

Algorytmy Metaheurystyczne

Lista 1

Adam Chojnacki, Wojciech Leśniak

1. Opis Problemu

Lista 1 zakładała implementację programu rozwiązującego problem komiwojażera za pomocą metaheurystyk:

- a) metoda k-Random,
- b) metoda najbliższego sąsiada,
- c) rozszerzona metoda najbliższego sąsiada,
- d) algorytm Opt2.

Następnie należało przeprowadzić serię eksperymentów dla losowo generowanych instancji problemów i porównać ich wyniki.

2. Rozwiązanie problemu

Zadanie zostało wykonane w języku programowania Java 17. Przetestowano pod systemem Manjaro 21 oraz Windows 10.

Program zawiera komponenty odpowiedzialne za:

- a) wczytanie problemu,
- b) wygenerowanie losowego problemu,
- c) przetrzymanie instancji problemu,
- d) przetrzymanie rozwiązania problemu i obliczenie wartości funkcji celu,
- e) próbę rozwiązania problemu za pomocą zaimplementowanych metaheurystyk,
- f) przeprowadzenie eksperymentów.

Dla algorytmu k-random użyto wartości $k = 100$.

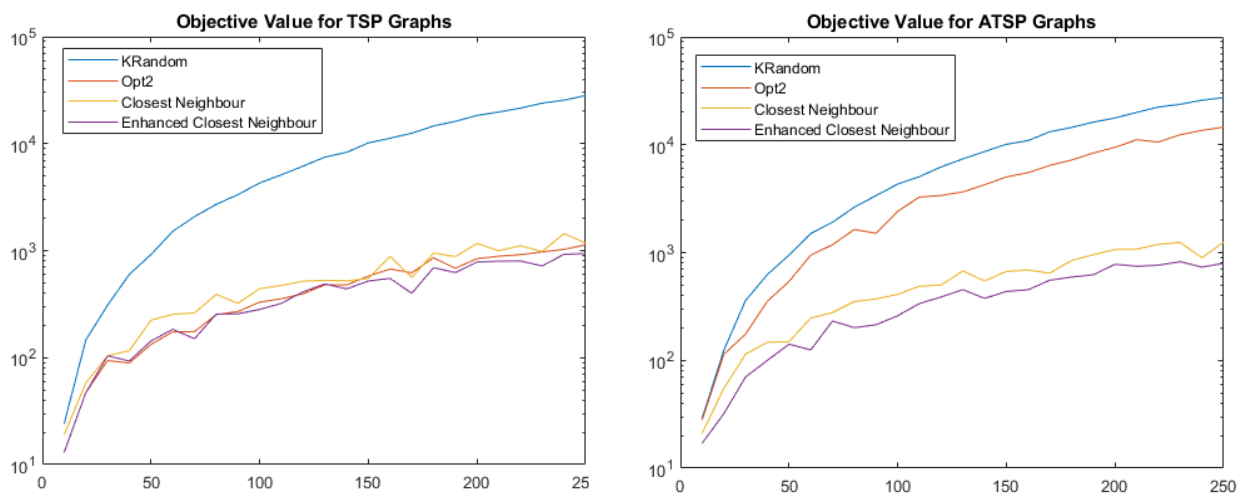
3. Wyniki eksperymentów

Wykonano 4 eksperymenty:

- a) wartości funkcji celu dla rozwiązań danych instancji problemu. zaproponowane przez metaheurystyki. Wykres przedstawia zależność tych wartości dla danych algorytmów od rozmiaru problemu. Badania przeprowadzono dla instancji TSP oraz ATSP, generowanych losowo,
- b) wartości funkcji celu dla rozwiązań danych instancji problemu. zaproponowane przez metaheurystyki znormalizowane do postaci PRD. Wykres przedstawia zależność tych wartości dla danych algorytmów od rozmiaru problemu. Badania przeprowadzono dla instancji TSP oraz ATSP, generowanych losowo,
- c) wartości czasu wykonywania się meteahursytek nad daną instancją problemu. Wykres przedstawia zależność tych wartości dla danych algorytmów od rozmiaru problemu. Badania przeprowadzono dla instancji TSP i ATSP (osobno), generowanych losowo,
- d) wartości funkcji celu dla rozwiązań danej instancji przez metaheurystyki k-random dla różnych wartości parametru k. Wykres przedstawia zależność wartości funkcji celu w zależności od wartości k.

