

Problem komiwojażera

Problem komiwojażera jest to zagadnienie optymalizacyjne, polegające na znalezieniu minimalnego cyklu Hamiltona w pełnym grafie ważonym. Cyklem hamiltona nazywamy taki cykl, w którym każdy wierzchołek oprócz pierwszego został odwiedzony tylko raz. W grafie ważonym występują połączenia między każdą parą wierzchołków.

Implementacja zadania

Problem komiwojażera zaimplementowałem w języku C++ używając algorytmów:

- Najbliższego sąsiada
- Brute force

Algorytm najbliższego sąsiada

Algorytm ten polega na przeszukiwaniu nie odwiedzonych jeszcze wierzchołków i wybraniu najbliższego wierzchołka.

Implementacja

```
void nearestNeighbor(std::vector<City> c)
{
    City actual = c[0];
    c.erase(c.begin());
    std::vector<City> result;
    result.push_back(actual);
    std::clock_t t = std::clock();
    while(!c.empty())
    {
        int min = 0;
        for (int i = 1; i < c.size(); i++)
        {
            if ( (actual - c[i]) < (actual - c[min]) )
            {
                min = i;
            }
        }
        result.push_back( c[min] );
        actual = c[min];
        c.erase(c.begin() + min);
    }
    t = std::clock() - t;
    std::cout << "Nearest Neighbor Distance: " << distance(result) << " Time: " <<
    ((double)t)/CLOCKS_PER_SEC << "s" << std::endl;
}
```

Algorytm Brute force

Algorytm polega na siłowym dopasowaniu najlepszej trasy - sprawdza każdą możliwą kombinację.

Implementacja

```
void bruteForce(std::vector<City> c)
{
    std::vector<City> result = c;
    double minDistance = distance(c);
    std::clock_t t = std::clock();
    while(std::next_permutation(c.begin(), c.end()) )
    {
        double dist = distance(c);
        if (dist < minDistance )
        {
            minDistance = dist;
            result = c;
        }
    }
    t = std::clock() - t;
    std::cout << "Brute Force Distance: " << distance(result) << " Time: " <<
((double)t)/CLOCKS_PER_SEC << "s" << std::endl;
}
```

Porównanie wyników

W tabeli przedstawiony jest czas oraz dystans dla każdego z algorytmów:

Liczba wierzchołków	Najbliższy sąsiad	BruteForce
5	0s, 140.422	0s, 129.804
7	0s, 154.976	0.003s, 130.831
8	0s, 176.653	0.027s, 151.447
9	0s, 177.774	0.213s, 153.331
10	0s, 183.421	1.995s, 158.978
11	0s, 204.279	26.338s, 179.226
12	0s, 244.385	318.496s, 219.333
1000	0.062s	-
10000	6.161s	-
20000	26.052s	-
50000	155.107s	-
70000	318.578s	-
90000	539.978s	-

Uruchomienie algorytmu brute force dla większej ilości wierzchołków nie było opłacalne na dostępnym sprzęcie.

Wnioski

Algorytm Brute force jest dużo bardziej czasochłonny ale oblicza najkrótszą trasę. Już dla 12 wierzchołków algorytm brute force zajmuje bardzo dużą ilość czasu. Algorytm najbliższego sąsiada osiąga taki czas przy około 70.000 wierzchołków.

Bartosz Ujazdowski - 202322