AEM - Zadanie nr 4

Bartosz Sobkowiak 125342 Joanna Świda 138675

10.05.2020

1 Opis zadania

Rozważany problem to zmodyfikowana wersja problemu komiwojażera. Dany jest zbiór wierzchołków i macierz symetrycznych odległości między nimi. Zadanie polega na implementacji trzech metod - multiple start local search oraz dwóch rodzajów iterated local search.

2 Pseudokod

Wybraliśmy przeszukiwanie typu steepest, gdyż dawało najlepsze wyniki.
Używaliśmy wersji z zadania nr 2, czyli bez usprawnień.

Data: zbiór wierchołków, macierz odległości pomiędzy wierzchołkami
Result: najlepsze rozwiązanie
wygeneruj losowe rozwiązanie S
wyznacz elementy (wierzchołki) które nie znajdują się w rozwiązaniu
while *dopóki nie przekroczono czasu* **do**
  perturbate() - wykonaj niewielką perturbację na rozwiązaniu S:
  czyli wylosuj X punktów do zamiany, zamień te z rozwiązania z losowymi spoza rozwiązania
  // optymalne X zostało ustawione na wartość X=20
  wykonaj *LocalSearch* (z zadania 2) na rozwiązaniu S po wprowadzonych zmianach
  IF: jeśli rozwiązanie po wprowadzonych zmianach jest lepsze - zapisz jako najlepsze znalezione do tej pory rozwiązanie
end
return: najlepsze rozwiązanie

Algorithm 1: Iterated Local Search 1

Data: zbiór wierchołków, macierz odległości pomiędzy wierzchołkami
Result: najlepsze rozwiązanie
wygeneruj losowe rozwiązanie S
wyznacz elementy (wierzchołki) które nie znajdują się w rozwiązaniu
while *dopóki nie przekroczono czasu* **do**
  perturbate() - wykonaj niewielką perturbację na rozwiązaniu S:
  destroy() - usuń Y% punktów z rozwiązania
  repair() - napraw rozwiązanie w sposób zachłanny
  // optymalne Y zostało ustawione na wartość Y=20%
  wykonaj *LocalSearch* (z zadania 2) na rozwiązaniu S po wprowadzonych zmianach
  IF: jeśli rozwiązanie po wprowadzonych zmianach jest lepsze - zapisz jako najlepsze znalezione do tej pory rozwiązanie
end
return: najlepsze rozwiązanie

Algorithm 2: Iterated Local Search 2 - Destroy-Repair

MSLS bazuje na algorytmie z zadania 2

Data: zbiór wiercholków, macierz odległości pomiędzy wierzchołkami
Result: najlepsze rozwiązanie
for *wykonaj 100 razy* **do**
  wygeneruj losowe rozwiązanie
  while *dopóki znalezione rozwiązania są lepsze* **do**
    for *dla każdej pary indeksów w zakresie 0-99 w losowej kolejności* **do**
      podmień krawędzie według indeksów
      oblicz deltę dla zamienionych krawędzi
      jeśli delta jest mniejsza od najlepszej jak dotąd znalezionej delty: przypisz nową wartość najlepszej delty, to podmień w obecnym rozwiązaniu
    end
    jeśli zmienione rozwiazanie jest lepsze: wybierz je jako najlepsze
  end
  dodaj do listy rozwiązań
  posortuj i wybierz najlepsze
  IF: najlepsze rozwiązanie z tej iteracji jest lepsze niż najlepsze do tej pory znalezione rozwiązanie - ustaw je jako najlepsze znalezione, kontynuuj
end
return: najlepsze rozwiązanie

Algorithm 3: Multiple Start Local Search

3 Wyniki obliczeń i wizualizacje

Zbiór	Wersja	Typ	Min	Avg	Max
$kroA_{200}$	ILS	1	15279	17071	18271
$kroA_{200}$	ILS	2	14142	14814	15466
$kroA_{200}$	MSLS	-	16013	16935	18800
$kroB_{200}$	ILS	1	15867	17068	18062
$kroB_{200}$	ILS	2	14639	15147	15687
$kroB_{200}$	MSLS	-	16631	16947	18348

