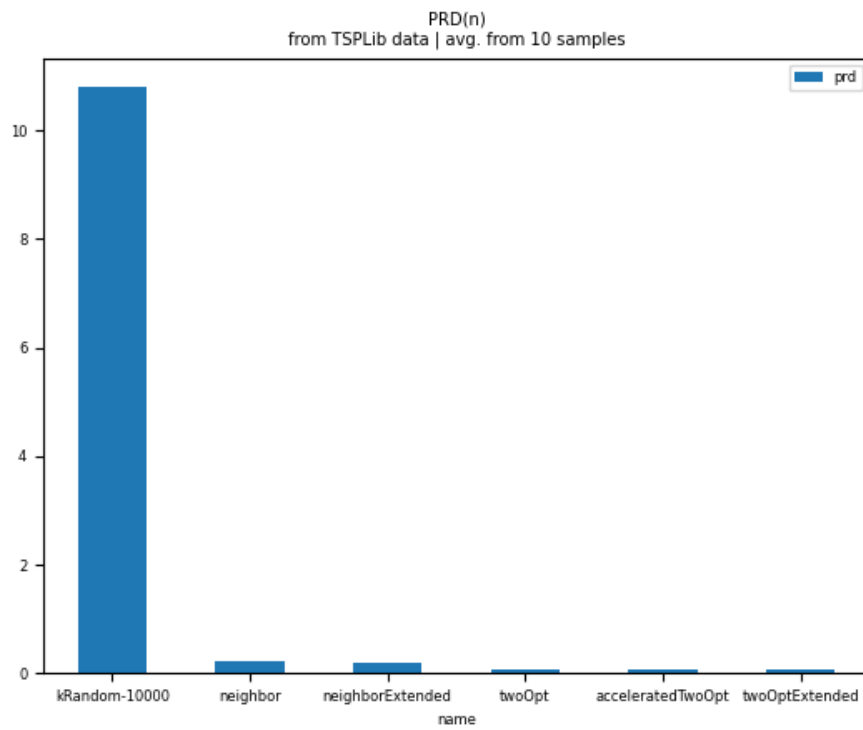
Przeprowadziliśmy testy na 6 plikach z TSPLIB'a, natomiast w sprawozdaniu znajdują się 2 z nich. Pozostałe można zobaczyć w plikach umieszczonych na repozytorium.

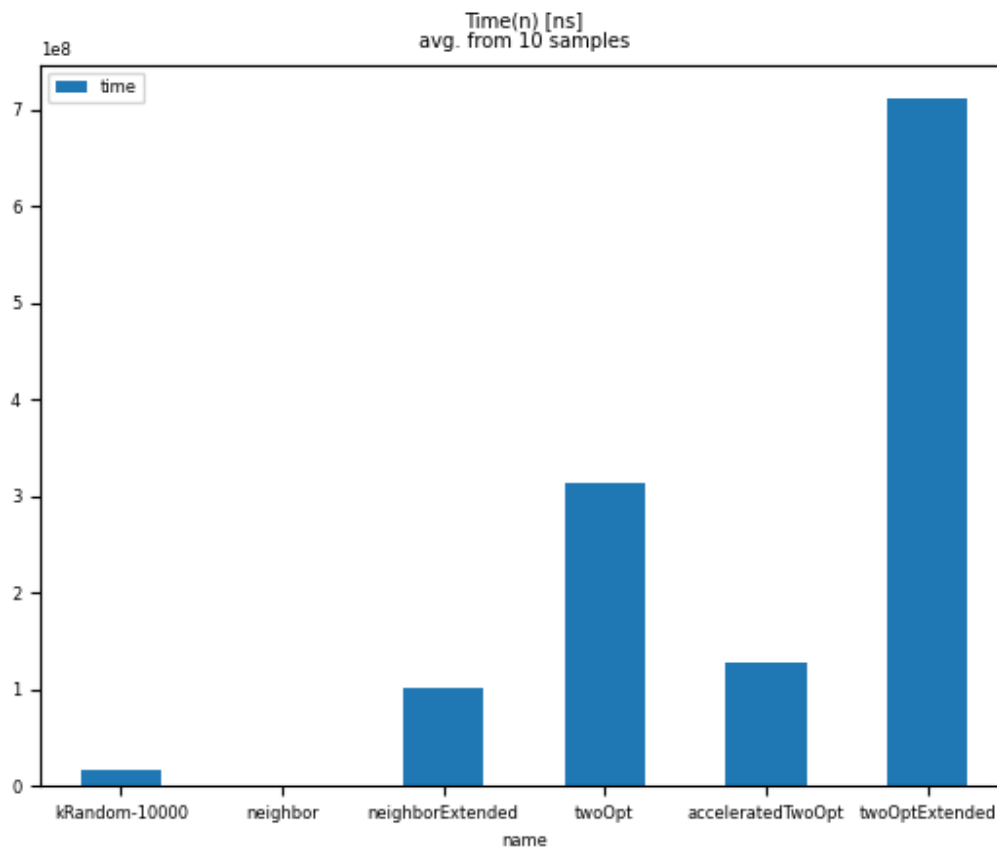
att48:





