

PAMSI - Sprawozdanie 8

Filip Guzy 218672

17 maja 2016

Graf nieskierowany ważony i algorytmy branch and bound

Graf nieskierowany ważony jest strukturą danych zbudowaną z wierzchołków i krawędzi, których wartości opisują relacje pomiędzy poszczególnymi wierzchołkami. W celu przetestowania algorytmów wyszukiwania najbardziej optymalnych tras pomiędzy wierzchołkami z rodziny branch and bound (tł. metoda podziału i ograniczeń) zaimplementowano je jako metody wyżej wspomnianego grafu. W celu ułatwienia realizacji obydwu przeszukań uzupełniono całą strukturę o macierz kosztów opisującą wagi poszczególnych krawędzi przy zachowaniu istniejącej już listy sąsiedztwa. Algorytm branch and bound został zaimplementowany w dwóch wersjach: bez listy rozszerzonych wierzchołków (extended list) oraz z listą. Realizacja obu przeszukań została zrealizowana na bazie zaimplementowanej w tym celu kolejki priorytetowej.

Pomiary czasów działania i rozszerzanych ścieżek

Algorytmy branch and bound charakteryzują się różną złożonością w zależności od kilku czynników. Przede wszystkim należy wziąć pod uwagę to, że wybrana implementacja jest oparta na algorytmie przeszukiwania BFS (Best-First Search). W tym przypadku złożoność czasowa działania algorytmu jest zależna od liczby sąsiadów dowolnego wierzchołka, do których koszt przejścia jest zbliżony do kosztu osiągnięcia tego wierzchołka lub jest równy zero. Biorąc pod uwagę fakt, że prawdopodobieństwa wylosowania wartości wag były równe (rozkład równomierny) oraz że każdy wierzchołek posiadał kilku sąsiadów można przyjąć, że większość wierzchołków miała sąsiada o zbliżonym koszcie przejścia. Zgodnie z takimi założeniami, złożoność czasowa testowanego przypadku powinna wynosić $O(n^2)$, gdzie n jest liczbą wierzchołków. Poniżej przedstawiono trzy różne możliwości złożoności czasowej tego algorytmu. Oznaczmy jako L liczbę sąsiadów pojedynczego wierzchołka o zbliżonym koszcie przejścia.

1. $O(n)$, gdy $L > 1$
2. $O(n^2)$, gdy $L = 1$
3. $O(k^n)$, gdy $L < 1$ i k jest pewną stałą wartością.

W celu sprawdzenia przewidywanej złożoności przeprowadzono pomiary czasu oraz rozwiniętych ścieżek dla grafów o rozmiarach 10, 100, 1000, 10000. W poniższych tabelach przedstawiono pomiary dla wersji branch and bound bez listy rozszerzonych wierzchołków oraz z listą, a także wykresy porównujące obie wersje implementacji.

Tabela 1: Branch and bound - czas wyszukiwania

Ilość danych	10	100	1000	10000
Czas [ms]	0,049	0,155	19,074	2738,1
	0,0370001	0,176	19,601	2753,46
	0,0359999	0,16	21,368	2686,3
	0,036	0,152	22,87	2780,22
	0,041	0,152	24,204	2734,58
	0,041	0,162	21,393	2847,54
	0,039	0,16	22,603	2767,24
	0,046	0,156	22,526	2740,64
	0,0400001	0,161	23,377	2758,81
	0,041	0,166	23,52	2758,19
Średnia	0,04060001	0,16	22,0536	2756,508

Tabela 2: Branch and bound - rozwinięte ścieżki

Ilość danych	10	100	1000	10000
Rozwinięte ścieżki	18	239	2489	24993
	14	239	2478	24989
	14	247	2498	24989
	14	239	2493	24980
	14	239	2483	24989
	30	247	2493	24989
	14	247	2489	24994
	25	243	2493	24989
	18	250	2498	24993
	18	251	2489	24989
Średnia	17,9	244,1	2490,3	24989,4

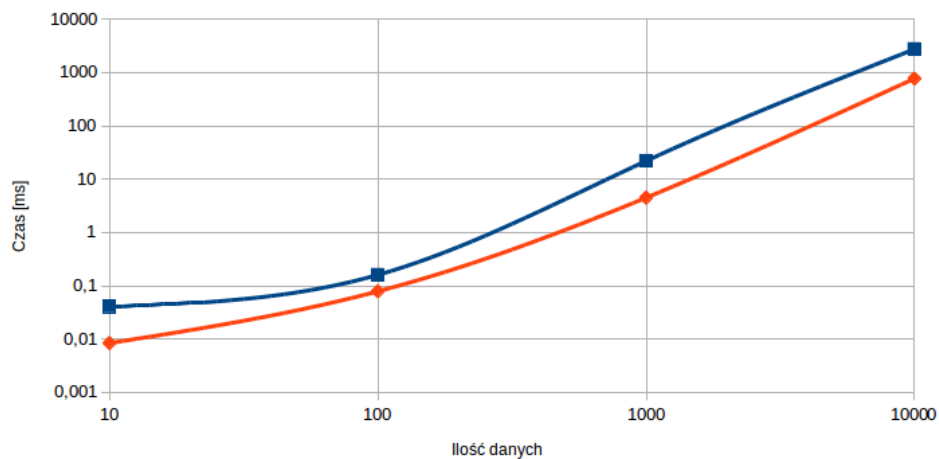
Tabela 3: Branch and bound with extended list - czas wyszukiwania