Tabela 1: Wartości rozwiązań

Zbiór	Wersja	Typ	Min	Avg	Max
$kroA_{200}$	ILS	1	200.0444	200.0628	200.1011
$kroA_{200}$	ILS	2	200.118	200.1526	200.2765
$kroA_{200}$	MSLS	-	200.013	200.126	200.098
$kroB_{200}$	ILS	1	200.0407	200.0598	200.1157
$kroB_{200}$	ILS	2	200.003	200.0487	200.0996
$kroB_{200}$	MSLS	-	200.1207	200.1581	200.1457

Tabela 2: Czasy trwania

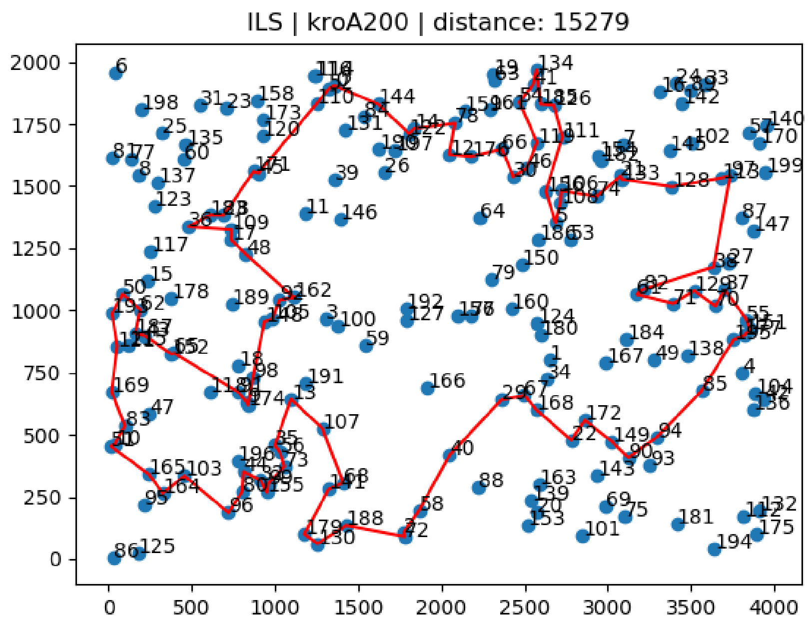
Średnia liczba iteracji lokalnego przeszukiwania dla ILS wynosiła 483 dla ILS2 (min:471 max:499) i 421 dla ILS1 (min:401 max:446).

