Autorzy:

Druszcz Bartłomiej

Polaczek Jakub

Grafy i Sieci

Wyznaczanie najszybszego przepływu

Sprawozdanie 2 – proponowane rozwiązanie problemu

Treść zadania:

Sieć przepływowa ma jedno źródło i jedno ujście. Każdy łuk sieci ma określone dwa parametry całkowitoliczbowe: przepustowość oraz czas przepływu. Problem najszybszego przepływu polega na wyznaczeniu najszybszego sposobu przesłania ze źródła do ujścia zadanej wielkości towaru nie przekraczając przepustowości łuków. W ramach projektu należy opracować algorytm i program dla problemu najszybszego przepływu.

Rozważania wstępne:

W rozważanym grafie skierowanym G = (V,E) każdy łuk sieci posiada parametry e = (c, τ) – przepustowość oraz czas przepływu (opóźnienie). Wyróżnione są również dwa węzły sieci: źródło s oraz ujście t oraz objętość przepływu którą należy przetransportować przez sieć – F. Celem opracowanego algorytmu jest minimalizacja globalnego czasu T niezbędnego do przesłania F jednostek przepływu z źródła do ujścia.

Problem najszybszego przepływu jest blisko spokrewniony z klasycznym problemem maksymalnego przepływu, przedstawionym przez Forda-Fulkersona [4]. Algorytm Forda-Fulkersona pozwala wyznaczyć maksymalną przepustowość w sieci przepływowej. Generalizacja tego algorytmu, pozwalająca na uwzględnienie czasu i rozszerzenie zagadnienia o wymiar czasowy i dodanie parametrów reprezentujących opóźnienie przepływu w łuku nazywane jest problemem maksymalnego przepływu dynamicznego [1]. Rozwiązując problem maksymalnego przepływu dynamicznego iteracyjnie modyfikując horyzont czasowy przepływu T metodą bisekcji szukamy takiego czasu Tx, dla którego maksymalny przepływ wynosi F.

Tradycyjnie stosowanym rozwiązaniem problemu maksymalnego przepływu dynamicznego, będącego częścią rozwiązania zadania najszybszego przepływu, jest zastosowanie algorytmu Forda-Fulkersona do rozwiązywania przepływu w tzw. sieci rozszerzonej czasowo (ang. Time expanded network) zawierającej kopię przepływu i sieci dla każdego dyskretnego kroku czasowego. Jest to rozwiązanie dość kosztowne pamięciowo oraz obliczeniowo w porównaniu z najnowszymi metodami opierającymi się na wielu optymalizacjach tej metody m-in parametrycznym przeszukiwaniu grafu przy znajdowaniu ścieżek powiększających [2], czy zgrubnej dyskretyzacji czasowej [1]. Wszystkie te rozwiązania wychodzą jednak z tej samej metody i to właśnie to klasyczne podejście zastosujemy do rozwiązania zadania.

Maksymalny przepływ dynamiczny:

Rozwiązanie maksymalnego przepływu dynamicznego opiera się na iteracyjnym rozwiązywaniu statycznego przepływu algorytmem Forda-Fulkersona. Algorytm ten działa w opierając się o następujące mechanizmy:

- Tworzona jest tzw. Sieć residualna **Gf(V,Ef)** będąca początkowo kopią sieci przepływowej, w czasie działania algorytmu zmieniają się jednak przepustowości łuków tej sieci zgodnie ze wzorem $cf(u,v) = c(u,v) = f(u,v)$, gdzie f(u,v) jest przepływem przez dany łuk w sieci.
- Definiujemy ścieżkę powiększającą jako dowolną ścieżkę p ze źródła s do ujścia t w sieci residualnej i obliczamy związany z nią parametr $cf(p)$ będący minimalną wartością przepustowości krawędzi w danej ścieżce. Ścieżka może być wyznaczona przy pomocy przeszukania w głąb / wszerz (Algorytm Edmundsa-Karpa).

Pseudokod metody Forda-Fulkersona dla statycznego przepływu prezentuje się następująco:

WHILE istnieje ścieżka powiększająca p należąca do grafu powiększającego:

Dla każdej krawędzi (u,v) należącej do p:

$f(u,v) := f(u,v) + cf(p)$

$f(v,u) := f(v,u) - cf(p)$

Praktyczna implementacja tego algorytmu uwzględnia przeszukiwanie sieci wszerz w celu znalezienia ścieżki o najmniejszej liczbie krawędzi nazywana jest algorytmem Edmundsa-Karpa i będzie zaimplementowana w projekcie. Złożoność obliczeniowa tego kroku wynosi O(VE^2), gdzie E jest liczbą krawędzi grafu, V ilością krawędzi.

Rozszerzenie tego algorytmu do problemu maksymalnego przepływu dynamicznego [3] o horyzoncie czasowym T zaczynamy od zdefiniowania sieci rozszerzonej czasowo **D(t).** Wierzchołki P_i^t(i =0 ,n; t=0,T) są wierzchołkami sieci **D(t)**. Krawędzie budujemy z łuków o nieskończonej przepustowości $P_i^t P_i^{t+1}(0 \le t \le T-1)$ oraz dla $i \neq j$ $P_i^t P_i^{t+t_{ij}}(0 \le t \le T - t_{ij})$. Utworzenie takiej rozszerzono czasowo sieci umożliwia sprowadzenie zagadnienia do maksymalnego przepływu statycznego. Taka sieć modeluje zarówno czas przepływu między węzłami sieci jak i pojemność danego węzła. Złożoność czasowa budowy takiej sieci wynosi O(VT + ET) , gdzie T to ilość kroków czasowych (horyzont) [5].

