|  |  |
| --- | --- |
| Sprawozdanie 1 - Heurystyki konstrukcyjne | |
| **Imie Nazwisko** | **Nr indexu** |
| Patryk Jedlikowski  Mikołaj Sienkiewicz | 136723  136309 |

## Opis zadania

## Algorytm zachłanny inspirowany metodą najbliższego sąsiada

Inicjalizacja zmiennych: (min\_result,max\_result,results[],dataset)

distance=Oblicz\_macierz\_dystansu(dataset)

distance(przekatna)=nieskonczonosc

Wybierz losowo wierzchołki startowe dla cycleA oraz cycleB

distance(kolumna=wierzchołki\_startowe) = nieskonczonosc

Dla i od 0 do 98:

    wybierz co drugi cykl do rozbudowy:

    kandydaci=[]

    Dla każdego licznik\_iteracji,index\_wierzchołka z wybranego cyklu:

        kandydaci.append([index\_NN,wartosc\_NN,licznik\_iteracji])

    najlepszy=min(kandydaci,klucz=kandydaci[1])

    wybrany\_cykl.insert(najlepszy[2],index\_wstawienia=najlepszy[0])

    distance(kolumna=najlepszy[0]) = nieskonczonosc

## Algorytm zachłanny inspirowany metodą rozbudowy cyklu

Wybierz losowo wierzchołki startowe dla cycleA oraz cycleB

Dla i od 0 do 98:

    wybierz co drugi cykl do rozbudowy

    minDistance = nieskończoność

    Dla vertex od 0 do 100:

        if vertex w użyciu:

            pomiń iteracje

        distance, newCycle = findNewCycle(vertex)

        if distance < minDistance:

            minDistance = distance

            minCycle = newCycle

    Podmień cycle na minCycle w aktualnie rozbudowywanym cyklu

Funkcja - findNewCycle(vertex):

    minDistance = nieskończoność

    Dla i od 0 do długość rozbudowywanego cyklu + 1:

        wstaw sprawdzany vertex w miejsce "i" do aktualnego cyklu

        distance = obliczDługośćCyklu()

        if distance < minDistance:

            minDistance = distance

            minVertex = vertex

            minPath = path

    return minDistance, minPath

## Algorytm typu „regret heuristics” (2-regret)

## Tabela prezentująca wyniki eksperymentu obliczeniowego

## Wizualizacje najlepszych rozwiązań

## Wnioski

## Kod programu