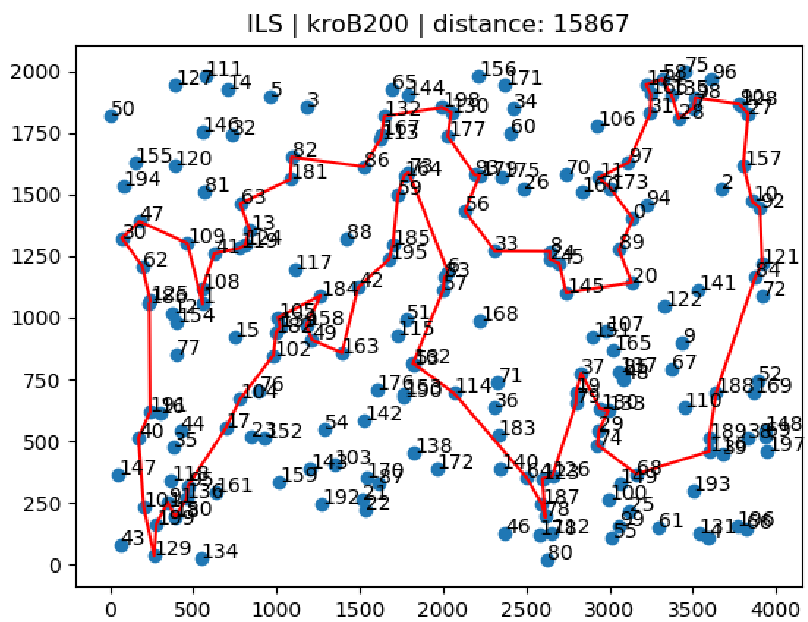
4 Wnioski

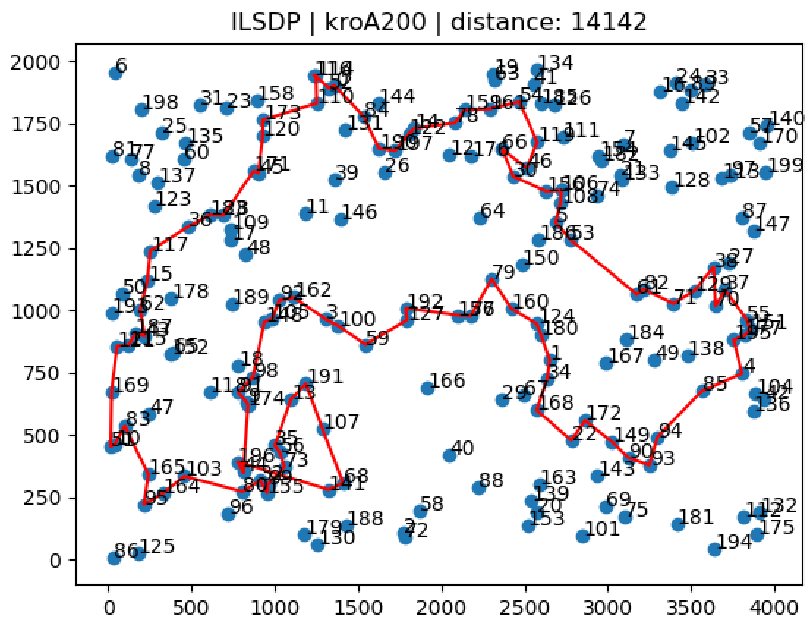
Z tabeli wyników można odczytać, że metody Iterated Local Seach dają lepsze wyniki niż metoda Multiple Start Local Search. Czas wykonywania po stronie ILS jest lepszy oczywiście dlatego, że w metodzie MLS zazwyczaj lokalnie przeszukujemy w pełni losowego rozwiązania. Jednakże w porównaniu do poprzednich metod czasy wykonywania zwiększyły się znacząco. Metoda ILS2 z naprawą rozwiązania okazała się nieco bardziej skuteczna od metody w której wymienialiśmy X wierzchołków. Prawdopodobnie wynikało to z tego, że w sposób losowy wymienialiśmy znaczącą liczbę wierzchołków - gdyby ta liczba była mniejsza (a procent "niszczonych" wierzchołków w ILS2 był większy), to wynik byłby odwrotny. Niemniej cel został osiągnięty, stosowanie metod ILS w tym przypadku jest lepszą metodą rozwiązania tego problemu.

