Algorytmy i Złożoność

Wojciech Penczek

1 Znajdowanie nakrótszej ścieżki w grafie

Zastosowania: znalezienie najkrótszej trasy między dwoma miastami.
Dany graf $G = (V, E, M)$, w którym $M(v, v')$ przypisuje krawędziom wagi o wartościach rzeczywistych nieujemnych.
Wagą ścieżki w grafie nazywamy sumę wag jej krawędzi.
Waga najkrótszej ścieżki w grafie pomiędzy v i v' (ozn. $\delta(v, v')$) to minimum ze wszystkich wag ścieżek z v do v' lub ∞ jeśli taka ścieżka nie istnieje.
Algorytm BFS znajduje najkrótsze ścieżki w grafach z jednorodną wagą wszystkich krawędzi.
Rozważamy graf G z wyróżnionym wierzchołkiem s. Należy znaleźć najkrótszą ścieżkę z s do v dla każdego wierzchołka v.

Badamy podgraf poprzedników $G_\pi = (V_\pi, E_\pi, M_\pi)$, gdzie
$V_\pi = \{v \in V \mid \pi[v] \neq nil\} \cup \{s\}$,
$E_\pi = \{(\pi[v], v) \mid v \in V_\pi \setminus \{s\}\}$.
$M_\pi = M \mid_{(V_\pi \times V_\pi)}$.
Algorytm będzie tak budował graf G_π by był on grafem najkrótszych ścieżek tzn. takim podgrafem G, że każda ścieżka G_π jest najkrótszą ścieżką w G.
Zauważmy, że graf G_π jest drzewem, a więc dla każdego wierzchołka v istnieje co najwyżej jedna ścieżka z s do v.

Przykład 1.1 *Graf i najkrótsze ścieżki w grafie.*

Fakty:

- Podścieżki najkrótszych ścieżek są najkrótszymi ścieżkami,
- $\delta(s, v) \leq \delta(s, w) + M(w, v)$ dla każdej krawędzi (w, v).

1.1 Relaksacja

Atrybut $d[v]$ utrzymuje najmniejszą obliczoną wagę ścieżki z s do v.
Inicjalizacja: początkowo wszystkie $d[v] = \infty$ za wyjątkiem $d[s] = 0$, $\pi[v] = nil$.
Relaksacja krawędzi (u, v) polega na sprawdzeniu czy przechodząc przez wierzchołek u można znaleźć ścieżkę krótszą

od dotychczasowej najkrótszej ścieżki do v i jeśli tak na zmodyfikowaniu $\pi[v]$ i $d[v]$.
RELAX(u,v,M)

1. if $d[v] > d[u] + M(u,v)$ then $d[v] := d[u] + M(u,v)$, $\pi[v] := u$;

Własności relaksacji

1. Po wykonaniu relaksacji zachodzi $d[v] \leq d[u] + M(u,v)$,
2. Warunek $d[v] \geq \delta(s,v)$ jest niezmiennikiem dowolnego ciągu relaksacji po wykonaniu inicjalizacji,
3. Kiedy $d[v] = \delta(s,v)$, to już więcej nie ulega zmianie.
4. Niech $s - - > u \rightarrow v$ będzie najkrótszą scieżką w grafie. Jeśli po wykonaniu ciągu relaksacji $d[u] = \delta(s,u)$ i zostanie wykonane $RELAX(u,v,M)$, to $d[v] = \delta(s,v)$.
5. Po wykonaniu inicjalizacji graf G_π jest drzewem o korzeniu s i własność ta jest niezmiennikiem każdego ciągu relaksacji.
6. Jeśli po wykonaniu inicjalizacji i dowolnego ciągu relaksacji w grafie G zachodzi $d[v] = \delta(s,v)$ dla każdego wierzchołka v, to G_π jest drzewem najkrótszych ścieżek w G.

Dijkstra(G,s)
1 inicjalizacja G, s;
2 $S := \emptyset$;
3 $Q := V$;
4 while $Q \neq \emptyset$;
5 $u := Extract-Min(Q)$
6 $S := S \cup \{u\}$;
7 for each $v \in L[u]$;
8 $RELAX(u, v, G)$;

Q - kolejka (priorytetowa),
Wierzchołki w Q są ustawione zgodnie ze wzrastającymi wagami d.
$Extract–Min(Q)$ oznacza wybranie i usunięcie z Q wierzchołka o minimalnej wadze d.

BELLMAN-FORD(G,s)
1 inicjalizacja G, s;
2 for $i := 1$ to $|V| - 1$;
3 do for each $(u, v) \in E$;
4 do $RELAX(u, v, G)$;