a280:





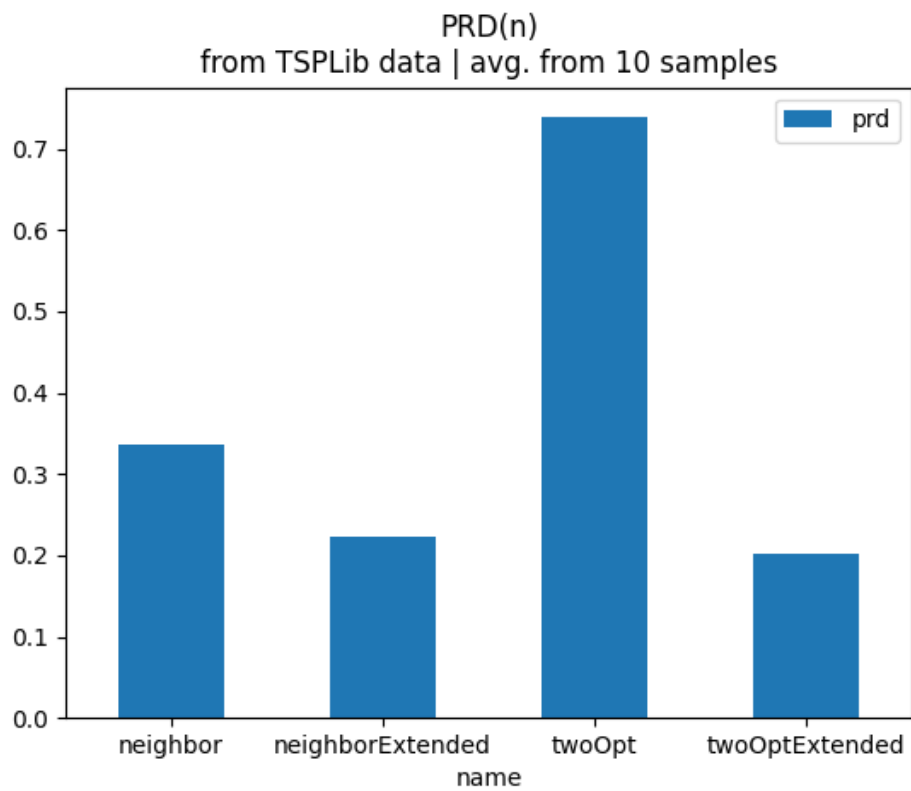
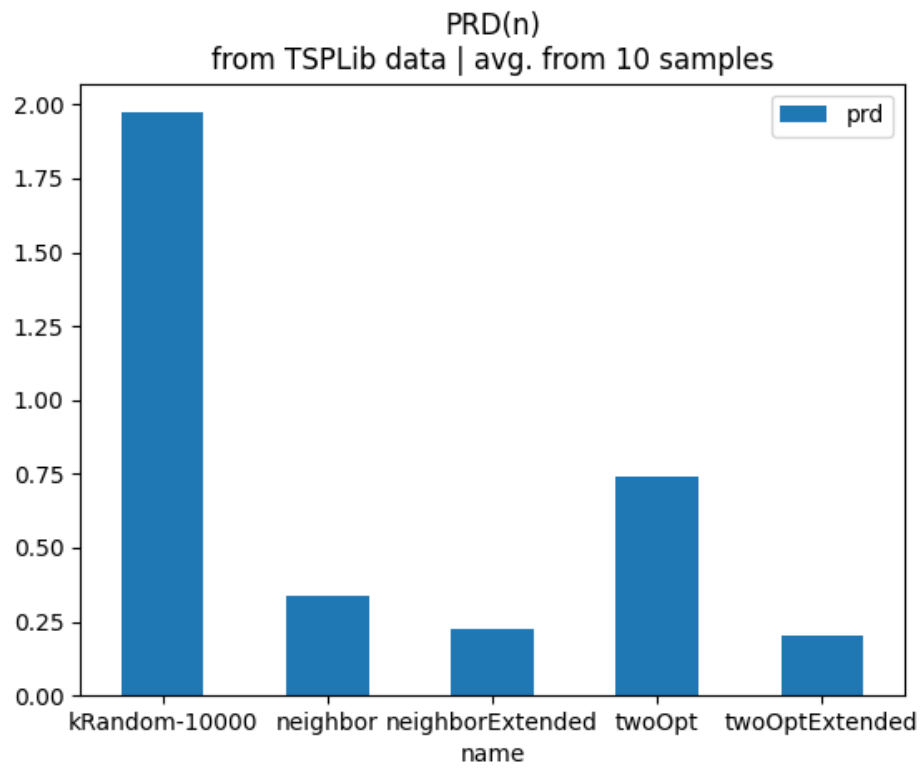
Wnioski:

