

Implementacja algorytmu LocalSearch w kodzie:

Dane są wczytywane z pliku "xqf131.txt". Plik ten zawiera informacje o punktach na płaszczyźnie, z których każdy reprezentowany jest przez współrzędne (x, y).

Na podstawie wczytanych punktów obliczane są wagi krawędzi między nimi. Następnie algorytm Kruskala jest używany do stworzenia minimalnego drzewa rozpinającego. Wagi krawędzi reprezentują odległości między punktami.

Algorytm Local Search dla problemu komiwożera (TSP) zastosowany jest na wcześniej utworzonym minimalnym drzewie rozpinającym. Algorytm iteruje się, zamieniając losowo wybrane pary punktów w bieżącym rozwiązaniu. Zamiana jest dokonywana w taki sposób, aby minimalizować różnicę w długości cyklu.

Algorytm kontynuuje iteracje, dopóki istnieje para punktów, którą można zamienić w celu poprawy rozwiązania.

W programie używam generatora mt19937 liczb losowych do losowego wyboru par punktów i podejmowania losowych decyzji w przypadku równych różnic.

Algorytm jest uruchamiany wielokrotnie (zdefiniowane przez zmienną **times**), a wyniki są uśredniane.

Zastanów się, dlaczego ten algorytm dla grafów euklidesowych zawsze zwróci rozwiązanie bez krzyżujących się krawędzi. Znajdź prosty przykład, że nie jest to prawdą, jeśli stosujemy zaokrąglenia takie jak w danych, które używamy:

Jeżeli korzystamy z zaokrągleń, tak jak w naszych danych, istnieje możliwość wystąpienia pewnych niedokładności, które potencjalnie mogą wpłynąć na relacje między odległościami. To z kolei może prowadzić do sytuacji, w których algorytm lokalnego przeszukiwania nie zawsze dostarcza rozwiązania bez krzyżujących się krawędzi.

Przykładem takiej sytuacji, gdzie zaokrąglenia generują krzyżujące się krawędzie, może być sytuacja, w której dla idealnej geometrii (bez zaokrągleń) krawędzie pozostają nieskrzyżowane. Jednakże z uwagi na zaokrąglenia, które wprowadzają pewne błędy, algorytm może napotykać trudności w ich uniknięciu.

zad.1

Zadanie polegało na wyliczeniu minimalnego drzewa rozpinającego i wykonaniu $[Vn]$ razy (n to liczba wierzchołków) następujący algorytm:

(a) Wylosuj wierzchołek.

(b) Skonstruuj cykl zaczynając przegląd drzewa od wylosowanego wierzchołka.

(c) Zastosuj algorytm Local Search do tak uzyskanego cyklu.

Dla każdego danych podano wagę minimalnego drzewa rozpinającego, średnią wartość uzyskanego rozwiązania, średnią liczbę kroków poprawy oraz najlepsze uzyskane rozwiązanie.

Przykład uruchomienia programu dla XQF131:

MST Weight: 474
Average Steps: 53.6667
Average Cost: 603.667
Minimum Cost: 591
Best Solution: (61,47) (61,45) (57,44) (51,47) (51,45) (41,36) (41,35) (41,34) (41,32) (38,30) (34,29) (34,26) (33,26) (32,26) (33,29) (32,31) (33,31) (34,31) (35,31) (38,34) (34,38) (34,41) (28,47) (28,43) (28,40) (28,34) (28,30) (28,28) (25,29) (25,28) (25,26) (25,24) (25,23) (25,22) (28,20) (28,16) (25,15) (18,13) (18,15) (18,17) (18,19) (18,21) (18,23) (18,25) (18,27) (18,29) (18,31) (18,33) (18,35) (18,37) (18,39) (18,41) (18,42) (18,44) (18,45) (15,43) (15,37) (15,31) (15,25) (15,19) (15,13) (12,10) (11,10) (10,10) (9,10) (5,13) (5,19) (5,25) (5,31) (5,37) (5,43) (0,39) (0,27) (0,26) (0,13) (5,8) (2,0) (8,0) (12,5) (15,8) (18,11) (25,11) (25,9) (34,5) (33,15) (34,15) (35,17) (38,16) (38,20) (40,22) (41,23) (48,22) (48,27) (56,25) (57,25) (57,12) (48,6) (63,6) (64,22) (74,24) (77,21) (74,20) (74,16) (71,16) (71,13) (71,11) (74,12) (78,10) (80,10) (79,10) (74,6) (80,5) (84,6) (107,27) (81,17) (84,20) (84,24) (84,29) (84,34) (84,38) (80,41) (78,39) (79,37) (78,35) (79,33) (78,32) (74,29) (74,35) (74,39) (71,45) (71,47)

zad.2

Zadanie polegało na wykonaniu algorytm Local Search dla n losowych permutacji. Dla każdych danych podano średnią wartość uzyskanego rozwiązania, średnią liczbę kroków poprawy oraz najlepsze uzyskane rozwiązanie.

Przykład uruchomienia programu dla XQF131:

MST Weight: 474
Average Steps: 133.206
Average Cost: 605.489
Minimum Cost: 571
Best Solution: (51,47) (51,45) (41,36) (41,35) (41,34) (41,32) (38,34) (38,30) (35,31) (34,29) (33,26) (32,26) (34,26) (33,29) (34,31) (33,31) (32,31) (34,38) (34,41) (28,47) (28,43) (28,40) (28,34) (28,30) (28,28) (25,29) (25,28) (25,26) (25,24) (25,23) (25,22) (28,20) (28,16) (25,15) (18,15) (18,17) (15,19) (18,19) (18,21) (18,23) (18,25) (15,25) (18,27) (18,29) (15,31) (18,31) (18,33) (18,35) (18,37) (15,37) (18,39) (18,41) (18,42) (18,45) (18,44) (15,43) (5,43) (0,39) (5,37) (5,31) (0,27) (0,26) (5,25) (5,19) (5,13) (0,13) (5,8) (2,0) (8,0) (12,5) (10,10) (9,10) (11,10) (12,10) (15,8) (18,11) (15,13) (18,13) (25,11) (25,9) (34,5) (33,15) (34,15) (35,17) (38,16) (38,20) (40,22) (41,23) (48,22) (48,27) (56,25) (57,25) (64,22) (57,12) (48,6) (63,6) (74,6) (80,5) (84,6) (80,10) (79,10) (78,10) (74,12) (71,11) (71,13) (71,16) (74,16) (74,20) (77,21) (74,24) (74,29) (74,35) (78,35) (78,32) (79,33) (84,29) (84,24) (81,17) (84,20) (107,27) (84,34) (84,38) (80,41) (79,37) (78,39) (74,39) (71,45) (71,47) (61,47) (61,45) (57,44)