Cesty v grafoch

Stanislav Palúch

Fakulta riadenia a informatiky, Žilinská univerzita

22. marca 2011

Sled, ťah, cesta

Definícia

Nech $G = (V, H)$ je graf.

Sled (v_1–v_k sled) *v grafe G je ľubovoľná alternujúca (striedavá) postupnosť vrcholov a hrán tvaru*

$$\boldsymbol{\mu}(v_1, v_k) = (v_1, \{v_1, v_2\}, v_2, \{v_2, v_3\}, v_3, \dots, \{v_{k-1}, v_k\}, v_k). \tag{1}$$

Ťah (v_1–v_k ťah) *v grafe G je taký v_1–v_k sled v grafe G, v ktorom sa žiadna hrana neopakuje.*

Cesta (v_1–v_k cesta) *v grafe G je taký v_1–v_k sled v grafe G, v ktorom sa žiaden vrchol neopakuje.*

Pripúšťame aj tzv. **triviálny sled**, *pre $k = 1$, t. j. sled tvaru (v_1).*

Definícia

Nech $\overrightarrow{G} = (V, H)$ *je digraf.*

Orientovaný sled (orientovaný v_1–v_k*sled***)** *v digrafe* $\overrightarrow{G}$ *je ľubovoľná alternujúca (striedavá) postupnosť vrcholov a hrán tvaru*

$$\boldsymbol{\mu}(v_1, v_k) = (v_1, (v_1, v_2), v_2, (v_2, v_3), v_3, \dots, (v_{k-1}, v_k), v_k). \qquad (2)$$

Orientovaný ťah *v digrafe* $\overrightarrow{G}$ *je taký orientovaný* v_1–v_k *sled v digrafe* $\overrightarrow{G}$*, v ktorom sa žiadna hrana neopakuje.*

Orientovaná cesta *v digrafe* $\overrightarrow{G}$ *je taký orientovaný* v_1–v_k *sled v digrafe* $\overrightarrow{G}$*, v ktorom sa žiaden vrchol neopakuje.*

Definícia

Nech $\overrightarrow{G} = (V, H)$ *je digraf.*

Polosled (v_1–v_k **polosled)** *v digrafe* $\overrightarrow{G}$ *je ľubovoľná alternujúca (striedavá) postupnosť vrcholov a hrán tvaru*

$$\boldsymbol{\mu}(v_1, v_k) = (v_1, h_1, v_2, h_2, \dots, v_{k-1}, h_{k-1}, v_k),$$

v ktorej je každá hrana h_i *incidentná s obomi susednými vrcholmi* v_i, v_{i+1} *tak, že jeden z nich je začiatočným a druhý koncovým vrcholom hrany h.*

Poloťah (v_1–v_k **poloťah)** *v digrafe* $\overrightarrow{G}$ *je taký* v_1–v_k *polosled v digrafe* $\overrightarrow{G}$*, v ktorom sa žiadna hrana neopakuje.*

Polocesta (v_1–v_k **polocesta)** *v digrafe* $\overrightarrow{G}$ *je taký* v_1–v_k *polosled v digrafe* $\overrightarrow{G}$*, v ktorom sa žiaden vrchol neopakuje.*

Uzavretý sled

Definícia

Sled (polosled, ťah, poloťah)

$$\boldsymbol{\mu}(v_1, v_k) = (v_1, h_1, v_2, h_2, \ldots, v_{k-1}, h_{k-1}, v_k)$$

nazveme **uzavretý***, ak* $v_1 = v_k$. t.j. začiatočný a koncový vrchol sa rovnajú
Inak sled (polosled, ťah, poloťah) $\boldsymbol{\mu}(v_1, v_k)$ *nazveme* **otvorený***.*

Poznámka

Uzavretú cestu a polocestu nemožno týmto spôsobom definovať, pretože by došlo k sporu s požiadavkou, že jeden vrchol sa v týchto štruktúrach nesmie vyskytovať viackrát.
Namiesto uzavretej cesty a polocesty budeme mať cyklus a polocyklus.

