

Teoria grafów - projekt

Część programistyczna

Zaimplementuj algorytm Bellmana-Forda

Przeanalizuj powyższy algorytm:
jakie problemy rozwiązuje,
konkretne przykłady wykorzystania,
z jakich metod korzysta się obecnie
do rozwiązywania tych problemów

Algorytm został zaimplementowany w języku C++
i znajduje się w repozytorium w folderze /proj.

W pliku Dokumentacja.pdf znajduje się dokumentacja
programu, zawierająca m. in. instrukcję do jego
poprawnego uruchomienia.

Ta sama dokumentacja jest zamieszczona w pliku
readme.txt w folderze /proj.

W tym dokumencie przedstawiona jest odpowiedź
na drugie polecenie – opisanie zastosowań algorytmu
Bellmana-Forda oraz problemów z nim związanych.

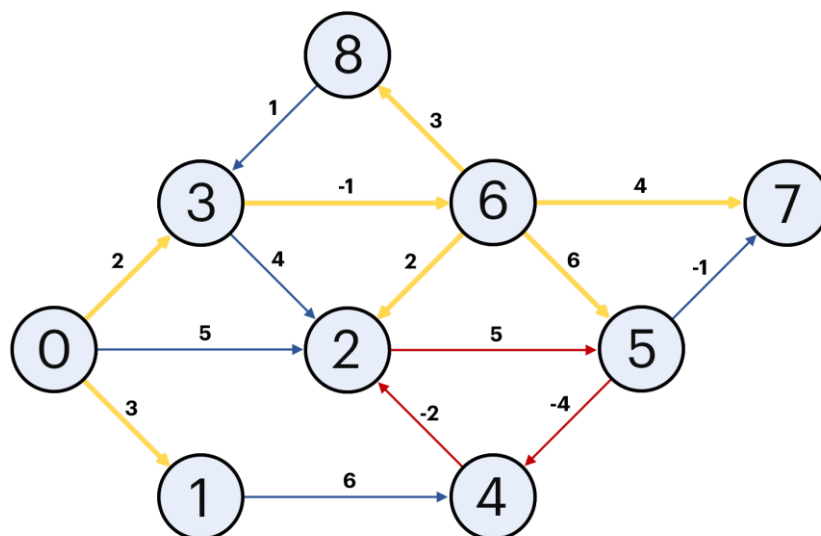
Algorytm Bellmana-Forda: jakie problemy rozwiązuje?

Algorytm Bellmana-Forda służy do znajdowania najkrótszych ścieżek z danego wężła startowego, w grafie skierowanym ważonym spójnym. Działa na zasadzie relaksacji – iteruje po grafie o jeden mniej razy niż liczba wężłów grafu. Z każdym krokiem zmniejsza koszty ścieżki od startu do każdego wężła, aż do osiągnięcia optymalnego rozwiązania.

Nie jest on najbardziej optymalny obliczeniowo – czas pracy wzrasta znacznie w grafach o wielu wężłach i krawędziach. W zamian za to może być wykorzystywany do pracy z grafami zawierającymi krawędzie o wagach ujemnych, w przeciwieństwie do innych algorytmów, np. algorytmu Dijkstry.

Wadą aplikacji algorytmu do grafów o wagach ujemnych jest fakt, że mogą w nich występować tzw. „cykle ujemne” – cykle zawierające krawędzie o wagach ujemnych, których suma jest także ujemna. Algorytm może wtedy redukować wagi ścieżek za jego pomocą w nieskończoność, przez co nie znajduje on poprawnego rozwiązania dla takiego grafu.

Jednocześnie, za pomocą małej modyfikacji, zamiast szukać i budować drzewo minimalnych ścieżek, algorytm może wykrywać obecność takich cykli, co również bywa jego odrębnym zastosowaniem.



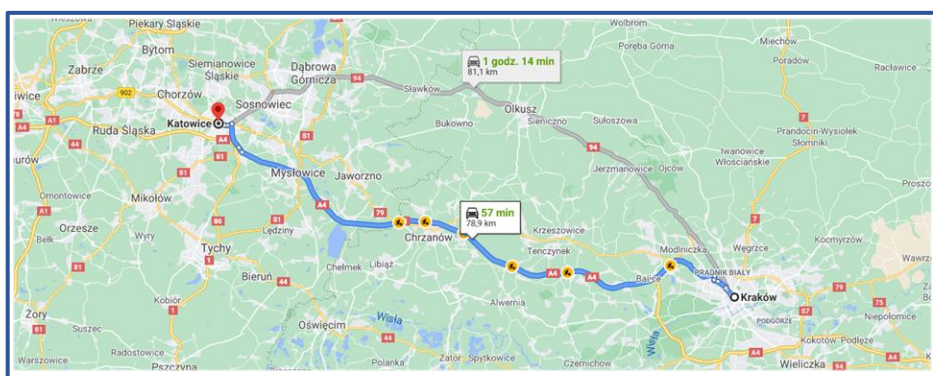
Algorytm Bellmana-Forda: konkretne przykłady wykorzystania

