
Instrukcja obsługi algorytmu Forda-Fulkersona
Testowanie i weryfikacja oprogramowania 2012/2013 — Projekt

Data	2012-12-18
Wersja	1.0
Autorzy	MO, RW, TC, MM

1 Wstęp merytoryczny

Opracowywany w ramach projektu algorytm Forda-Fulkersona pozwala na wyznaczenie sieci przepływowej o maksymalnym przepływie. Algorytm operuje na grafach spójnych, skierowanych, których krawędzie posiadają przypisane dwie wartości *przepływ* i *przepustowość*. Dodatkowo, algorytm jest kompletny, gdy graf, na którym operuje, spełnia trzy warunki:

1. *Ograniczenie przepustowości* — Dla każdej krawędzi *przepływ* ma wartość nieujemną i nie przekracza wartości *przepustowości* tej krawędzi.
2. *Zachowanie przepływu* — Dla każdego wierzchołka, suma przepływów wszystkich krawędzi wchodzących jest równa sumie przepływów wszystkich krawędzi wychodzących.
3. *Symetria przepływu krawędzi* — Dla każdej krawędzi wartość przepływu krawędzi jest równa wartości ujemnej dla tej krawędzi w przeciwnym zwrocie.

Definicje pojęć wykorzystanych w opisie znajdują się w paragrafie 4 na stronie 4.

Szczegółowy opis działania algorytmu znajduje się w paragrafie 5 na stronie 4.

2 Instrukcja obsługi

2.1 Dane wejściowe

Algorytm Forda-Fulkersona opisany powyżej przyjmuje dwa argumenty:

- FlowNetwork – graf przepływu sieci, posiadający dwie możliwe reprezentacje,
- Search – metoda wykorzystywana do wyszukiwania ścieżek powiększających.

2.1.1 Graf przepływu FlowNetworkAdjacencyArray

Korzystamy z reprezentacji grafu przepływu zrealizowanej z użyciem list powiązanych. Ten model grafu zaimplementowany jest w klasie FlowNetworkAdjacencyArray. Do każdego wierzchołka grafu przypisane są dwie listy krawędzi, przednich oraz tylnych. Korzystając z takiej struktury modelu grafu, otrzymujemy łatwo zrozumiałą implementację algorytmu. Jednak z powodu tej redundancji nasza implementacja wykazuje się niską wydajnością pamięciową. Niezalecane jest stosowanie tej reprezentacji w przypadku dużych, gęstych grafów.

Argumenty wejściowe konstruktora FlowNetworkAdjacencyArray:

- int – ilość wszystkich węzłów w grafie,
- int – indeks węzła początkowego,
- int – indeks węzła docelowego,
- Iterator<EdgesInfo> – lista krawędzi wraz z ich przepustowością.

2.1.2 Krawędzie EdgeInfo

Informacje o krawędziach grafu przechowywane są jako kolekcje instancji klasy EdgeInfo. Zawierają informacje o wierzchołkach, pomiędzy którymi dana krawędź się znajduje oraz wartość jej przepustowości.

Argumenty konstruktora EdgeInfo:

- int – indeks węzła startowego,
- int – indeks węzła końcowego,
- int – wartość przepustowa krawędzi.

2.1.3 Metoda wyszukiwania ścieżki DFS_SearchList

Dostępne są dwie, standardowe metody wyszukiwania ścieżki; metoda przeszukiwania wszerz i wzdłuż. Oryginalnie algorytm Forda-Fulkersona korzysta z wyszukiwania ścieżki metodą wzdłuż. Z tej metody korzystamy.

2.2 Dane wyjściowe

Wyniki wyznaczania maksymalnego przepływu są zapisywane wewnątrz struktury reprezentującej graf. Instancja FlowNetwork posiada metodę toString(), która wyświetla aktualny stan grafu. Wynik wywołania tej metody po wyznaczeniu maksymalnego przepływu jest zapisywany w formacie:

```
[indeks wezla startowego] -> [indeks wezla koncowego] przeplyw/przepustowosc
```

Przykładowy wydruk wywołania tej metody:

Przed:

```
[0] -> [1] 0/3 @ 0
[0] -> [2] 0/2 @ 0
[1] -> [3] 0/2 @ 0
[1] -> [4] 0/2 @ 0
[2] -> [4] 0/3 @ 0
[2] -> [3] 0/2 @ 0
[3] -> [5] 0/3 @ 0
[4] -> [5] 0/2 @ 0
```