Presná definícia týchto pojmov je nasledujúca:

Cyklus (orientovaný cyklus, polocyklus)

Definícia

Cyklus (orientovaný cyklus, polocyklus) *je netriviálny uzavretý ťah (orientovaný ťah, poloťah), v ktorom sa okrem prvého a posledného vrchola žiaden vrchol nevyskytuje viac než raz.*

Súvislosť grafov

Definícia

Hovoríme, že graf $G = (V, H)$ *je* **súvislý**, *ak pre každú dvojicu vrcholov* $u, v \in V$ *existuje* u–v *cesta. Inak hovoríme, že graf* G *je* **nesúvislý**.

Definícia

Komponent grafu $G = (V, H)$ *je jeho ľubovoľný maximálny súvislý podgraf.*

Definícia

Mostom *v grafe* $G = (V, H)$ *nazveme takú hranu grafu* G, *po vylúčení ktorej vzrastie počet komponentov.*

Artikuláciou *v grafe* G *nazveme taký vrchol, po vylúčení ktorého spolu s incidentnými hranami vzrastie počet komponentov.*

Typy súvislosti digrafov, komponent grafu

Definícia

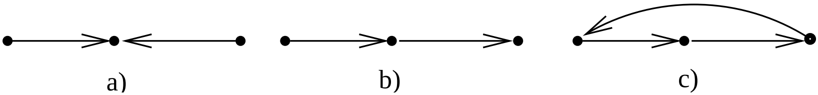
Nech $\overrightarrow{G} = (V, H)$ *je digraf.*
Povieme, že digraf $\overrightarrow{G}$ *je* **neorientovane súvislý**, *alebo* **slabo súvislý**, *ak pre každú dvojicu vrcholov* $u, v \in V$ *existuje v* G u*–*v *polosled; inak je digraf* $\overrightarrow{G}$ **nesúvislý**.

Povieme, že digraf $\overrightarrow{G}$ *je* **orientovane súvislý**, *alebo* **jednostranne súvislý**, *ak pre každú dvojicu vrcholov* $u, v \in V$ *existuje v* $\overrightarrow{G}$ u*–*v *sled alebo* v*–*u *sled.*

Digraf $\overrightarrow{G}$ *je* **silne súvislý**, *ak pre každú dvojicu vrcholov* $u, v \in V$ *existuje aj orientovaný* u*–*v *sled aj orientovaný* v*–*u *sled.*

Komponent digrafu $\overrightarrow{G}$ *je maximálny neorientovane súvislý podgraf digrafu* $\overrightarrow{G}$.

Typy súvislosti digrafov



Obr.: Digrafy s rôznymi typmi súvislosti.

a) neorientovane súvislý b) orientovane súvislý c) silne súvislý

Algoritmus

Tarryho algoritmus *na konštrukciu takého sledu v grafe $G = (V, H)$, ktorý začína v ľubovoľnom vrchole $s \in V$, prejde všetkými hranami komponentu grafu G a skončí vo vrchole s. Výsledný sled budeme volať* **Tarryho sled**.

- **Krok 1.** *Začni z ľubovoľného vrchola $s \in V$, polož $u := s$, $T = (u)$.*
  $\{T$ je inicializačne triviálny sled, obsahujúci jediný vrchol.$\}$
- **Krok 2.** *Ak môžeš, vyber k poslednému vrcholu u sledu T ďalšiu incidentnú hranu $\{u, v\}$ podľa nižšie uvedených pravidiel* **T1, T2** *a zaraď ju do sledu T. Zaznač si smer použitia hrany $\{u, v\}$. Ak doteraz vrchol v ešte nebol zaradený do sledu T, označ hranu $\{u, v\}$ ako hranu prvého príchodu. Pri výbere hrany dodržuj nasledujúce pravidlá:*

  T1: *Každú hranu možno v jednom smere použiť iba raz*
  T2: *Hranu prvého príchodu možno použiť, iba ak niet inej možnosti*
- **Krok 3.** *Ak taká hrana neexistuje – STOP.*
  Inak polož $u := v$ a pokračuj Krokom 2.

♣

Dĺžka sledu

Definícia

Nech $\boldsymbol{\mu}(u, v)$ je u–v sled (resp. orientovaný sled, resp. polosled) v hranovo ohodnotenom grafe $G = (V, H, c)$ (resp. digrafe $\overrightarrow{G} = (V, H, c)$).
Dĺžkou sledu (polosledu) $\boldsymbol{\mu}(u, v)$ *alebo tiež* **cenou sledu (polosledu)** *nazveme súčet ohodnotení jeho hrán, pričom ohodnotenie každej hrany započítavame toľkokrát, koľkokrát sa táto hrana v slede vyskytuje.*

Dĺžku sledu $\boldsymbol{\mu}(u, v)$ budeme značiť $d(\boldsymbol{\mu}(u, v))$.

