

PROJEKTOWANIE EFEKTYWNYCH ALGORYTMÓW

PROJEKT

07/10/2021

252736 Hutnik Szymon

Held-Karp (2)

Strona	Spis treści
2	Treść zadania
3	Opis metody
4	Opis algorytmu
6	Dane testowe
7	Procedura badawcza
9	Wyniki
11	Analiza

1. Treść zadania

Opracować, napisać, zbadać rozwiązanie problemu komiwojażera w wersji optymalizacyjnej algorytmem programowania dynamicznego.

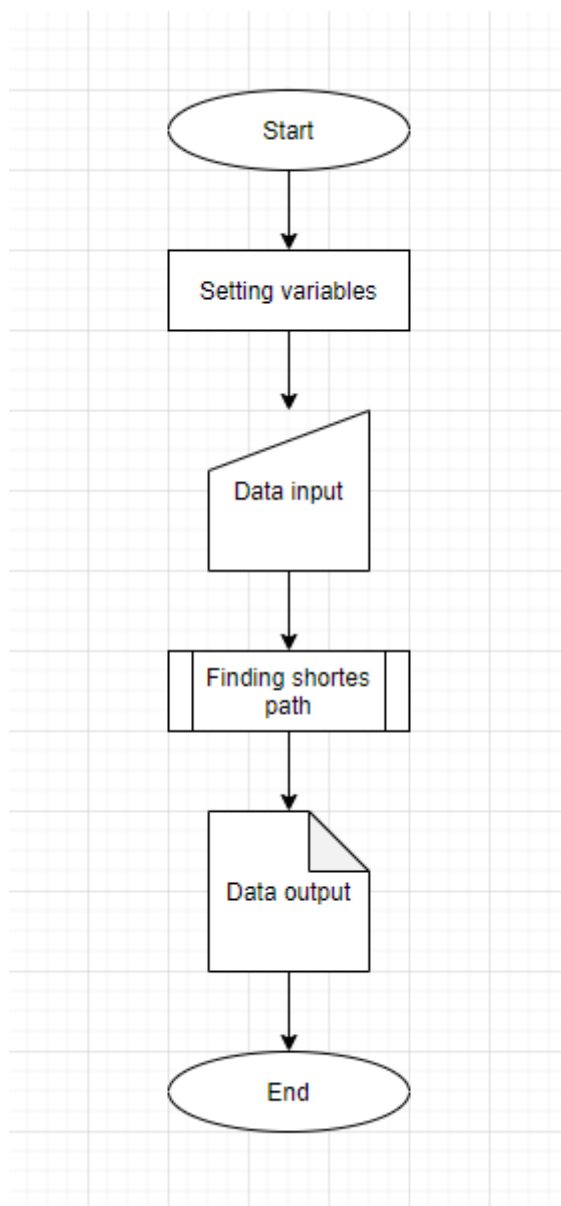
Problem komiwojażera (*Travelling Salesman Problem*) polega na znalezieniu minimalnego cyklu Hamiltona (przejście przez wszystkie wierzchołki tylko raz, startując i kończąc w tym samym punkcie) w pełnym grafie ważonym.

2. Opis metody

Metoda programowania dynamicznego (eng. *dynamic programming*) polega na podzieleniu problemu na mniejsze części, a następnie rozwiązanie każdej z nich. Każdy następny etap wykorzystuje otrzymane wcześniej wyniki co pozwala skrócić czas obliczenia całości, bo nie ma potrzeby wykonywania kilkakrotnie tych samych obliczeń. Metoda jest trudniejsza do zaimplementowania i zazwyczaj limitem jest pamięć, a nie czas. W tym problemie sprowadza się ona do znalezienia wszystkich możliwych podścieżek, a następnie „rozbudowywaniu” ich o kolejne węzły, aż otrzymamy pełny cykl i wybierzemy najtańszy z nich.

