UDK 519.1
Izvorni znanstveni rad

Dr. sc. Antoaneta Klobučar, dr. sc. Miljenko Crnjac
Ekonomski fakultet u Osijeku

PRONALAŽENJE NAJKRAĆEG PUTA NA GRAFU

U članku se uvodi definicija udaljenosti dva vrha na grafu.

Nakon toga je dan Fordov algoritam za određivanje najkraćeg puta kao i način primjene opisanog algoritma.

Određivanje najkraćeg puta javlja se kad stvarno treba odrediti najkraći put, npr. između dvaju gradova, ali i u slučaju kad tražimo put kojim možemo najbrže ili možda najjeftinije stići.

Problem se sastoji u nalaženju puteva najmanje duljine između dvaju vrhova na grafu, između jednog vrha i svih ostalih, ili između svaka dva vrha.

Bez smanjenja općenitosti možemo pretpostaviti da su uklonjene sve petlje, svi višestruki bridovi reducirani na jedan brid (i to najkraći), da je graf orijentiran (inače bi svaki brid bilo moguće zamijeniti dvama orijentiranim bridovima).

Ako su bridovi neoznačeni duljinom, ali ravnopravni, uzimamo da je duljina svakog jednaka 1.

Brojeve pridružene bridovima nazvat ćemo duljinama, premda su moguće i druge interpretacije. Pretpostavljamo da su negativni. Duljinom puta smatra se suma svih duljina pojedinih puteva. Nepostojeći bridovi u grafu formalno neka imaju duljinu ∞.

Navodimo Fordov algoritam za nalaženje najkraćeg puta u grafu.

Označimo vrhove grafa sa $x_1, ..., x_n$, duljinu brida između vrhova x_i i x_j označimo sa $l(x_i, x_j)$. Pretpostavimo da tražimo najkraći put između x_1 i x_n.

Svakom vrhu x_i pridružimo broj λ_i. Prvo pretpostavimo da je $\lambda_1 = 0$ i $\lambda_i = \infty$ (i=2, ..., n).

Zatim radimo u etapama. U svakoj etapi svakom od vrhova koji još nisu označeni ($\lambda_i = \infty$) pridružimo minimum njegove trenutne oznake i udaljenosti preko posljednjeg konačno označenog vrha do vrha x_i.

Taj postupak primjenjujemo sve dok je moguće mijenjati brojeve λ_i. Pritom je moguće da neki brid (x_i, x_j) više puta promatramo. Pri završetku posla najkraći put dobivamo polazeći od x_n. Iz x_n idemo u njemu susjedan i najmanje udaljeni vrh, i tako redom.

Algoritam

$$for\ x \in V\ do \begin{cases} label(x) := \infty \\ final\ (x). = false \end{cases}$$

$$label(x_1) := 0$$
$$final(x_1) := true$$
$$last := x_1$$

for $x \in Adj(last)$ *do*

while not final (x_n) *do* {
if not final(x) *and*
$label(x) > label(last) + W_{last,x}$ *then*
$label(x) := label(last) + W_{last,x}$ *or*
$pre(x) := last$
}

$final(k) := true;\ last := k$

(* neka je k bilo koji vrh s minimalnom oznakom i final (k) = falce*)

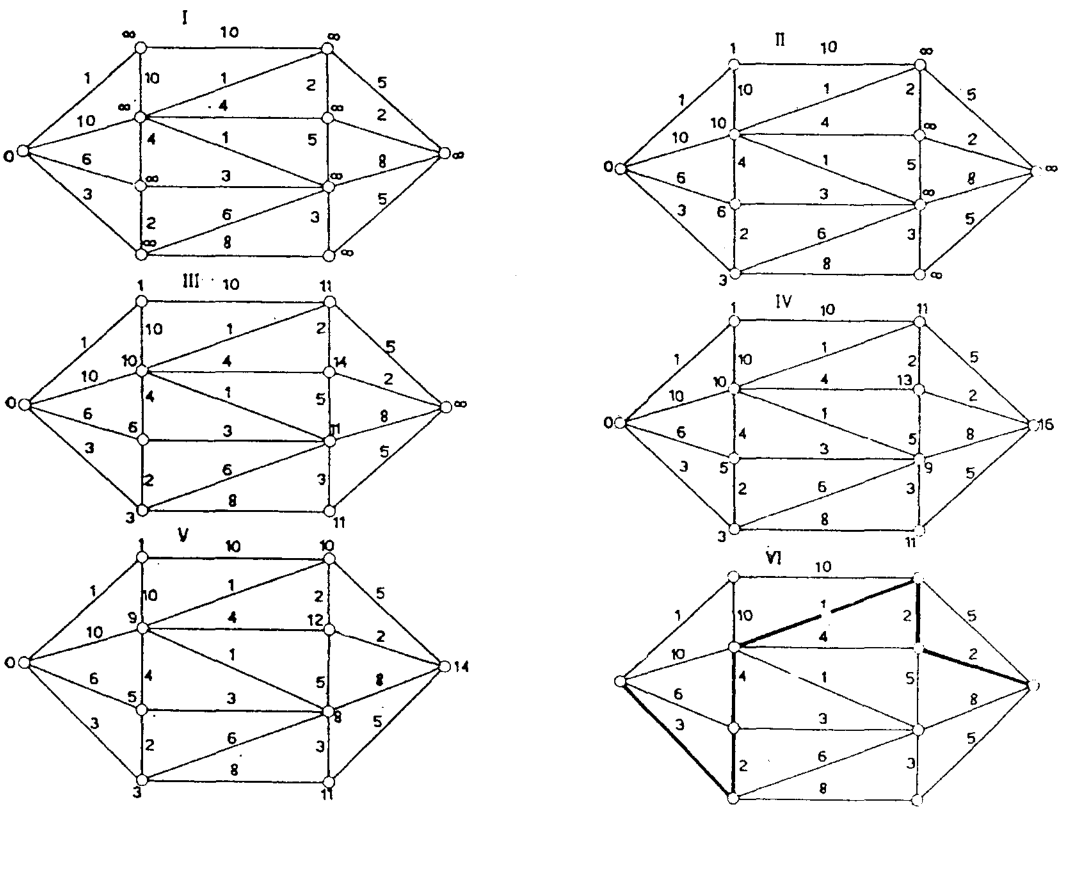
Oznake

V	skup svih vrhova grafa
Adj (x)	susjedni vrhovi vrha x
(**)	komentar

NAPOMENA

Algoritam je izražen u obliku koji nalikuje na program pisan u programskom jeziku Pascal. Odatle potječe značenje upotrebljavanih riječi for, do, if, true, false, then,...

Primjer 1.



LITERATURA

1. G. Chartrand and L. Lesniak, Graphs & Digraphs, third edition, Chapman & Hall, New York, 1996.

2. B. Roy, Algebre moderne et theorie des graphes, I, II, Paris 1969-1970.

3. D. Veljan, Kombinatorika s teorijom grafova. Školska knjiga, Zagreb, 1989.

4. Sendai, Graph Theory and Algorithms, Japan, Octobre 1980.

Antoaneta Klobučar, Ph. D.,
Miljenko Crnjac, Ph. D.

LOCATING THE SHORTEST CUT ON THE GRAPH

Summary

The definition of two apices distance on the graph is introduced in the paper. It is followed by the Ford algorithm to determine the shortest cut as well as the way of application of the described algorithm.