Poznámka

Podľa definície sledu sa pripúšťa aj triviálny sled s jediným vrcholom a žiadnou hranou. Dĺžka takéhoto sledu je nulová.

Poznámka

Predchádzajúcou definíciou je definovaná i dĺžka ťahu, orientovaného ťahu, cesty, poloťahu a polocesty, pretože všetky tieto pojmy sú špeciálnym prípadom sledu, resp. polosledu.

Dĺžka ťahu, dĺžka cesty

Poznámka

Pre ťah, poloťah, cestu, a polocestu, (v ktorých sa podľa ich definície každá hrana môže vyskytovať len raz), môžeme zjednodušene definovať:

Dĺžka $d(\boldsymbol{\mu}(u,v))$ ťahu (poloťahu, cesty alebo polocesty) $\boldsymbol{\mu}(u,v)$ je súčet ohodnotení ich hrán, t. j.

$$d(\boldsymbol{\mu}(u,v)) = \sum_{h \in \boldsymbol{\mu}(u,v)} c(h).$$

Poznámka

Často býva užitočné definovať dĺžku sledu $\boldsymbol{\mu}(u,v)$ v grafe $G = (V, H)$, resp. digrafe $\overrightarrow{G} = (V, H)$, ktorý nie je hranovo ohodnotený, ako počet hrán sledu $\boldsymbol{\mu}(u,v)$. Takto definovaná dĺžka sledu je totožná s dĺžkou sledu v hranovo ohodnotenom grafe, (resp. digrafe) $G' = (V, H, c)$, kde $c(h) = 1$ pre každú hranu $h \in H$.

Najkratšia cesta v grafe a digrafe

Definícia

Najkratšia *u-v* **cesta** *v hranovo ohodnotenom grafe* $G = (V, H, c)$ *(resp. v hranovo ohodnotenom digrafe* $\overrightarrow{G} = (V, H, c)$*) je tá u-v cesta v G (resp. tá orientovaná u-v cesta v* $\overrightarrow{G}$*), ktorá má najmenšiu dĺžku.*

Dohoda

- *Všetky algoritmy na hľadanie najkratšej cesty (okrem Floydovho algoritmu) budú formulované* **pre hranovo ohodnotené digrafy** $\overrightarrow{G} = (V, H, c)$*, v ktorých sa predpokladá, že* $c(h) \geq 0$.
- *Predpokladáme, že* $0 \notin V$.

Základný algoritmus pre hľadanie najkratšej cesty

Algoritmus

Základný algoritmus *na hľadanie najkratších orientovaných u–v ciest z pevného vrcholu $u \in V$ do všetkých dosiahnuteľných vrcholov $v \in V$ v hranovo ohodnotenom digrafe $\overrightarrow{G} = (V, H, c)$ s nezápornou cenou hrany $c(h)$ (a kde $0 \notin V$).*

- **Krok 1. Inicializácia.**
  Pre každý vrchol $i \in V$ priraď dve značky $t(i)$ a $x(i)$.
  {Značka $t(i)$ predstavuje horný odhad dĺžky doteraz nájdenej najlepšej u–i cesty a $x(i)$ jej predposledný vrchol.}
  Polož $t(u) := 0$, $t(i) := \infty$ pre $i \in V$, $i \neq u$ a $x(i) := 0$ pre každé $i \in V$.
- **Krok 2.** *Zisti, či existuje orientovaná hrana $(i, j) \in H$, pre ktorú platí*

$$t(j) > t(i) + c(i, j). \tag{3}$$

  Ak taká hrana $(i, j) \in H$ existuje, potom polož

$$t(j) := t(i) + c(i, j), \quad x(j) := i$$

  a opakuj Krok 2.
- **Krok 3.** *Ak taká orientovaná hrana (z kroku 2.) neexistuje, STOP.* ♣

Algoritmus

Dijkstrov algoritmus *Algoritmus pre zostrojenie najkratšej orientovanej u–v cesty v hranovo ohodnotenom digrafe* $\overrightarrow{G} = (V, H, c)$ *s nezáporným ohodnotením hrán (a kde* $0 \notin V$*).*