Po:

```
[0] -> [1] 3/3 @ 0
[0] -> [2] 2/2 @ 0
[1] -> [3] 2/2 @ 0
[1] -> [4] 1/2 @ 0
[2] -> [4] 1/3 @ 0
[2] -> [3] 1/2 @ 0
[3] -> [5] 3/3 @ 0
[4] -> [5] 2/2 @ 0
```

3 Przykład użycia

W pierwszej kolejności ustalane są główne parametry grafu:

- liczba węzłów grafu (poza źródłem i ujściem),
- indeks węzła źródłowego,
- indeks węzła docelowego.

```
int numVertices = 6;
int srcIndex = 0;
int sinkIndex = 5;
```

Następnie definiowane są poszczególne krawędzie grafu wraz z ich przepustowością:

- indeks węzła początkowego krawędzi,
- indeks węzła końcowego krawędzi,
- maksymalna przepustowość krawędzi.

```
ArrayList preIterator = new ArrayList();
EdgeInfo edge1 = new EdgeInfo(0, 1, 3);
EdgeInfo edge2 = new EdgeInfo(1, 3, 2);
EdgeInfo edge3 = new EdgeInfo(3, 5, 3);
EdgeInfo edge4 = new EdgeInfo(1, 4, 2);
EdgeInfo edge5 = new EdgeInfo(0, 2, 2);
EdgeInfo edge6 = new EdgeInfo(2, 4, 3);
EdgeInfo edge7 = new EdgeInfo(4, 5, 2);
EdgeInfo edge8 = new EdgeInfo(2, 3, 2);
preIterator.add(edge1);
preIterator.add(edge2);
preIterator.add(edge3);
preIterator.add(edge4);
preIterator.add(edge5);
preIterator.add(edge6);
preIterator.add(edge7);
preIterator.add(edge8);

Iterator<EdgeInfo> edges = preIterator.iterator();
```

Tworzony jest obiekt reprezentujący cały graf przepływu.

```
FlowNetworkAdjacencyList network = new FlowNetworkAdjacencyList(numVertices, srcIndex, ↵
    sinkIndex, edges);
System.out.println(network.toString());
```

Wybierana jest funkcja wyszukiująca. Następnie inicjowany i wykonywany jest algorytm Forda-Fulkersona.

```
// algorytm wyszukiwania sciezki wdluz
DFS_SearchList search = new DFS_SearchList(network);

// algorytm Forda-Fulkersona
FordFulkerson fordFulkerson = new FordFulkerson(network, search);

// wywołanie algorytmu
fordFulkerson.compute();
```

Wyświetlenie wyznaczonego maksymalnego przepływu.

```
System.out.println("Wynik:");
System.out.println(network.toString());
```

4 Słownik pojęć

Pojęcie	Opis
Sieć przepływowa	Sieć reprezentująca przepływy pomiędzy węzłem źródłowym i docelowym, z uwzględnieniem węzłów pośrednich.
Graf przepływu	Graf skierowany stanowiący abstrakcję sieci przepływowej, wierzchołki w grafie, odpowiadają węzłom sieci.
Wierzchołek grafu przepływu	Wierzchołek w grafie przepływu reprezentuje węzeł w sieci przepływowej. Wierzchołki w grafie przepływu, z wyjątkiem wierzchołka źródłowego i docelowego, muszą spełniać warunek, że suma $f(u, v)$ wszystkich krawędzi (u, v) we wpływie do wierzchołka u , musi być równa sumie $f(u, w)$ wszystkich krawędzi w wypływie z wierzchołka u . Co oznacza, iż żaden wierzchołek, poza wierzchołkami źródłowym i docelowym, nie mogą produkować ani konsumować przepływu.
Krawędź grafu przepływu	Krawędź w grafie przepływu łączy dwa wierzchołki. Wszystkie krawędzie w grafie przepływu są krawędziami skierowanymi.
Przepustowość krawędzi	Przepustowość krawędzi wyraża ograniczenie, co do maksymalnej liczby jednostek, które mogą tą krawędzią przepłynąć.
Przepływ krawędzi	Przepływ krawędzi definiuje liczbę jednostek przepływających z u do v (z punktu początkowego krawędzi do jej punktu końcowego).
Wierzchołek źródłowy	Wierzchołek źródłowy (źródło, ang. source) to wierzchołek wytwarzający jednostki towarów, które przepływają przez krawędzi grafu do wierzchołka docelowego. Przyjmuje się, że wierzchołek źródłowy jest w stanie wytworzyć dowolną wymaganą liczbę jednostek towarów, które zostaną z niego odebrane.
Wierzchołek docelowy	Wierzchołek docelowy (ujście, stacja końcowa, ang. target, terminus) to wierzchołek, który konsumuje otrzymane jednostki towarów dostarczone z wierzchołka źródłowego za pośrednictwem krawędzi grafu. Przyjmuje się, że wierzchołek źródłowy jest w stanie skonsumować dowolną liczbę jednostek towarów jakie zostaną do niego dostarczone.
Ścieżka	Ścieżka oznacza niecykliczną ścieżkę w grafie z niepowtarzalnymi wierzchołkami, prowadzącą z wierzchołka źródłowego do wierzchołka docelowego.
Ścieżka powiększająca	Ścieżka powiększająca (ang. augmenting path) to taka ścieżka, do której można dodać więcej przepływu. Co oznacza, że dla każdej krawędzi w ścieżce, przepływ jest mniejszy od jej przepustowości.