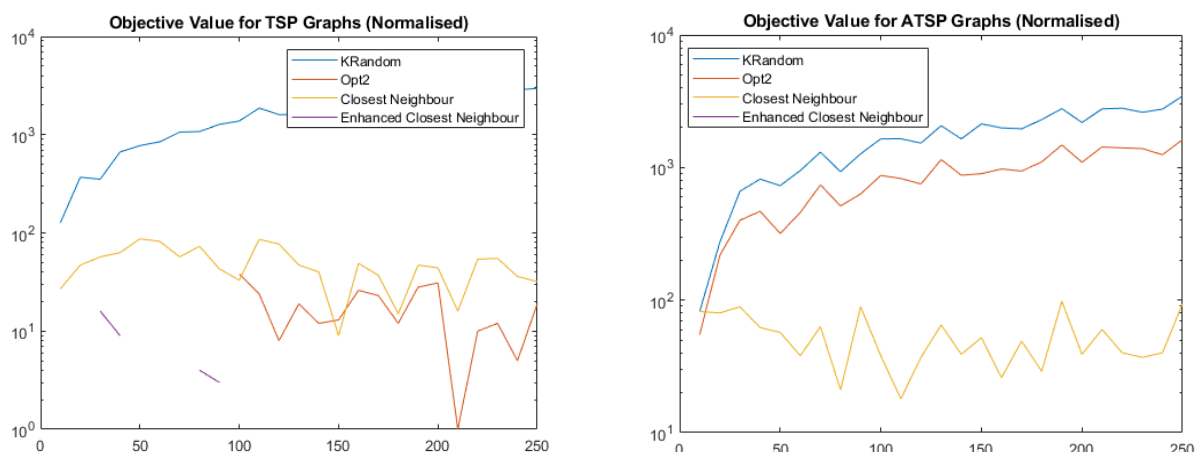
Przejdę teraz do prezentacji wyników eksperymentów.

Eksperyment A



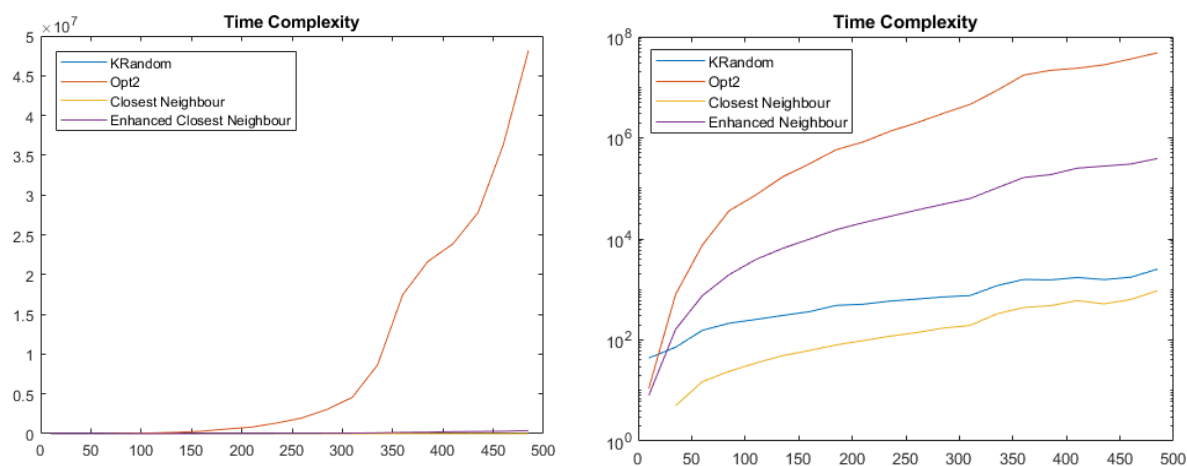
Oś Y w skali logarytmicznej. Wyraźnie widać, że dla obu typu problemów, algorytm k-random wyraźnie odbiega od pozostałych. Ciekawą obserwacją natomiast jest algorytm Opt2, który zdecydowanie lepiej radzi sobie z instancjami TSP, a dla ATSP jego wyniki rosną coraz szybciej. Natomiast oba algorytmy zachłanne CN i ECN radzą sobie podobnie dobrze z losowo generowanymi instancjami, z lekką wyższością ECN.