Ilość danych	10	100	1000	10000
Czas [ms]	0,00899994	0,096	4,034	761,906
	0,00800002	0,0749999	3,942	781,769
	0,00800002	0,0749999	4,09	770,823
	0,00800002	0,0749999	5,397	798,609
	0,00800002	0,0749999	5,01	781,514
	0,00800002	0,076	3,907	748,453
	0,00800002	0,0799999	5,611	768,92
	0,00900006	0,0779999	4,164	787,121
	0,00900006	0,077	4,785	776,154
	0,00899994	0,0779999	4,119	778,916
Średnia	0,008400012	0,07849993	4,5059	775,4185

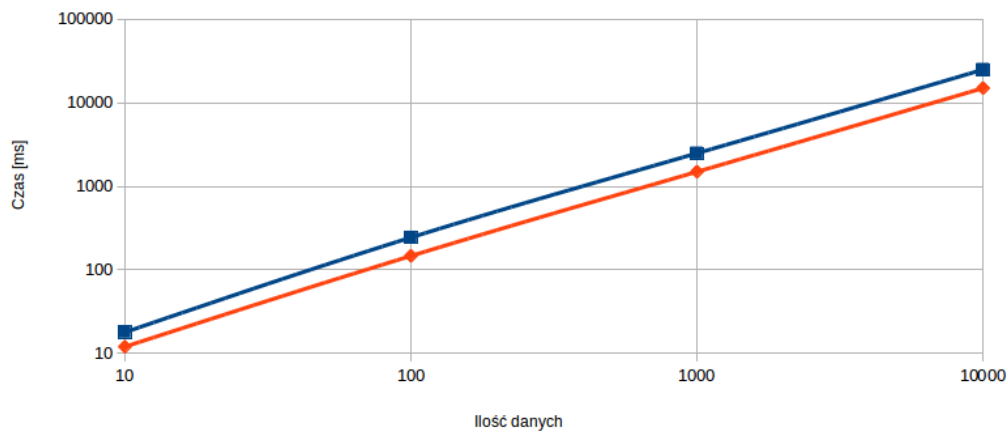
Tabela 4: Branch and bound with extended list - rozwinięte ścieżki

Ilość danych	10	100	1000	10000
Rozwinięte ścieżki	12	147	1497	14994
	11	147	1493	14997
	11	146	1497	14997
	11	147	1497	14994
	11	147	1494	14997
	15	146	1497	14997
	11	146	1497	14997
	14	147	1497	14997
	12	147	1496	14997
	12	147	1497	14997
Średnia	12	146,7	1496,2	14996,4

Porównanie złożoności czasowej B&B (niebieski) oraz B&B with extended list (pomarańczowy)



Porównanie rozwijanych ścieżek przez B&B (niebieski) oraz B&B with extended list (pomarańczowy)



Jak można zauważyć z rysunków, przeszukiwanie branch and bound wykonuje się w czasie kwadratowym, zatem złożoność jest zgodna z teorią. Ilość rozwijanych ścieżek rośnie liniowo. Po przeanalizowaniu obu wersji algorytmu można stwierdzić, że zastosowanie listy odwiedzonych wierzchołków umożliwiło znaczne polepszenie czasu wykonywania oraz zmniejszenie ilości rozwijanych ścieżek, przez co algorytm stał się skuteczniejszy w wyszukiwaniu optymalnej ścieżki.

Wnioski

Z przeprowadzonych pomiarów wynika, że algorytmy branch and bound w skuteczny sposób umożliwiają odnalezienie optymalnej ścieżki pomiędzy dwoma wierzchołkami. Zaimplementowane przeszukiwania mają złożoność obliczeniową zgodną z teorią. Zastosowanie dodatkowej listy zawierającej informacje o odwiedzonych wierzchołkach umożliwia znaczne polepszenie efektywności wyznaczania optymalnych tras.

Bibliografia

Weixiong Zhang, *Branch-and-bound Search Algorithms and Their Computational Complexity*, May 1996