

Algorytmy grafowe – algorytmy zachłanne dla zagadnienia komiwojażera

Zadanie 1

- Zaimplementuj wybrany algorytm poszukiwania rozwiązania problemu TSP:
 - Alg. najbliższego sąsiada Nearest Neighbor (NN) – 3 pkt.,
 - Alg. G-TSP – 4 pkt.,
 - Alg. NEARIN / FARIN – 4 pkt.,
 - Alg. k-opt / 2-opt / 3-opt – 4 pkt.,
 - Alg. Christofidesa – 4 pkt.,
- Wklej kod źródłowy z komentarzami opisującymi realizowane kroki
- Algorytm ma zwracać cykl Hamiltona (listę wierzchołków) oraz jego długość (sumaryczną wagę)

Zadanie 2

- Krótko opisz jakie własności zadania (wagi, wierzchołek startowy) mogą być istotne dla algorytmu, z punktu widzenia możliwości znalezienia lub nie rozwiązania optymalnego
- Zdefiniuj dane oraz zaprezentuj uzyskane wyniki dla powyższych przypadków
- Przyjmij liczbę wierzchołków grafu $n=10$

Zadanie 3

- Określ złożoność obliczeniową algorytmu
- Krótko opisz jaka jest istotna różnica implementowanego algorytmu od pozostałych

Algorytm

Algorytm najbliższego sąsiada (NN).

1. Start z dowolnego wierzchołka (wierzchołek aktualny), który markujemy jako odwiedzony (wstawienie do rozwiązania).
2. Znajdź krawędź o najmniejszej wadze łączącą wierzchołek aktualny z nieodwiedzonymi wierzchołkami - v .
3. Przejdź do v (wierzchołek aktualny).
4. Oznacz v jako odwiedzony (wstawienie do rozwiązania).
5. Jeżeli wszystkie wierzchołki V są odwiedzane: STOP - zwróć sekwencję odwiedzonych wierzchołków.
6. Idź do kroku 2 .

Algorytm zachłany (G-TSP) – greed TSP

1. Uporządkuj łuki (krawędzie) wg wag w ciąg niemalejący.
2. Dla kolejnych łuków (krawędzi) i jeśli nie wszystkie wierzchołki V są odwiedzane dołącz kolejną do rozwiązania, jeżeli nie powoduje to powstania podcyklu (podkonturu)
3. Zwróć sekwencję odwiedzonych wierzchołków.

Algorytm Nearest-Insertion-Heuristik (NEARIN) / Farthest Insertion Heuristik (NEARIN)

1. Algorytm zaczyna się od rozwiązania (drogi komiwojażera), które składa się z jednego, losowo wybranego wężła.
2. Wybierz najbliższy / najdalszy wierzchołek v

(wierzchołek nieoznaczony najbliższy / najdalszy do wierzchołków rozwiązania)
3. Wstaw v do rozwiązania w "najtańszym" miejscu sekwencji
4. Jeżeli wszystkie wierzchołki V są odwiedzane: STOP - zwróć sekwencję odwiedzonych wierzchołków.
5. Idź do kroku 2.

Algorytm 2-optimalny (2-opt) – poprawy TSP, szczególny przypadek algorytmu k-opt

1. Utwórz rozwiązanie początkowe TSP (alg. konstrukcyjny, losowe).
2. Dla każdej pary krawędzi, z wyjątkiem krawędzi sąsiadujących ze sobą, usuń z cyklu dwie krawędzie i zastąp je innymi krawędziami tak, aby utworzyć inny cykl.
3. Wybierz modyfikację, która najbardziej skróciła trasę i powtórz krok 2. Jeśli żadna nie poprawiła trasy, zwróć sekwencję odwiedzonych wierzchołków.

Algorytm Christofidesa (graf pełny).

1. Utwórz minimalne drzewo rozpinające T dla grafu G .
2. Dla O - zbioru wierzchołków o nieparzystym stopniu w T - znajdź minimalne skojarzenie doskonałe M

(tzn. podzbiór M krawędzi grafu G o tej własności, że każdy wierzchołek jest M -nasycony = jest końcem krawędzi należącej do M - musi być parzysta liczba wierzchołków)

3. H jest multigrafem utworzonym z M i T .

4. Wyznacz cykl Eulera w grafie H

(graf H jest eulerowski ponieważ ma wszystkie wierzchołki parzystego stopnia).

5. Z cyklu Eulera zrobmy cykl Hamiltona poprzez pomijanie odwiedzonych wierzchołków (skracać).