

Algorytmy metaheurystyczne 2

P. Cegieła, W. Sęk

5 maja 2022

1 Teoretyczna złożoność

Warunkiem wyjścia w naszym algorytmie było przekroczenie $15n$ iteracji, gdzie $n = |V|$ lub n ruchów bez zmiany na lepsze rozwiązanie. Rozważmy najgorszy możliwy przypadek, gdzie wykonujemy $15n$ iteracji. Niech k to długość listy tabu. Implementacja listy tabu za pomocą VecDeq pozwala na dostęp do i -tego elementu w czasie stałym, a usuwanie i dodawanie elementów w czasie liniowym.

W każdym kroku algorytmu przeglądamy wszystkich $\frac{n(n-1)}{2}$ sąsiadów danego rozwiązania i dla każdego sprawdzamy z $O(k)$ czy jest na liście tabu. Sprawdzenie o ile sąsiad zmienia wartość permutacji jest stały (dla *invert* liczymy wcześniej z $O(n^2)$ pomocnicze tablice. Niech $O(l)$ to złożoność przybliżenia początkowego (dla *2-opt* $O(n^3)$). Wybieramy najlepszego z nich. Ostatecznie mamy (dla k stałego i 2-opta):

$$O\left(15n \cdot \frac{n(n-1)}{2} \cdot k + l\right) = O(n^3)$$