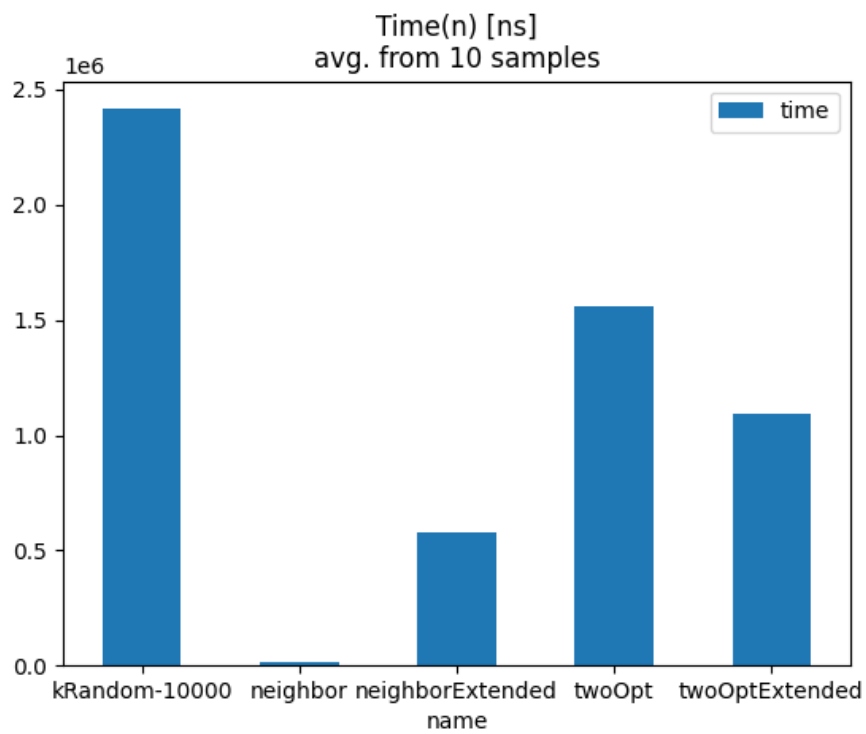
- Jak widać z pierwszego att48, algorytm twoOptExtended nie zawsze daje lepsze wyniki od zwykłego 2 Opt'a (czasami permutacja początkowa (1 2 ... n) może okazać się lepsza)
- Dla problemów symetrycznych TSPLIB algorytm 2 Opt i twoOptExtended dawał średnio najlepsze wyniki

