Dokumentacja końcowa do projektu

Analiza algorytmów

Treść zadania

10. Listonosz

Dany jest graf nieskierowany $G(V, E)$, podzbiór krawędzi $F \subset E$. Odnaleźć możliwie najkrótszy cykl w grafie, który zawiera każdą krawędź co najmniej raz.

Należy porównać rozwiązanie dokładne z heurystyką.

Metody rozwiązania

Algorytm dokładny:

1. Sprawdzenie spójności grafu wykorzystując algorytm przeszukiwania w głąb.
2. Wyszukanie w grafie wierzchołków o nieparzystych stopniach.
3. Wyznaczenie najkrótszej ścieżki łączącej ze sobą wierzchołki o nieparzystym stopniu wykorzystując algorytm przeszukiwania wszerz.
4. Wyszukanie skojarzenia tychże wierzchołków w pary tak, aby połączenia między parami były jak najkrótsze.
5. Zdublowanie krawędzi wchodzących w skład wyznaczonych w poprzednim punkcie ścieżek.
6. Wyznaczenie w grafie cyklu Eulera wykorzystując algorytm przeszukiwania w głąb.

Algorytm heurystyczny:

1. Wybranie dowolnego wierzchołka i oznaczenie go jako wierzchołek startowy.
2. Dodanie wierzchołka do cyklu i sprawdzenie jego sąsiadów.
3. Sprawdzenie, czy istnieje nieodwiedzona krawędź do któregoś z sąsiadów, w pierwszej kolejności wybierając wierzchołki inne niż startowy.
4. Jeśli tak to ustawienie wierzchołka poprzedniego dla niego (jeśli taki nie został jeszcze ustawiony lub jeśli sprawdzany wierzchołek był startowy), przejście do niego oraz powrót do punktu 2.
5. Jeśli sprawdzanym wierzchołkiem nie jest wierzchołek startowy, to skok do punktu 8.
6. Jeśli odwiedzono wszystkie krawędzie cyklu to koniec.
7. Jeśli nie, to sprawdzenie od początku cyklu, który z wierzchołków posiada krawędzie jeszcze nieodwiedzone, następnie przejście do takiego wierzchołka, oznaczenie go jako wierzchołek startowy i powrót do punktu 2.
8. Sprawdzenie, czy w wyniku ustawiania wierzchołków poprzednich powstała pętla.
9. Jeśli nie to dodanie nowej krawędzi pomiędzy sprawdzanym wierzchołkiem, a wierzchołkiem poprzednim, przejście do wierzchołka poprzedniego i powrót do punktu 2.
10. Wyszukanie w powstałej pętli pierwszego wierzchołka, który kiedyś był startowym i dodanie najkrótszej prowadzącej do niego sekwencji do cyklu.
11. Wyznaczenie najkrótszej ścieżki od wyszukanego wierzchołka do wierzchołka startowego, wykorzystując listę wierzchołków budowanego cyklu..
12. Dodanie wyznaczonej poprzednio ścieżki do cyklu, przejście do wierzchołka startowego i powrót do punktu 2.

Ideą algorytmu jest na stworzeniu małego cyklu w grafie i rozbudowywaniu go o kolejne podcykle tak, aby przejść po wszystkich krawędziach. Podczas przechodzenia po grafie zapamiętywane są wierzchołki startowe, czyli takie, od których zaczął się cykl lub podcykl i do których trzeba wrócić, bo go zamknąć. Ponadto ścieżki pomiędzy tym samym wierzchołkiem startowym są budowane tak, aby były jak najdłuższe. Aby możliwy był powrót do wierzchołka startowego, zapamiętywane są wartości wierzchołków poprzednich podczas przechodzenia po grafie. Dodatkowo algorytm eliminuje przypadki, kiedy podczas przechodzenia po wierzchołkach poprzednich napotkana zostanie pętla, co jest szczególnie częste w grafach "rzadkich".

