sdizo – Projekt III

# Wstęp teoretyczny

Problem komiwojażera opisuje znajdywanie minimalnego cyklu Hamiltona w grafie pełnym. Innymi słowy: mamy dane miasta i odległości między nimi. Komiwojażer ma za zadanie obejść je wszystkie jak najkrótszą drogą.

Dyskretny problem plecakowy jest jednym z najpopularniejszych problemów optymalizacyjnych. Nazwa pochodzi od problemu wyboru przedmiotów, by ich wartość była największa, ale rozmiar nie za duży, by zmieściły się w plecaku o zadanej pojemności.

Przegląd zupełny jest obliczeniowo najmniej efektywnym z testowanych algorytmów, jednakże daje najlepszy wynik. Jego złożoność wynosi O(2n) dla plecaka i O(n!) dla komiwojażera. Testuje on wszystkie możliwe rozwiązania i z nich wybiera najlepsze.

Algorytm zachłanny w danym kroku / momencie dokonuje najlepszego z możliwych wyborów, bez sprawdzania sensu podjętej decyzji w następnych krokach.

Przeszukiwanie lokalne tworzy rozwiązanie problemu, lepszym lub gorszym, które następnie jest lokalnie poprawiane.

Algorytm dynamiczny ma zmaksymalizować sumę wartości elementów, przy zachowaniu sumy ich wagi mniejszej bądź równej zadanej wadze.

# Plan eksperymentu

Do rozwiązania pierwszego problemu zostały użyte 4 algorytmy: przegląd zupełny, algorytm zachłanny (wg kryterium wartości i stosunku wartości do wagi) oraz algorytm dynamiczny. Problem komiwojażera został rozwiązany przez przegląd zupełny, algorytm zachłanny i przeszukiwanie lokalne. Dla każdego wykonano po 100 testów. Problem plecakowy był rozwiązywany dla plecaków o pojemności 5, 10 i 15 i dla liczby elementów 5, 10, 15, 20 i 25, zaś problem komiwojażera dla 5, 6, 7, 8 i 9 miast.

Dane do algorytmów zostały wygenerowane automatycznie

# Wyniki

Celem lepszego zobrazowania wyników pomiarów, czasy na wykresach zostały przedstawione w skali logarytmicznej.

## Problem komiwojażera

Tabela 1 – pomiary czasu dla problemu komiwojażera (w ms)  


Wykres 1 – pomiary czasu dla problemu komiwojażera

Z uwagi na złożoność przeglądu zupełnego równą O(n!) niewykonalne wręcz staje się rozwiązanie problemu komiwojażera tą metodą dla dużych liczb miast do odwiedzenia. Dla 50 miast sięga rzędu 1053, dla 100 10146, a dla 1000 miast 102556 lat. Te dane mogą się różnić w zależności od komputera i trybu pracy (debug/relase), jednakże to i tak niesłychanie dużo.

## Dyskretny problem plecakowy

### Dla stałego rozmiaru plecaka

Tabela 2 – pomiary czasu dla problemu plecakowego pogrupowane wg rozmiarów plecaka – 5, 10 i 20 (w ms)  


Wykres 2 – pomiary czasu dla problemu plecakowego z plecakiem o rozmiarze 5

Wykres 3 – pomiary czasu dla problemu plecakowego z plecakiem o rozmiarze 10

Wykres 4 – pomiary czasu dla problemu plecakowego z plecakiem o rozmiarze 20

### Dla stałej liczby przedmiotów

Tabela 3 – pomiary czasu dla problemu plecakowego pogrupowane wg liczby przedmiotów – 5, 10, 15, 20 i 25 (w ms)  


Wykres 5 – pomiary czasu dla problemu plecakowego dla 5 przedmiotów

Wykres 6 – pomiary czasu dla problemu plecakowego dla 10 elementów

Wykres 7 – pomiary czasu dla problemu plecakowego dla 15 elementów

Wykres 8 – pomiary czasu dla problemu plecakowego dla 20 elementów

Wykres 9 – pomiary czasu dla problemu plecakowego dla 25 elementów

# Wnioski

W obu przypadkach, w przeglądzie zupełnym widać szybko rosnącą złożoność czasową – zgodnie z założeniami. Im więcej miast lub przedmiotów do spakowania, tym czas wyszukiwania optymalnego rozwiązania drastycznie rośnie. Kiedy liczba elementów jest stała, a zmienia się rozmiar plecaka, różnica w czasie działania jest już znacznie mniej dostrzegalna.

Można zauważyć też, że najefektywniejszym algorytmem jest algorytm zachłanny, a tuż po nim plasuje się dynamiczny. W przypadku komiwojażera drugie jest przeszukiwanie lokalne.

Momentami można zauważyć nagły spadek czasu wykonywania się algorytmów (poza przeglądem zupełnym). Nie udało mi się odnaleźć przyczyny tych spadków i być może jest to spowodowane optymalizacjami implementowanymi przez środowisko programistyczne, na którym pracuję (Visual Studio 2013).