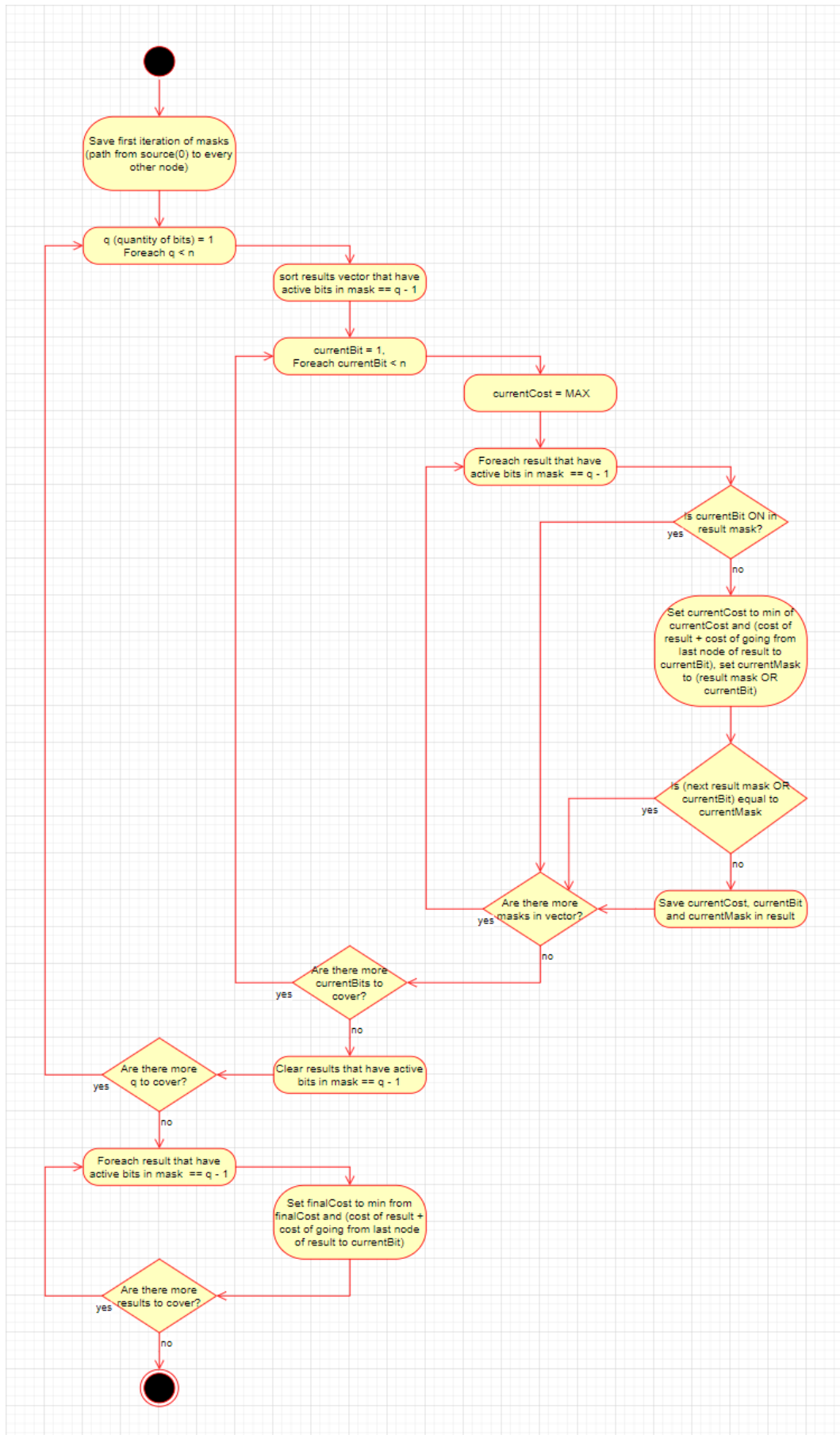
3. Opis algorytmu

Rozwiązanie zaimplementowano w postaci programu opisanego przez poniższe diagramy:



Rysunek 1: Ogólny diagram czynności programu

Najpierw inicjalizowane są zmienne, najlepsza ścieżka jest ustawiona na maksymalną wartość, maksymalna maska jest zerowana (później otrzyma wartość 2^n-1). Po wczytaniu danych z konsoli następuje uruchomienie właściwej części algorytmu, następnie wypisywany jest wynik oraz czas wykonania właściwego algorytmu.