Przewidywana złożoność

Dla rozwiązania dokładnego, zgodnie z dostępną literatura przyjęto złożoność $O\big(T(n)\big) = n^3$, gdzie n to ilość wierzchołków w grafie. Z kolei w przypadku rozwiązania przybliżonego można się spodziewać złożoności zbliżonej do liniowej, jednak przyjęto $O\big(T(n)\big) = \frac{n}{2}$, gdzie $n = \frac{2e}{v} * (v-1)$, v to ilość wierzchołków, a e to ilość krawędzi w grafie.

Tabele pomocnicze przy oszacowaniu złożoności algorytmu dokładnego, kolejno od lewej wartości prawdopodobieństwa to 0.8, 0.5 oraz 0.2.

n	t(n)	q(n)
200	0,301	1,05
220	0,403	1,055
240	0,715	1,442
260	0,553	0,876
280	0,731	0,928
300	0,969	1
320	1,267	1,078
340	1,566	1,11
360	1,987	1,187
380	2,571	1,305
400	3,118	1,358

n	t(n)	q(n)
200	0,179	0,811
220	0,244	0,832
240	0,297	0,779
260	0,423	0,872
280	0,55	0,909
300	0,744	1
320	0,953	1,055
340	1,255	1,159
360	1,541	1,198
380	2,026	1,339
400	2,452	1,389

n	t(n)	q(n)
200	0,073	1,022
220	0,086	0,905
240	0,112	0,904
260	0,154	0,98
280	0,198	1,006
300	0,242	1
320	0,298	1,013
340	0,409	1,161
360	0,486	1,162
380	0,624	1,268
400	0,696	1,213

Tabele pomocnicze przy oszacowaniu złożoności algorytmu heurystycznego, kolejno od lewej wartości prawdopodobieństwa to 0.8, 0.5 oraz 0.2.

n	t(n)	q(n)
200	0,02	1,065
220	0,018	0,796
240	0,021	0,783
260	0,027	0,852
280	0,034	0,918
300	0,042	1
320	0,054	1,127
340	0,064	1,178
360	0,095	1,147
380	0,09	1,321
400	0,108	1,429

n	t(n)	q(n)
200	0,005	0,685
220	0,007	0,79
240	0,008	0,768
260	0,01	0,832
280	0,014	0,946
300	0,016	1
320	0,019	1,033
340	0,024	1,127
360	0,028	1,195
380	0,035	1,302
400	0,045	1,536

n	t(n)	q(n)
200	0	0,841
220	0,001	0,84
240	0,001	0,948
260	0,001	0,913
280	0,001	0,967
300	0,001	1
320	0,002	1,018
340	0,002	1,146
360	0,002	1,133
380	0,002	1,165
400	0,003	1,186

Pomiary czasu wykonania i wnioski

Algorytm dokładny znajduje rozwiązanie o wiele wolnej wraz ze wzrostem n, z kolei algorytm heurystyczny jest zdecydowanie szybszy, jednak nie zapewnia idealnego rozwiązania. Otrzymane odchylenia są zależne gęstości grafu oraz jego rozmiaru. Dla rzadkich grafów jest to nawet 25% w przypadku grafów małych lub do 5% w przypadku grafów dużych. Natomiast dla gęstych grafów algorytm działa dużo lepiej i pozwala na uzyskanie odchylenia kilku procentowego dla grafów małych i mniej niż 1% dla grafów dużych.

Tabela z czasami wykonania dla przykładowych wartości parametrów

n	p-stwo	a. dokładny	a. heurystyczny
200	0.1	0.541	0.01
200	0.9	0.921	0.145
400	0.1	2.006	0.011
400	0.5	8.811	0.176
400	0.9	15.451	0.496
800	0.1	31.171	0.235
800	0.5	119.491	0.666
800	0.9	235.154	4.466
1600	0.1	464.802	0.876
1600	0.5	654.429	4.101
1600	0.9		14.596
3200	0.1		16.216
3200	0.5		37.326
3200	0.9		153.027