- **Krok 1.** *Inicializácia. Pre každý vrchol* $i \in V$ *priraď dve značky* $t(i)$ *a* $x(i)$*. Značky* $t(i)$ *budú dvojakého druhu, a to dočasné (ktoré sa ešte v priebehu výpočtu môžu zmeniť) a definitívne (ktoré sa už nemôžu zmeniť).*

  Polož $t(u) = 0$, $t(i) = \infty$ *pre* $i \in V$, $i \neq u$ *a* $x(i) = 0$ *pre každé* $i \in V$*. Zvoľ riadiaci vrchol* $r := u$ *a značku* $t(\ )$ *pri vrchole* $r = u$ *prehlás za definitívnu, ostatné značky za dočasné.*

- **Krok 2.** *Ak je* $r = v$*, STOP. Ak* $t(v) < \infty$*, značka* $t(v)$ *predstavuje dĺžku najkratšej u–v cesty, ktorú zostroj spätne z v pomocou smerníkov* $x(i)$*. Inak pre všetky hrany tvaru* $(r, j) \in H$*, kde* j *je vrchol s dočasnou značkou, urob:*

  Ak $t(j) > t(r) + c(r, j)$*, potom* $t(j) := t(r) + c(r, j)$*,* $x(j) := r$ *.*

  Ponechaj zmenené značky ako dočasné.

Algoritmus (- pokračovanie)

- **Krok 3.** *Zo všetkých dočasne označených vrcholov nájdi ten vrchol i, ktorý má značku $t(i)$ minimálnu.*

  Značku pri tomto vrchole i prehlás za definitívnu a zvoľ za nový riadiaci vrchol $r := i$.
  {Pokiaľ existuje viac vrcholov s rovnakou minimálnou značkou, akú má vrchol i, za definitívnu značku môžeme prehlásiť len značku pri jednom z týchto vrcholov – ten potom berieme za riadiaci. Na tie ďalšie dôjde v nasledujúcich krokoch výpočtu.}
  GOTO Krok 2.

♣

Poznámka

Ak upravíme podmienku zastavenia v Kroku 2. na podmienku:
Ak sú značky všetkých vrcholov definitívne, STOP,
dostaneme verziu algoritmu, ktorá hľadá najkratšie cesty z vrchola u do všetkých dosiahnuteľných vrcholov.

Algoritmus

Floydov algoritmus *na výpočet matice vzdialeností vrcholov v hranovo ohodnotenom grafe, resp. digrafe* $G = (V, H, c)$, *kde* $c(h) \geq 0$.

- **Krok 1.** *Zostroj maticu* $\mathbf{C} = (c_{ij})$, *ktorej prvky sú definované nasledovne:*

$$c_{ii} = 0 \quad \textit{pre všetky} \quad i \in V$$

a pre všetky $i, \ j$, *také, že* $i \neq j$

$$c_{ij} = \begin{cases} c(i,j), & \textit{ak } \{i,j\} \in H, \textit{ resp. } (i,j) \in H \\ \infty, & \textit{ak } \{i,j\} \notin H, \textit{ resp. } (i,j) \notin H \end{cases}$$

Zostroj aj maticu $\mathbf{X} = (x_{ij})$, *kde*

$$x_{ii} = i \quad \textit{pre všetky} \quad i \in V$$

a pre všetky $i, \ j$, *také, že* $i \neq j$

$$x_{ij} = \begin{cases} i \quad , & \textit{ak } \{i,j\} \in H, \textit{ resp. } (i,j) \in H \\ \infty, & \textit{ak } \{i,j\} \notin H, \textit{ resp. } (i,j) \notin H \end{cases}$$

Algoritmus (- pokračovanie)

- **Krok 2.** *Urob pre všetky* $k = 1, 2, \dots, n = |V|$*:*
  Pre všetky $i \neq k$ *také, že* $c_{ik} \neq \infty$*, a pre všetky* $j \neq k$ *také, že* $c_{kj} \neq \infty$*, urob:*
  Ak $c_{ij} > c_{ik} + c_{kj}$*, potom polož:*

$$c_{ij} := c_{ik} + c_{kj}$$
$$x_{ij} := x_{kj}$$

♣

Po skončení Floydovho algoritmu je matica **C** maticou vzdialeností vrcholov grafu, resp. digrafu G.