VII - TSPLIB (Asymetryczne)

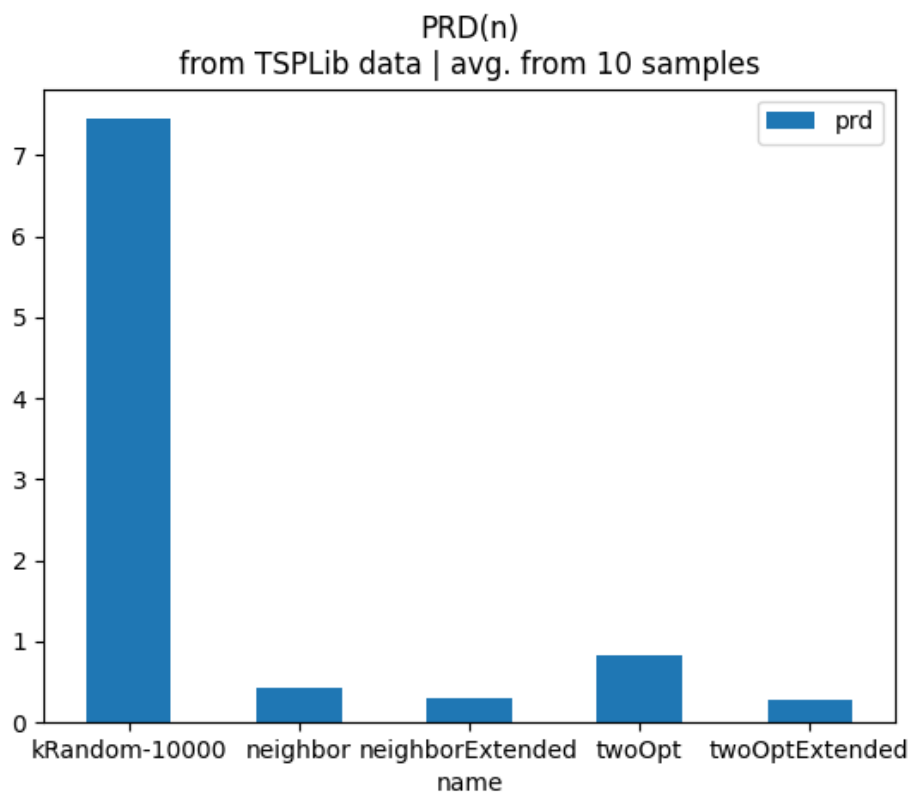
Przeprowadziliśmy testy na 6 plikach z TSPLIB'a, natomiast w sprawozdaniu znajdują się 2 z nich. Pozostałe można zobaczyć w plikach umieszczonych na repozytorium.