Kolejny krok algorytmu to opisany wcześniej algorytm Edmundsa-Karpa. W przypadku rozszerzonej czasowo sieci o horyzoncie T jego złożoność wyniesie O(VT(ET)^2).

Ostatnim krokiem jest rekonstrukcja sieci czasowo rozszerzonej do przepływu dynamicznego. W przypadku najszybszego przepływu interesuje nas parametr określający ilość przepływu który dotarł do ujścia w analizowanym czasie **Q**.

Najszybszy przepływ:

Główna pętla algorytmu opiera się na uruchamianiu algorytmu wyszczególnionego w poprzednim paragrafie dla zmienianego horyzontu czasowego T i wyszukiwaniu binarnym takiego Tx, Dla którego **Q=F** [1]. Otrzymany najszybszy przepływ należy następnie

zrekonstruować oraz zapisać w wyjściu z programu. Złożoność wyszukiwania binarnego wynosi $O(\log_2 n)$, a pojedynczej iteracji O(VT+ET + VT(ET)^2), zatem całego algorytmu powinna mieścić się w granicach $O(\log_2(VT+ET + VT(ET)^2))$

Wejście i wyjście programu:

Dane wejściowe zapisane będą w postaci listy w pliku tekstowym.

Format zapisu:

s, t, F, liczba_lukow, liczba_wierzcholkow
Vi, Vj, p, c
...

Gdzie:

Vi, Vj - wierzchołki grafu połączone łukiem
s - indeks wierzchołka źródłowego
t - indeks wierzchołka końcowego
F - ilość jednostek przepływu do przetransportowania
p - przepustowość łuku
c - czas przepływu przez łuk

Dane wyjściowe przedstawione będą w formie mapy obciążenia łuków w czasie. W kolejnych kolumnach znajduje się obciążenie jednostkowe dla jednego łuku w mapie, każdy wiersz przedstawia inny moment w czasie.

Format wyjściowy:

czas	(s,Vi)	(Vi,Vj)	…	(Vk.t)
0	0	0	...	0
1	pi	0	...	0
2	2pi	0	…	0
3	2pi	pj	…	0
		.		
		.		
n	0	0	...	pk

Szczegóły implementacji:

Program opracowany zostanie w języku C++ w środowisku Windows/Linux. Do obsługi technicznej strony reprezentacji grafu użyta zostanie biblioteka LEMON pozwalająca na wydajną obsługę sieci tego typu. Biblioteka ta zawiera również podstawowe algorytmy grafowe, co może być przydatne przy opracowywaniu rozwiązania.

Testowanie wydajnościowe algorytmu:

Poprawność działania algorytmu weryfikowana będzie na dwa sposoby. Pierwszy uwzględnia ręczne rozwiązanie zadania dla trywialnych przypadków oraz porównanie wyników uzyskanych przez algorytm. Druga metoda zakłada stworzenie losowych, większych sieci oraz porównanie rozwiązania z algorytmem przepływu o minimalnym koszcie zaimplementowanym w ramach używanej biblioteki. Zgodnie z założeniami zagadnienia, czas uzyskany przez algorytm najszybszego przepływu powinien być taki sam lub krótszy od algorytmu najmniejszego kosztu.

Do testów działania algorytmu został napisany specjalny program w języku C++. Program ten generuje graf o losowych parametrach. Zdefiniowano w nim następujące parametry:

n – liczba wierzchołków,
m – liczba krawędzi skierowanych,
pmax – maksymalna przepustowość krawędzi,
tmax – maksymalny czas przepływu przez krawędź,
F – zadana objętość do przetransferowania między źródłem a ujściem,
s – numer wierzchołka początkowego (źródło),
t - numer wierzchołka końcowego (ujście).

W programie zdefiniowano trzy tablice reprezentujące:
- macierz sąsiedztwa grafu,
- przepustowości krawędzi,
- czas przepływu przez krawędź.

- Generacja macierzy sąsiedztwa:

Założeniem projektu jest, że z dowolnego wierzchołka można przetransportować określoną objętość F do innego wierzchołka w tym grafie. Aby ten warunek był spełniony zawsze wszystkie wierzchołki muszą tworzyć cykl. Przy zastosowaniu cyku wykorzystano n krawędzi. Pozostałe m-n krawędzi rozmieszczono losowo w grafie uwzględniając fakt, że na istniejącej krawędzi nie można już postawić innej krawędzi skierowanej.

- Generacja przepustowości krawędzi i czasu przepływu przez krawędź:

Dla wszystkich istniejących krawędzi wygenerowano losowe wartości z przedziału od 1 do odpowiednio pmax i tmax.

Tak wygenerowany graf testowy został zapisany w pliku tekstowym w formacie opisanym w podpunkcie *Wejście wyjście programu*. Kolejne kolumny będą oddzielone tabulatorami.

Cytowane prace

[1] L. Fleischer and M. Skutalla, “Quickest flows over time,” *Society for Industrial and Applied Mathematics,* pp. 1600-1630, 2007.

[2] R. E. Burkard, K. Dlaska i B. Klinz, „The quickest flow problem,” *ZOR Methods Models,* p. 31–58., 1993.

[3] L. R. Ford i D. Fulkerson, „Constructing maximal dynamic flows from static flows,” The Rand Corporation, Santa Monica, California, 1958.

[4] L. R. Ford i D. Fulkerson, Flows in Networks, Santa Monica, California: Rand corporation, 1962.

[5] M. A. Fonoberova i D. D. Lozovanu, „The maximum flow in dynamic networks,” *Computer Science Journal of Moldova,* nr 387,396, 2004.