Rysunek 2: Szczegółowy diagram czynności algorytmu

4. Dane testowe

Dane, na których była badana efektywność algorytmu pochodzą ze zbioru udostępnionego przez dr Rudego. Do badania użyto wartości z następujących plików:

- m15.astp
- gr17.tsp
- gr21.tsp
- ulysses22.stp
- gr24.stp
- fri26.stp
- bayg29.stp

<http://jaroslaw.rudy.staff.iiar.pwr.wroc.pl/files/pea/instances.zip>

5. Procedura badawcza

Należało zbadać zależność czasu rozwiązania problemu od wielkości instancji. W przypadku algorytmu realizującego przegląd zupełny przestrzeni rozwiązań dopuszczalnych nie występowały parametry programu, które mogły mieć wpływ na czas i jakość uzyskanego wyniku. W związku z tym procedura badawcza polegała na uruchomieniu programu i wklejeniu do niego danych z plików wybranych do badania.

Każda z instancji została wykonana do 10 razy, aby uśrednić czasy, limitem okazało się $n=29$, mimo usuwania poprzednich wyników zajęta pamięć (według Visual Studio 2019) przekroczyła 22GB, co poskutkowało memory overflow i niepowodzeniem testu. Wyniki były zapisywane w Excelu, następnie na ich podstawie została przeprowadzona analiza.

Pomiar czasu został zbadany przy użyciu biblioteki chrono. Po otrzymaniu wyniku należy go podzielić przez liczbę powtórzeń wywołań algorytmu.

```
auto startTime = chrono::steady_clock::now();

for (int q = 0; q < repeats; ++q)
{
    cout << "Iteration " << q << endl;
    shortest_path = INT_MAX;
    maxMask = 1;
    findShortestPath();
}

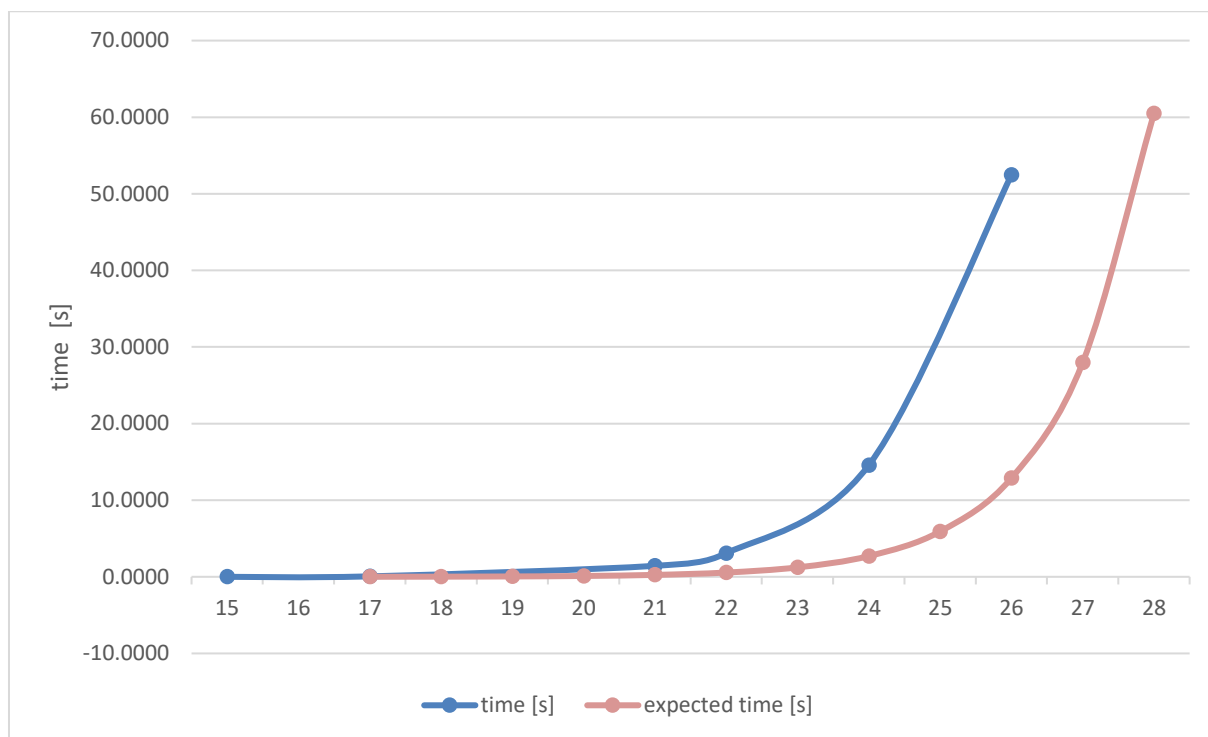
auto resultTime = chrono::steady_clock::now() - startTime;

cout << shortest_path << " - expected: " << result << endl;
cout << chrono::duration <double, milli>(resultTime).count() << "ms \n";
```

