# Wstęp

Problem komiwojażera należy do problemów optymalizacyjnych. Polega na znalezieniu cyklu Hamiltona w pełnym grafie ważonym. W takim grafie istnieje połączenie w ramach każdej pary wierzchołków. Cykl Hamiltona to taki cykl w którym każdy wierzchołek oprócz pierwszego zostaje odwiedzony dokładnie raz, cykl zaczyna się i kończy w wierzchołku początkowym.

Typowe jest przedstawienie problemu jako szukanie najszybszej trasy by objechać n miast.

# Szczegóły ćwiczenia

Zaimplementowano programy w języku matlab które rozwiązują problem komiwojażera generując kolejne stany potomne poczynając od stanu początkowego. Pierwszy program przeszukuje graf miast wszerz, a drugi w głąb. Istnieją też dwie wersje funkcji generującej stan potomny funkcja naiwna i funkcja zachłanna. Użycie funkcji zachłannej nie gwarantuje uzyskania rozwiązania optymalnego!

Poniżej zostały przedstawione pomiary czasu działania algorytmu dla metody brute-force i zachłannej. Uruchamianie programu dla ilości miast większej niż 9 nie było opłacalne na dostępnym sprzęcie.

|  |  |
| --- | --- |
| Ilość miast | czas[s] |
| 4 | 0,043 |
| 5 | 0,085 |
| 6 | 0,241 |
| 7 | 1,15 |
| 8 | 8,489 |
| 9 | 155,816 |

|  |  |
| --- | --- |
| Ilość miast | czas[s] |
| 4 | 0,045 |
| 5 | 0,049 |
| 6 | 0,043 |
| 7 | 0,057 |
| 8 | 0,044 |
| 9 | 0,057 |
| 15 | 0,06 |
| 20 | 0,055 |
| 40 | 0,14 |
| 100 | 0,459 |

Jak widać metoda zachłanna jest o wiele szybsza od podejścia siłowego, w przeciwieństwie do niej nie gwarantuje jednak otrzymania rozwiązania optymalnego.

Implementacja BFS okazała się o wiele wolniejsza od implementacji DFS opartej o stos, prawdopodobnie z powodu mało wydajnej implementacji kolejki. Dla 9 miast algorytm BFS trwał 1090 sekund, czyli prawie 10 razy wolniej niż DFS dla takiej samej ilości wierzchołków.