5 Analiza algorytmu

Algorytm Forda-Fulkersona analizuje znaną ścieżkę powiększającą, obliczając maksymalny przepływ dostępny na tej ścieżce. Znajduje najmniejszy z maksymalnych dostępnych przepływów spośród krawędzi na ścieżce.

Dla krawędzi przednich dostępny przepływ obliczamy jako różnicę przepustowości i wykorzystanego przepływu $(u, v) \cdot \text{przepustowość} - (u, v) \cdot \text{przepływ}$. Dla krawędzi tylnych dostępny przepływ jest równy wykorzystanemu przepływowi (do przodu) na tej krawędzi $(u, v) \cdot \text{przepływ}$. Przepływ tylny krawędzi jest pojęciem abstrakcyjnym na potrzeby prowadzonych obliczeń, a biorąc pod uwagę, że przepływ musi być nieujemny, do-

stępnym przepływie tylnym dla krawędzi jest równy wykorzystanemu przepływowi do przodu dla tej krawędzi.

Po przeanalizowaniu maksymalnej ścieżki powiększającej i obliczeniu maksymalnego przepływu dostępnego na tej ścieżce, aktualizujemy przepływy krawędzi. Dla krawędzi przednich, powiększamy przepływ krawędzi o obliczoną wartość $(u, v) \cdot \text{przepływ} = \text{delta}$. Dla krawędzi tylnych, pomniejszamy przepływ krawędzi o obliczoną wartość $(u, v) \cdot \text{przepływ} = \text{delta}$. Dopóki możliwe jest znalezienie ścieżki powiększającej, kontynuujemy obliczanie maksymalnego przepływu dostępnego na kolejnych ścieżkach i aktualizujemy przepływy na poszczególnych krawędziach tych ścieżek.

Algorytm zakończy swoją pracę w momencie, kiedy nie istnieją już żadne ścieżki powiększające, tj. zostanie wykorzystana maksymalna przepustowość sieci ze źródła do ujścia.

Historia dokumentu

Data	Wersja	Autor	Szczegóły
2012-12-12	0.1	MO	Utworzono wstępną wersję instrukcji.
2012-12-14	0.2	MM	Dodano rozdział <i>Przykład użycia</i> .
2012-12-16	0.3	RW	Dodano rozdział <i>Słownik pojęć</i> .
2012-12-16	0.4	RW	Dodano rozdział <i>Kroki algorytmu</i> .
2012-12-17	0.5	RW	Dodano skrócony opis algorytmu. Dodano opis algorytmu w języku naturalnym.
2012-12-17	0.5.1	MO	Poprawiono literówki i formatowanie.
2012-12-18	1.0	TC	Zatwierdzono.
2012-12-21	1.1	TC	Usunięto kroki algorytmu.
2012-12-21	1.1.1	TC	Poprawiono stylistykę i literówki.
2012-12-21	1.1.2	TC	Zmieniono formatowanie list i wydruki kodu.