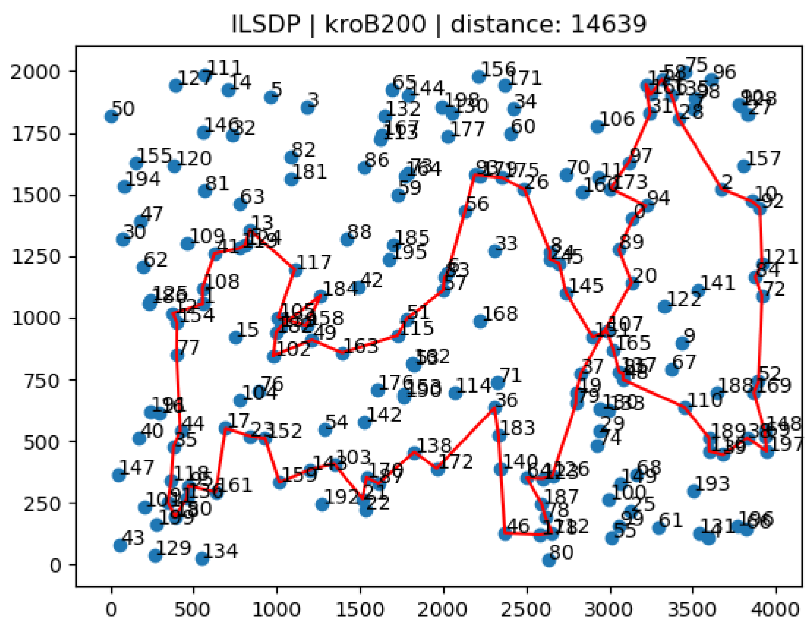
5 Kod programu

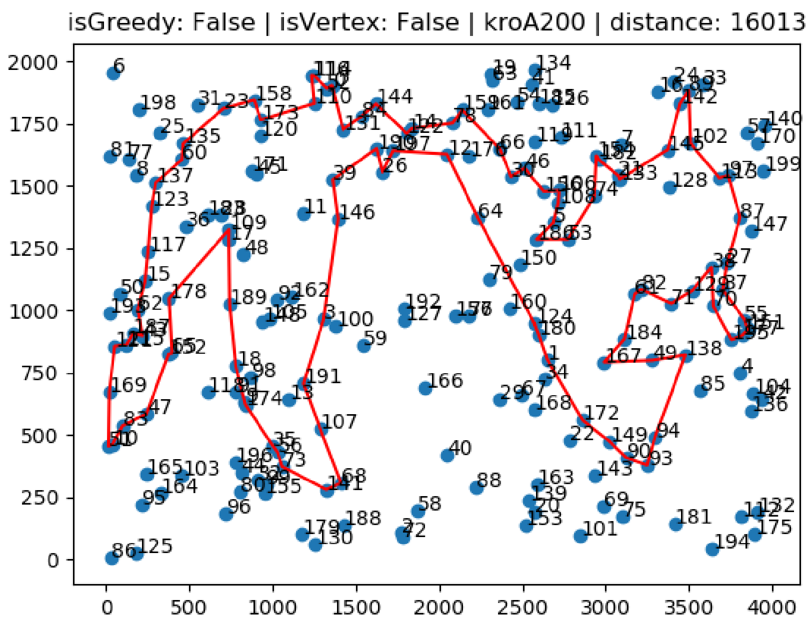
Repozytorium z kodem algorytmów dostępne jest pod: `https://github.com/bbbrtk/aem`

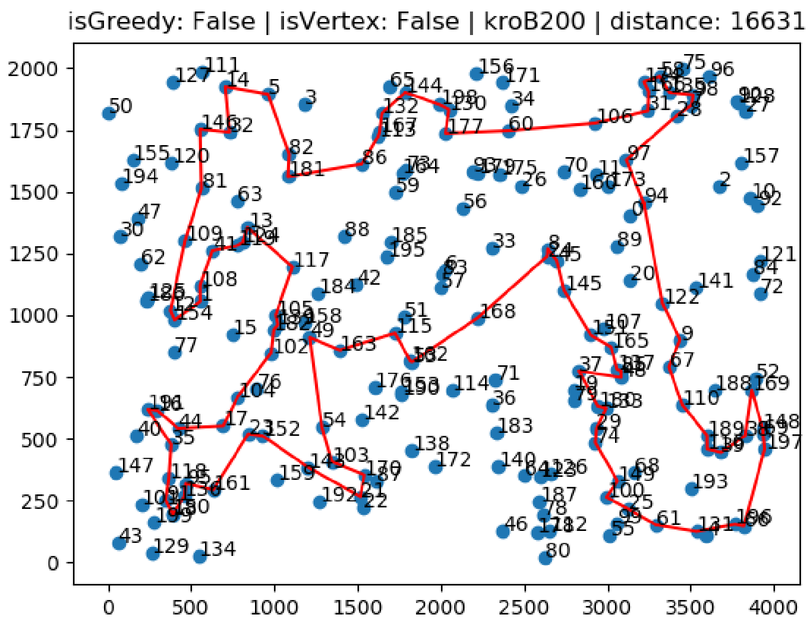

ILS | kroA200 | distance: 15279


ILS | kroB200 | distance: 15867


ILSDP | kroA200 | distance: 14142


ILSDP | kroB200 | distance: 14639


isGreedy: False | isVertex: False | kroA200 | distance: 16013


isGreedy: False | isVertex: False | kroB200 | distance: 16631