

Zadanie 6. Własna metoda i testy globalnej wypukłości

Oskar Kiliańczyk 151863 & Wojciech Kot 151879

1 Opis zadania

Zadanie polega na wygenerowaniu losowo 1000 optimów lokalnych, za pomocą zachłannej wersji przeszukiwania lokalnego, oraz bardzo dobrego rozwiązania za pomocą najlepszej do tej pory metody. Tą metodą powinna być własna metoda, której opracowanie również jest częścią tego zadania. Następnie dla każdego z lokalnych optimów należy obliczyć podobieństwo do rozwiązania referencyjnego oraz średnie podobieństwo do pozostałych optimów lokalnych ze zbioru, oraz współczynnik korelacji. Stosowane miary podobieństwa to:

1. Liczba par wierzchołków przydzielonych w obu rozwiązaniach razem do jednego cyklu
2. Liczba wspólnych par krawędzi

2 Własna Metoda

2.1 Pierwsza heurystyka konstrukcyjna

Już w ramach pierwszego sprawozdania zaimplementowaliśmy własną metodę, dlatego wykorzystamy ją tutaj w porównaniach. Dla przypomnienia oto jej pseudokod:

1. Utworzenie listy pozostałych wierzchołków (wszystkie możliwe, poza startowymi)
2. Utworzenie dwóch list zawierających odpowiednio dystanse każdego wierzchołka do pierwszego i do drugiego wierzchołka startowego
3. Sortowanie utworzonych uprzednio list dystansów wierzchołków
4. Aż do przydzielenia wszystkich wierzchołków z listy pozostałych wierzchołków wykonuje:
5. Zmianę decyzji, do którego zestawu wierzchołków obecnie będzie przydzielać wierzchołek (aby robić to naprzemiennie)
6. Wyszukuje pierwszy wierzchołek na liście dystansów, który nie został jeszcze przydzielony do żadnego zestawu i przydziela go tam

W skutek zastosowania takiego przydziału uzyskujemy dwa równo-liczne zbiory, oraz zapewniamy, że gdyby ilość badanych wierzchołków była nieparzysta, to zbiory będą różnić się długością najwyżej o 1.

Następnie wykorzystujemy tradycyjny algorytm rozbudowy cyklu na podstawie dwużalu, osobno na obu listach. Wygląda on następująco:

1. Algorytm zaczyna od ścieżki zawierającej wierzchołek startowy podwójnie
2. Dopóki w ścieżce nie znajdują się wszystkie wierzchołki z zadanego mu zestawu powtarza:
3. Dla każdego nieodwiedzonego wierzchołka oblicza koszty jego wstawienia
4. Następnie oblicza żal (dwużal) dla danego wierzchołka
5. Rozbudowuje cykl o wierzchołek z największym obliczonym żalem i zaznacza go jako odwiedzonego
6. Zwraca ścieżkę

2.2 Ulepszenie naszej heurystyki konstrukcyjnej

TODO Ulepszenie poprzez lepszy podział Trzeba wypróbować kilka, Kmeans, DBSCAN, etc.

2.3 Finalna własna metoda

TODO Finalnie zdecydowaliśmy się zastosować Large Neighbourhood Search

2.4 Wyniki obliczeniowe dla algorytmów

TODO TABELKA Z WYNIKAMI

2.5 Wizualizacje najlepszych wyników

TODO ALGO1 ALGO2 ALGO3 etc

2.6 Wnioski

Zaprojektowana przez nas heurystyka konstrukcyjna wykazuje się niebywałą szybkością i jest świetną opcją jeśli zależy nam na czasie. W przypadku, gdy dążymy do jak najdokładniejszych rozwiązań nie zważając tak na czas, to połączenie jej z TODO algorytmem TODO robi TODO

3 Testy globalnej wypukłości

Wygenerowaliśmy 1000 lokalnych optimów dla każdej z analizowanych instancji problemu, za pomocą lokalnego przeszukiwania w wersji zachłannej z losowych rozwiązań początkowych. Obliczyliśmy podobieństwa każdego lokalnego optimum do rozwiązania referencyjnego (bardzo dobrego) oraz średnie podobieństwo do pozostałych lokalnych optimów. Korzystamy z dwóch miar podobieństwa: Liczby par wierzchołków przydzielonych do tego samego cyklu w obu rozwiązaniach oraz z liczby wspólnych krawędzi w rozwiązaniach.

3.1 Wyniki dla instancji kroA200

3.2 Wyniki dla instancji kroB200

3.3 Wnioski

TODO Wnioski wnioseczki wnioskunie

4 Link do repozytorium

Kod źródłowy w repozytorium GitHub dostępny pod linkiem:
Repozytorium.