Paralelné algoritmy, časť č. 7

František Mráz

Kabinet software a výuky informatiky, MFF UK, Praha

Paralelné algoritmy, 2011/2012

Obsah

Komponenty súvislosti grafu

Vstup: Matica susednosti neorientovaného grafu $G = (V, E)$, kde $V = \{1, \ldots, n\}$

Výstup: vektor C: $C[i] = C[j] = k \Leftrightarrow i$ a j patria do tej istej komponenty súvislosti grafu G a naviac k je najmenšie číslo vrcholu z tejto komponenty

---

Alg.:

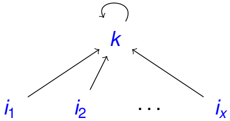
- inicializácia: $C[i] := i$, pre všetky $i = 1, \ldots, n$

  vrcholy i s rovnakou hodnotou $C[i]$ tvoria tzv. **pseudovrchol** i reprezentovaný **p-vrcholom** $C[i]$

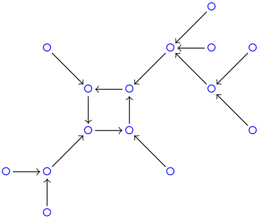
  Invariant: p-vrchol i je vrchol s najmenším číslom v pseudovrchole i
- idea: jedna iterácia alg. zmenší počet pseudovrcholov v každej komponente aspoň o polovicu – za $\log n$ iterácií bude každej komponente zodpovedať jediný pseudovrchol

Vytváranie pseudovrcholov

- ak interpretujeme $(i, C[i])$ ako hranu orientovaného grafu, tak jeden pseudovrchol tvorí tzv. zakorenenú hviezdu



- v jednom kroku sa každý p-vrchol pseudovrcholu, ktorý nepokrýva celú komponentu, pripojí na nejaký p-vrchol z tej istej komponenty súvislosti – vzniknú tzv. stromové cykly



- algoritmus strieda vytváranie stromových cyklov a ich "sťahovanie" do hviezd

Krok I – vytvorenie stromových cyklov

1. **for all** $i \in V$ **in parallel do**
   $$T[i] := \min_{j \in V}\{C[j] \mid (A[i,j] = 1) \;\&\; (C[j] \neq C[i])\}$$
   - ak sa počíta minimum cez prázdnu množinu, tak $T[i] := C[i]$
   - "$T[i]$ priradíme najmenší p-vrchol napojený na vrchol i a ležiaci v inom pseudovrchole, ak i nie je napojený na žiaden vrchol ležiaci v inom pseudovrchole, tak sa pripojí na svoj p-vrchol."
2. **for all** $i \in V$ **in parallel do**
   $$T[i] := \min_{j \in V}\{T[j] \mid (C[j] = i) \;\&\; (T[j] \neq i)\}$$
   - ak sa počíta minimum cez prázdnu množinu, tak $T[i] := C[i]$
   - "Ak je i p-vrchol, tak $T[i]$ bude najmenší p-vrchol susediaci s nejakým vrcholom v pseudovrchole i. Ak i nie je p-vrchol, tak $T[i]$ bude p-vrchol vrcholu i."

skrátene:

for all $i \in V, C[i] = i$ **in parallel do**
$$T[i] := \min_{j \in V}\{\, C[j] \mid \exists k : \; (C[k] = i) \;\&\; (A[j,k] = 1) \,\}$$

Krok I – vytvorenie stromových cyklov

pokrač.

ak komponenta nie je pokrytá pseudovrcholom

- každý pseudovrchol komponenty obsahuje aspoň jeden vrchol spojený s nejakým uzlom iného pseudovrcholu
- graf s hranami $(i, T[i])$ obsahuje cyklus – práve jeden a dĺžky presne 2:
  - dĺžky 1 nie, lebo $i \neq T[i]$
  - dĺžky > 2 nie, lebo pre nejaké i na cykle by $T[i]$ nemohlo byť najmenším p-vrcholom z vrcholov susediacich s vrcholom pseudovrcholu i

Krok II – sťahovanie stromových cyklov do zakorenených hviezd

všetky vrcholy jedného stromového cyklu sa zlejú do jedného pseudovrcholu, ktorý má najmenšie číslo:

