

Autorzy:

Druszcz Bartłomiej

Polaczek Jakub

Grafy i Sieci

Najszybszy przepływ

Sprawozdanie 1 - założenia wstępne

Opis zagadnienia:

Zagadnienie wyznaczenia najszybszego przepływu jest problemem optymalizacyjnym często spotykanym w transporcie, planowaniu ewakuacji, telekomunikacji czy tworzeniu harmonogramów. W tym projekcie opracowywane będzie wyznaczanie najszybszego przepływu dla grafu skierowanego $G = (V, E)$, gdzie każdy łuk sieci ma przypisaną stałą w czasie przepustowość oraz koszt czasowy transportu. Poprzez przepustowość łuku V rozumie się maksymalną liczbę jednostek przepływu jaką można przesłać w podstawowej jednostce czasu, a koszt czasowy to opóźnienie po którym jednostka dotrze do docelowego wierzchołka. Przepływ obejmuje jedno źródło (s) oraz jedno ujście t . Zadaniem opracowywanego algorytmu jest wyznaczenie takiej sekwencji połączeń pomiędzy źródłem, a ujściem pozwalającej na przetransportowanie pewnej objętości F w jak najkrótszym czasie.

Wejście i wyjście programu:

Dane wejściowe zapisane będą w postaci listy w pliku tekstowym.

Format zapisu:

$s, t, F, \text{liczba_lukow}, \text{liczba_wierzchołkow}$

V_i, V_j, p, c

...

Gdzie:

V_i, V_j - 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, V_i)	(V_i, V_j)	(V_k, t)
0	0	0	0
1	p_i	0	0
2	p_i	p_j	0
	...		
N	0	0	p_k

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.

Sposób testowania:

Poprawność działania algorytmu weryfikowana będzie na dwa sposoby. Pierwszy uwzględni 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.