Matica **X** obsahuje pre každú dvojicu vrcholov i, j takú, že j je dosiahnuteľný z i, predposledný vrchol najkratšej i–j cesty.

Ak potrebujeme nájsť najkratšiu i–j cestu, využijeme maticu smerníkov **X** nasledovne:
Pre predposledný vrchol j_1 najkratšej i–j cesty je $j_1 = x_{ij}$. Ďalší vrchol tejto cesty (odzadu) je $j_2 = x_{ij_1}$, ďalší $j_3 = x_{ij_2}$ atď., pokiaľ nedôjdeme do vrchola i.

Floyd

Floydov algoritmus 5 (str. 117) možno tiež modifikovať tak, že v prípade všeobecného digrafu nájde záporný cyklus.
Stačí v kroku 1. definovať začiatočnú maticu **C** nasledovne

$$c_{ij} = \begin{cases} c(i,j), & \text{ak } \{i,j\} \in H, \quad \text{resp.} \quad (i,j) \in H \\ \infty, & \text{ak } \{i,j\} \notin H, \quad \text{resp.} \quad (i,j) \notin H \end{cases}$$

Matica **C** má na rozdiel od štandardného Floydovho algoritmu na hlavnej diagonále ∞. Matica **X** je bez zmeny. Krok 2. je rovnaký.
Pozor! Treba meniť aj prvky hlavnej diagonály!!!

Po ukončení práce tohto algoritmu budú prvky c_{ii} na diagonále rovné dĺžke najkratšieho i–i cyklu.

Ak sa na hlavnej diagonále matice **C** v priebehu výpočtu Floydovým algoritmom objaví záporné číslo c_{jj} , objavili sme tým cyklus zápornej ceny obsahujúci vrchol j.

Tento cyklus určíme pomocou matice smerníkov **X**.

Algoritmus

Label-set a Label-correct implementácia algoritmu *na hľadanie najkratších orientovaných u–v ciest z pevného vrchola $u \in V$ do všetkých ostatných vrcholov $v \in V$ v hranovo ohodnotenom digrafe $\overrightarrow{G} = (V, H, c)$ s nezápornou cenou orientovanej hrany $c(h)$, kde $0 \notin V$.*

- **Krok 1: Inicializácia.**
  Polož $t(u) := 0$, $t(i) := \infty$ pre $i \in V$, $i \neq u$ a $x(i) := 0$ pre každé $i \in V$. Polož $\mathcal{E} := \{u\}$.

- **Krok 2:** *Vyber $r \in \mathcal{E}$, polož $\mathcal{E} := \mathcal{E} - \{r\}$.*
  Pre všetky orientované hrany tvaru $(r, j) \in H$ urob:
  Ak $t(j) > t(r) + c(r, j)$, potom
  $$t(j) := t(r) + c(r, j),\ x(j) := r,\ \mathcal{E} := \mathcal{E} \cup \{j\}.$$

- **Krok 3:** *Ak $\mathcal{E} \neq \emptyset$, choď na Krok 2.*
  Ak $\mathcal{E} = \emptyset$, potom $t(i)$ predstavuje dĺžku najkratšej orientovanej u–i cesty pre každý vrchol i. Najkratšiu orientovanú u–i cestu zostroj potom spätne pomocou značiek $x(i)$ [redacted].

♣

Label-set a Label-correct implementácia

Ak v druhom kroku posledného algoritmu vyberáme $r \in \mathcal{E}$ ľubovoľne, dostávame implementáciu základného algoritmu, ktorú voláme **label correct algoritmus**.

Ak za prvok $r \in \mathcal{E}$ vyberáme prvok z najmenšou značkou $t()$, potom dostaneme implementáciu Dijkstrovho algoritmu, ktorú voláme **label set algoritmus**.
Ak potrebujeme len jednu u-v cestu, label set algoritmus zastavíme v okamihu vybratia vrchola v z množiny $\mathcal{E}$.

Pre label correct algoritmus je výhodné organizovať $\mathcal{E}$ ako zásobník, pre label set algoritmus sa $\mathcal{E}$ organizuje ako prioritný front, prípadne ako halda.

Aby sme do zásobníka, resp. do prioritného frontu $\mathcal{E}$ nevkladali ten istý vrchol viackrát, je vhodné ku každému vrcholu $v \in V$ udržiavať indikátor hovoriaci, či vrchol v je v množine $\mathcal{E}$.