Wydział Zastosowań Informatyki i Matematyki,   
SGGW, maj 2018

Informatyka, semestr 4  
Grafy i sieci

Projekt 13  
Zuzanna Gałecka  
Bartosz Matyjasiak  
Maciej Zwoliński

Zadanie: W oparciu o metodę poszukiwania maksymalnego przepływu w sieci opracować i zaimplementować algorytm najszybszego przepływu gazu w sieci pomiędzy jej dwoma punktami.

Opis problemu maksymalnego przepływu:  
Problem maksymalnego przepływu to zagadnienie często spotykane w informatyce. Polega ono na znalezieniu dla danej sieci przepływowej takiego przepływu **f**, którego wartość jest maksymalna, gdzie wartość przepływu jest zdefiniowana jako łączny przepływ opuszczający źródło.   
Bardziej formalnie, dla danego przepływu **f**, w sieci **G = ( V , E )**, o źródle **s** i ujściu **t**, jego wartość jest zdefiniowana następująco:

Istnieje też uogólnienie tego problemu, w którym sieć **G** zawiera wiele źródeł i ujść.  
Innymi słowy , dodajemy do sieci **G** wierzchołek **s** połączony krawędziami o nieskończonej przepustowości ze wszystkimi źródłami oraz wierzchołek **t** połączony krawędziami o nieskończonej przepustowości ze wszystkimi ujściami. Wierzchołek **s** zwany jest **superźródłem**, zaś wierzchołek **t** - **superujściem**.

Jednym z algorytmów pozwalających rozwiązać problem maksymalnego przepływu jest algorytm Edmondsa-Karpa, na podstawie którego powstał nasz program.

Algorytm Edmondsa-Karpa jest jedną z realizacji metody Forda-Fulkersona rozwiązywania problemu maksymalnego przepływu w sieci przepływowej. Jego złożoność czasowa wynosi O(VE2), jest zatem wolniejszy od innych znanych algorytmów przepływowych działających w czasie O(VE3), takich jak algorytm relabel-to-front, czy algorytm trzech Hindusów. W praktyce jednak złożoność pesymistyczna rzadko jest osiągana, co w połączeniu z prostotą czyni algorytm Edmondsa-Karpa bardzo użytecznym, szczególnie dla grafów rzadkich.

Algorytm ten został odkryty przez rosyjskiego naukowca, E. A. Dinica w roku 1970, i niezależnie przez Jacka Edmondsa i Richarda Karpa w roku 1972. Artykuł Dinica zawiera dodatkowe techniki, które obniżają czas działania do O(V2E), (algorytm z tą poprawką nazywa się obecnie algorytmem Dynica).

Idea algorytmu jest identyczna z ideą metody Forda-Fulkersona, z dodatkowym warunkiem: ścieżka powiększająca, którą szukamy w każdym kroku algorytmu, musi być najkrótsza, czyli zawierać minimalną możliwą liczbę (nie wagę!) krawędzi. Taką ścieżkę znajduje się uruchamiając algorytm przeszukiwania grafu wszerz w sieci residualnej.

Poprawność algorytmu wynika wprost z twierdzenia Forda-Fulkersona: po zakończeniu działania w grafie nie może być ścieżki powiększającej, przepływ jest więc maksymalny. Dowód opiera się on na fakcie, że długość ścieżki powiększającej nie może maleć, a utrzymywać się na tym samym poziomie może przez co najwyżej **O(E)** kroków algorytmu (czyli jest co najwyżej **O(VE)** kroków, jako że długość ścieżki nie przekroczy **V**).

[źródło: wikipedia]

* Opis rozwiązania (kodu)?
* Zastosowania
* Wykres złożoności dla 3 wybranych przykładów (izi, normal, hard lvl)
* Zdanie podsumowania porównania złożoności z wykresu, do tej z teorii
* Przeredagowanie żeby wyglądało ładniej