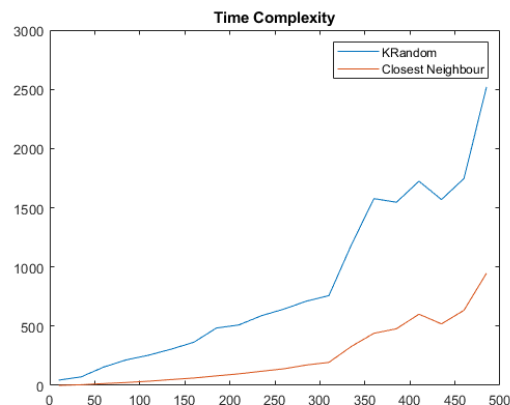
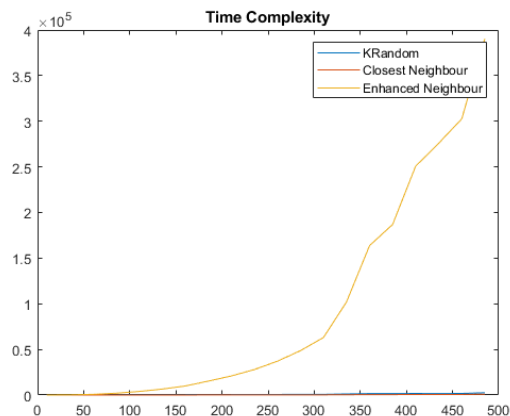
Eksperyment B



Oś Y w skali logarytmicznej. Eksperyment potwierdza wcześniejsze obserwacje. Algorytm K radzi sobie najgorzej, dla problemów TSP pozostałe 3 metaheurystyki dają zbliżone do siebie wyniki, ale algorytmy zachłanne radzą sobie lepiej niż Opt2 szczególnie dla mniejszych problemów. Natomiast dla problemów ATSP, Op2 już znacząco odbiega od wyników CN i ECN, a rozbieżności rosną razem z wielkością problemu. Zastosowanie skali logarytmicznej na osi Y spowodowało, że miejsca w których dany algorytm był najlepszy “znikają” z wykresu, ponieważ ich PRD jest tam równe 0%.

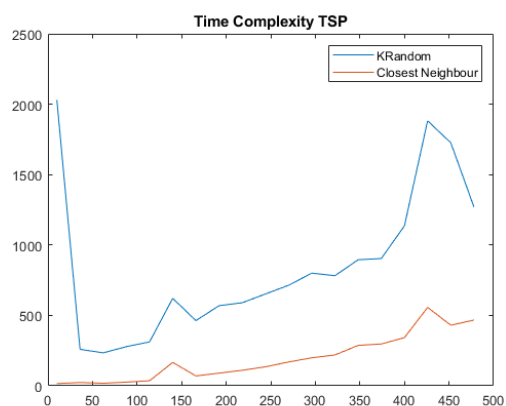
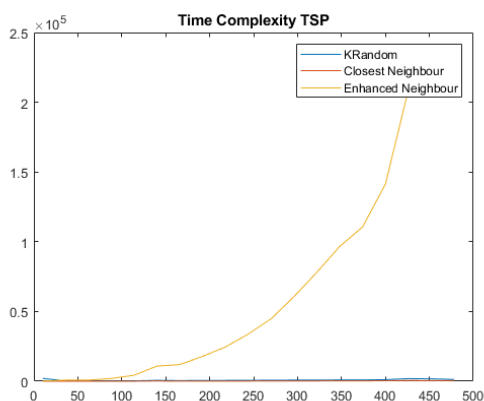
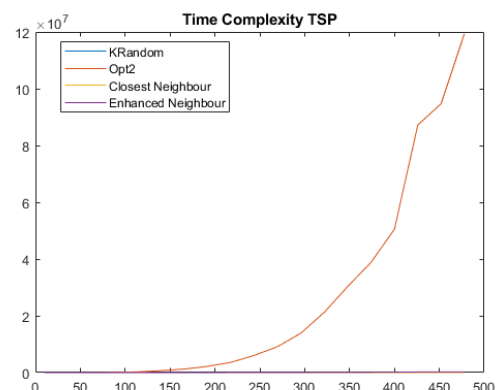
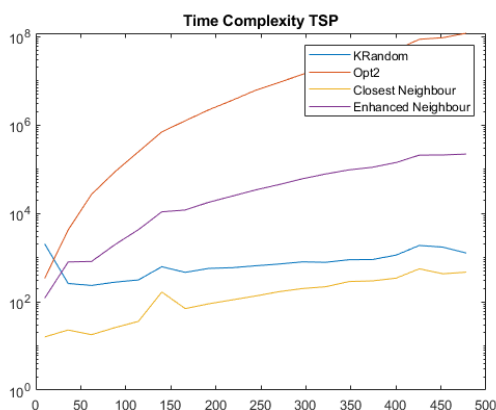
Eksperyment C