- zdvojovaním na hranách $(i, T[i])$ po $\log n$ krokoch $T[i]$ bude odkazovať na jeden z dvoch vrcholov v cykle
- vrcholy pripojíme na menší z dvoch vrcholov cyklu

1. **for all** $i \in V$ **in parallel do** $B[i] := T[i]$
2. **repeat** $\log n$ **times**
   for all $i \in V$ **in parallel do** $T[i] := T[T[i]]$
3. **for all** $i \in V$ **in parallel do** $C[i] := \min\{\ B[T[i]], T[i]\ \}$

celkovo:

inicializácia
repeat $\log n$ **times**
  Krok I
  Krok II

Zložitosť

$T[n]$	$P[n]$	kód
$O(1)$	$O(n)$	**for all** $i \in V$ **in parallel do** $C[i] := i$
		repeat $\log n$ **times**
$O(\log n)$	$O(n^2)$	**for all** $i \in V$ **in parallel do**
		$T[i] := \min_{j \in V}\{C[j] \mid (A[i,j] = 1) \,\&\, (C[j] \neq C[i])\}$
$O(\log n)$	$O(n^2)$	**for all** $i \in V$ **in parallel do**
		$T[i] := \min_{j \in V}\{T[j] \mid (C[j] = i) \,\&\, (T[j] \neq i)\}$
$O(1)$	$O(n)$	$B[i] := T[i]$
		repeat $\log n$ **times**
$O(\log n)$	$O(n)$	**for all** $i \in V, C[i] = i$ **in parallel do** $T[i] := T[T[i]]$
$O(1)$	$O(n)$	**for all** $i \in V, C[i] = i$ **in parallel do**
		$C[i] := \min\{\, B[T[i]], T[i] \,\}$

- najkratší čas $O(\log^2 n)$ s $O(n^2)$ procesormi
- zlepšenie: minimum z n čísel v čase $O(\log n)$ pomocou $O(\frac{n}{\log n})$ procesorov – čas $O(\log^2 n)$ s $O(\frac{n^2}{\log n})$ procesormi
- ďalšie zlepšenie: po každej iterácii vonkajšieho **repeat**-cyklu urobiť kompresiu grafu na aktívne p-vrcholy pseudovrcholov, ktoré nepokrývajú celé komponenty

Vylepšený algoritmus

Vstup: matica susednosti A veľkosti $n \times n$
Výstup: pole D veľkosti n, $D[i] =$najmenšie číslo vrcholu v komponente súvislosti, v ktorej leží vrchol i

krok	kód
	$A_0 := A; n_0 := n; k := 0;$
	while $n_k > 0$ **do**
	begin $k := k + 1;$
(1)	$C[i] := \begin{cases} \min \{ j \mid (A_{k-1}[j, i] = 1) \ \& \ (i \neq j) \} \\ \text{ak také } j \text{ neexistuje, tak } i \end{cases}$
(2)	stiahni pseudovrcholy (stromové cykly) definované prostredníctvom C na zakorenené hviezdy
(3)	každý koreň netriviálnej hviezdy označ ako “nový p-vrchol” a očísluj ich; $r(i)$ nech označuje číslo vrcholu i
(4)	$n_k :=$počet nových p-vrcholov
(5)	vytvor maticu A_k tvaru $n_k \times n_k$ – maticu susednosti pre nové p-vrcholy
	end
(6)	pre každý vrchol urči $D[i]$ – rovná sa i, ak $C[i] = i$, inak opačným postupom než (5) expanduj každý p-vrchol v na pseudovrchol a pre všetky prvky j pseudovrcholu $D[j] := v$

Return

Vylepšený algoritmus

zložitosť

- Pozorovanie: $n_k \leq \frac{n_{k-1}}{2}$, pre $k \geq 1$

Veta

Komponenty súvislosti grafu s n vrcholmi sa z matice susednosti dajú spočítať v čase $O(\log^2 n)$ pomocou $O(\frac{n^2}{\log^2 n})$ procesorov na COMMON PRAM.