Item	Value
OS Name	Microsoft Windows 11 [REDACTED]
Version	10.0.22471 Build 22471
Other OS Description	Not Available
OS Manufacturer	Microsoft Corporation
System Name	SUPERCIUPERPC
System Manufacturer	Micro-Star International Co., Ltd.
System Model	MS-7C95
System Type	x64-based PC
System SKU	To be filled by O.E.M.
Processor	AMD Ryzen 5 3600 6-Core Processor, 3600 Mhz, 6 Core(s), 12 Logical Proces...
BIOS Version/Date	American Megatrends International, LLC. 2.82, 22/06/2021
SMBIOS Version	2.8
Embedded Controller Version	255.255
BIOS Mode	UEFI
BaseBoard Manufacturer	Micro-Star International Co., Ltd.
BaseBoard Product	B550M PRO-VDH (MS-7C95)
BaseBoard Version	1.0
Platform Role	Desktop
Secure Boot State	Off
PCR7 Configuration	[REDACTED]
Windows Directory	[REDACTED]
System Directory	[REDACTED]
Boot Device	[REDACTED]
Locale	United Kingdom
Hardware Abstraction Layer	[REDACTED]
Username	[REDACTED]
Time Zone	[REDACTED]
Installed Physical Memory (RAM)	16.0 GB
Total Physical Memory	15.9 GB
Available Physical Memory	8.21 GB
Total Virtual Memory	18.3 GB
Available Virtual Memory	6.31 GB
Page File Space	2.38 GB
Page File	[REDACTED]
Kernel DMA Protection	Off
Virtualisation-based security	Not enabled
Device Encryption Support	Elevation Required to View
Hyper-V - VM Monitor Mode E...	Yes
Hyper-V - Second Level Addres...	Yes
Hyper-V - Virtualisation Enable...	Yes
Hyper-V - Data Execution Prote...	Yes