Podstawowym zastosowaniem algorytmu Bellmana-Forda jest znajdowanie najkrótszych ścieżek z (lub do) konkretnego wężła w grafie. Jego rozwiązania można więc wykorzystać przede wszystkim w problemie znajdowania najkrótszej drogi między dwoma punktami – przy planowaniu trasy przejazdu samochodem (również tym autonomicznym), lub przy przemieszczaniu się jednostek w środowiskach wirtualnych i symulacjach. Dodatkowo, z faktu że algorytm znajduje taką ścieżkę dla każdego osiągalnego wężła, można wspierać się nim przy opracowywaniu np. planu ewakuacji budynku.

Drzewo najkrótszych ścieżek można wykorzystać też w sytuacji, gdzie wagi krawędzi nie oznaczają dosłownie dystansu. Ścieżki znalezione przez algorytm mogą posłużyć podczas szukania najkorzystniejszego połączenia między jednostkami w sieci lub systemie. Dokładnie ten algorytm jest wykorzystywany w protokole IGRP stworzonym przez Cisco, służącym do kontroli przepływu pakietów w systemach autonomicznych.

Możliwość znajdowania ujemnych cykli jest równie pomocna. Inne algorytmy grafowe opierają poprawność swojego działania na braku takich cykli; stąd algorytm Bellmana-Forda może posłużyć do oceny, czy graf nadaje się do wykonania na nim danej operacji.

Do znajdowania cykli ujemnych można też sprowadzić istniejące problemy, takie jak wykrywanie arbitrażu walutowego (w uproszczeniu, możliwości zarobku na wymianie walut), lub wykrywanie niechcianych sprzężeń zwrotnych w projektach mikroobwodów.



Sieć dróg może zostać zamodelowana w postaci grafu ważonego skierowanego. // Źródło: Google Maps

Algorytm Bellmana-Forda:

Obecne metody rozwiązywania podobnych problemów

Pomimo swoich zalet, algorytm Bellmana-Forda nie jest rozwiązaniem uniwersalnym. Wiele problemów można rozwiązać sprawniej za pomocą innego narzędzia. Jest tak głównie za sprawą faktu iż algorytm ten słabo radzi sobie z modelami o wielu węzłach i krawędziach ze względu na swoją złożoność obliczeniową $O(|V| \cdot |E|)$.

Dokładne rozwiązania problemu znajdowania najkorzystniejszej trasy między punktami na mapie wykorzystywane w najpopularniejszych narzędziach (np. aplikacja Google Maps) nie są nigdy publikowane, z oczywistych względów. Najpewniej korzystają one jednak z algorytmów bardziej optymalnych, jak np. Dijkstry lub A^* , czyli jego ulepszonej wariacji. Nadają się one do tego m. in. ze względu na brak występowania wag ujemnych w tych modelach.

Dodatkowo, algorytmy szukające najkrótszej ścieżki wspiera się metodą Contraction Hierarchies, przyspieszającą ich działanie przez preprocessing danych przez przekazaniem ich do właściwego algorytmu.

Szukanie ścieżek w symulacjach prowadzone jest najczęściej za pomocą algorytmu A^* , lub na podstawie oddzielnych algorytmów multiagentowych.

Wśród protokołów trasowania w sieciach także znajdują się zastosowania dla algorytmu Dijkstry, np. w protokole OSPF, służącym do podobnego celu co wspomniany wcześniej protokół IGRP.

Wykrywanie ujemnych cykli również można prowadzić w lepszy sposób. Wiele prac naukowych poświęconych analizie algorytmów wykrywania takich cykli potwierdza, że algorytm Tarjana przynosi najlepsze efekty w ogólnym przypadku, z nielicznymi gdzie algorytm Goldberga-Radzika działa równie dobrze. Oba jednak wygrywają z algorytmem Bellmana-Forda w podstawowym formacie.

Pomimo pewnych wad, algorytm Bellmana-Forda jest popularnym i uznanym algorytmem grafowym. Jest tak nie tylko za sprawą możliwości wykorzystania go do modeli o ujemnych wagach, ale również dzięki jego stosunkowo prostemu i nieskomplikowanemu działaniu. Dzięki tym zaletom jest częstym wyborem do zastosowań amatorskich, jak również, jak w wypadku wspomnianego algorytmu Tarjana, bywa podstawą i częścią bardziej złożonych i zaawansowanych projektów i algorytmów.