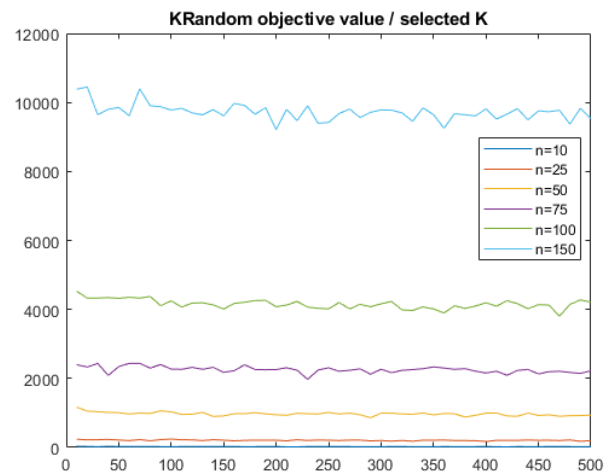
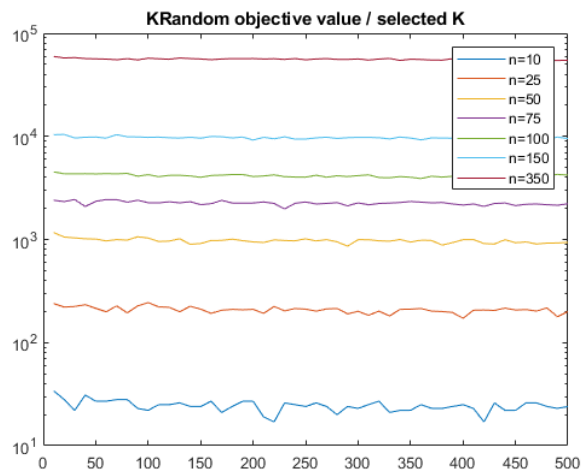


Wyniki eksperymentu pokazują, że algorytm Opt2 ma znacząco większą złożoność obliczeniową niż pozostałe. Dzięki analizie wykresu w skali logarytmicznej można zauważyć, że algorytm ECN działa wolniej niż k-Random oraz CN. Na koniec warto dodać, że generowanie losowo k rozwiązań działa wolniej niż algorytm CN. Widać też tutaj dla wszystkich algorytmów wzrost ponad liniowy - co się zgadza z tym że wraz ze wzrostem rozmiaru musimy wykonywać więcej obliczeń w każdym z algorytmów, dla 2Opta ten wzrost jest największy

Wykresy dla TSP:



Eksperyment D



Wyniki eksperymentu pokazują, że im większe k , tym mniejsze wahania wyniku końcowego.

Jako uzupełnienie wyników eksperymentów A i B dodamy, że dla realnych instancji problemu TSP, algorytm Opt2 radzi sobie lepiej niż ECN, jednak dane z biblioteki TSPLIB są zbyt rozproszone, żeby z nich samych stworzyć ładny wykres, a włączenie ich w badania nad losowymi instancjami zaburzą jednolitość eksperymentu.