6. Wyniki

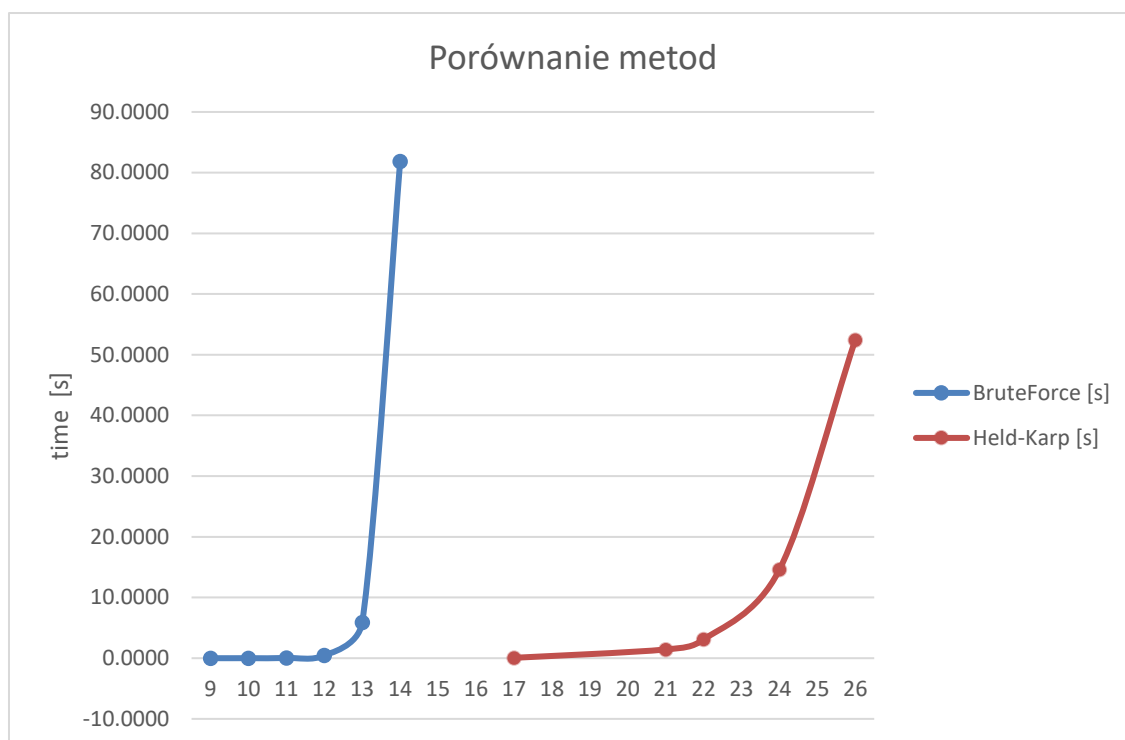
Wyniki opracowano w programie Excel:



Rysunek 3: Czasy działania algorytmu dla n wierzchołków - graf

	15	16	17	18	19	20	21	22	23	24	25	26	27	28
time [s]	0.001		0.061				1.436	3.083		14.57		52.44		
expected time [s]			0.009	0.022	0.050	0.113	0.252	0.559	1.233	2.706	5.916	12.88	27.96	60.48

Rysunek 4: Czasy działania algorytmu dla n wierzchołków - tabela



Rysunek 5: Porównanie czasu działania dotychczas opracowanych algorytmów dla n wierzchołków - graf

*dane testowe nie są kolejnymi wartościami n stąd połączenie znanych wyników linią

7. Analiza

Krzywa wzrostu czasu względem wielkości instancji ma charakter wykładniczy (rysunek 1). Nałożenie krzywej expected time ($f(n) = 2^n * n^2$) potwierdza, że badany algorytm wyznacza rozwiązania problemu komiwojażera dla badanych instancji w czasie $2^n * n^2$ zależnym od wielkości instancji (obie krzywe są zgodne co do kształtu). Złożoność czasowa opracowanego algorytmu wynosi $O(2^n * n^2)$. Warto zwrócić uwagę, że nawet wykorzystując dynamiczne struktury danych złożoność pamięciowa skutecznie ogranicza użyteczność tego algorytmu.

Porównanie dotychczas zrealizowanych algorytmów pokazuje jak dużą przewagę ma podejście dynamiczne nad siłowym. Czas wykonania Helda-Karpa zauważalnie niższy jednak jest to okupione wysokim zużyciem pamięci.