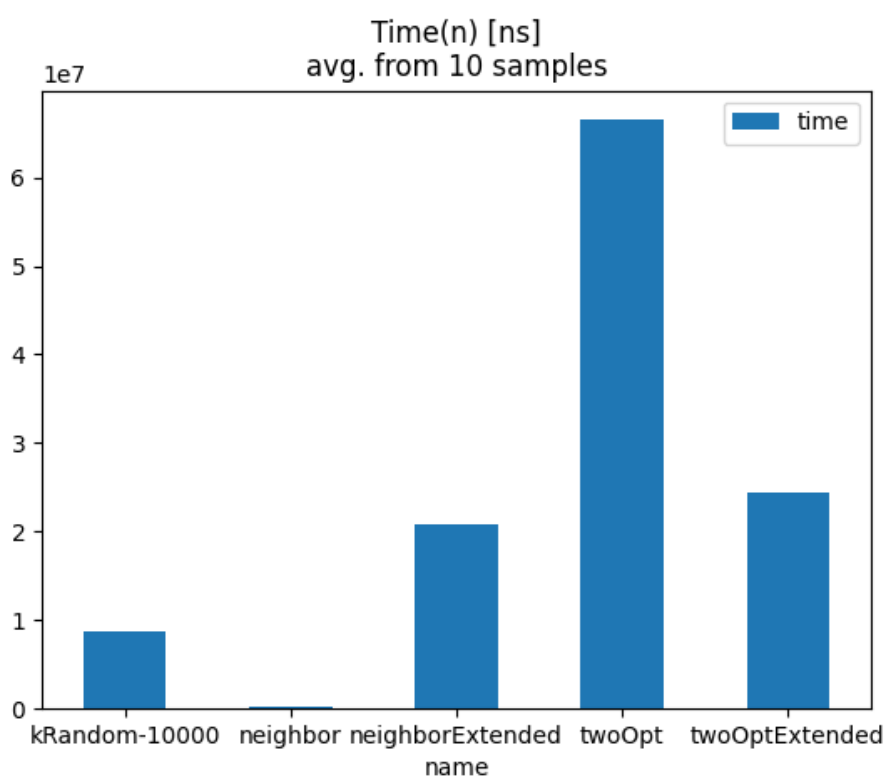
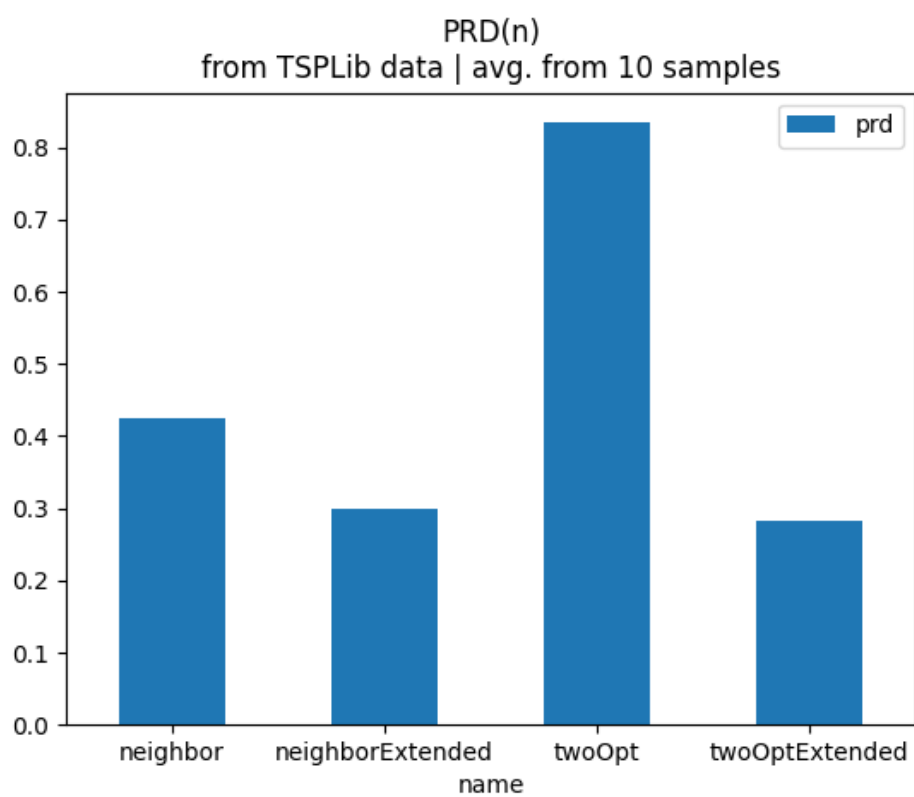
ftv47:





ftv170:





Wnioski:

- W przypadku problemów asymetrycznych z TSPLIB'a wyniki algorytmów Neighbor i Neighbor Extended były lepsze niż algorytmu 2 Opt.
- Algorytm twoOptExtended bardzo rzadko dawał znaczącą poprawę nad algorytmem Neighbor Extended.