Dôkaz:

krok	$T[n]$	$P[n]$
(1)	$O(\log n_{k-1})$	$O(\frac{n_{k-1}^2}{\log n_{k-1}})$
(2)	$O(\log n_{k-1})$	$O(n_{k-1})$
(3),(4)	$O(\log n_k)$	$O(\frac{n_k}{\log n_k})$

- krok (5): **for each** $i, j, 1 \leq i, j \leq n_{k-1}$ **in parallel do**
  if $A_{k-1}[i,j] = 1$ **then** $A_k[r(i), r(j)] := 1$

  čas $O(1)$, $O(n_{k-1}^2)$ operácií

Vylepšený algoritmus

zložitosť

- nech máme k dispozícii p procesorov
  n_k miním z n_k čísel – na výpočet 1 minima $\frac{p}{n_k}$ procesorov
  celkový čas na výpočet miním:

$$
\begin{aligned}
\sum_{k=0}^{\log n} (\log n_k) \left(1 + \frac{n_k}{\log n_k} \cdot \frac{1}{\frac{p}{n_k}}\right) &\leq \sum_{k=0}^{\log n} (\log n_k) \left(1 + \frac{n_k^2}{\log n_k} \cdot \frac{1}{p}\right) \\
&\leq \sum_{k=0}^{\log n} (\log n_k) + \sum_{k=0}^{\log n} \frac{1}{p} \left(\frac{n}{2^k}\right)^2 \\
&\leq \log^2 n + \frac{n^2}{p} \cdot \sum_{k=0}^{\log n} \left(\frac{1}{2^{2k}}\right) \\
&\leq O\left(\log^2 n + \frac{n^2}{p}\right)
\end{aligned}
$$

pre $p = \frac{n^2}{\log^2 n}$ dostaneme čas $O(\log^2 n)$

Vylepšený algoritmus

zložitosť

- podobne pre krok (5): s p procesormi sa dajú všetky iterácie kroku (5) urobiť v čase:

$$\sum_{k=0}^{\log n} \frac{n_k^2}{p} \leq \frac{1}{p} \sum_{k=0}^{\log n} \left(\frac{n}{2^k}\right)^2 \leq \frac{n^2}{p}$$

pre $p = \frac{n^2}{\log^2 n}$ dostaneme čas $O(\log^2 n)$

Minimálna kostra grafu

Lemma

Nech $G = (V, E)$ je neorientovaný ohodnotený graf. Pre každý vrchol $u \in V$ nech $\{u, C(u)\}$ je najlacnejšia hrana z u. Potom

1. *existuje minimálna kostra grafu G, ktorá obsahuje (naraz) všetky hrany $\{u, C(u)\}$ pre všetky vrcholy $u \in V$.*
2. *C definuje les stromových cyklov dĺžky 2.*

- TRIK: predpokladáme, že všetky hrany majú navzájom rôzne ohodnotenie $\Rightarrow$ minimálna kostra je určená jednoznačne
- algoritmus analogický algoritmu pre komponenty súvislosti Komponenty súvislosti
  - v kroku (1) $C[i] := j$ také, že $cena(i, j) = \min \{ cena(i, k) \mid i \neq k \}$
  - Pozor: pri kompresii grafu v kroku (5) musíme pre každú hranu v A_k poznamenať cenu a reprezentanta, ktorý ju má!
    Ako?

Uchovanie reprezentanta

1. fáza:
   - pre každý vrchol usporiadať hrany do "nových p-vrcholov" podľa čísla p-vrcholu
   - potom pre každý vrchol v každej skupine s rovnakým p-vrcholom vybrať minimum podľa ceny $\Rightarrow$ z matice rozmerov $n_{k-1} \times n_{k-1}$ dostaneme maticu susednosti rozmerov $n_{k-1} \times n_k$ – hrany s minimálnou cenou z každého vrcholu do nových p-vrcholov
   - čas $O(\log n_{k-1})$ s $O(n_{k-1} \frac{n_{k-1}}{\log n_{k-1}})$ procesormi, z toho triedenie na začiatku sa dá urobiť v čase $O(\log n_{k-1})$ s $O(n_{k-1})$ procesormi – tento algoritmus bude neskôr; počítanie miním – prefixový výpočet

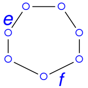
2. fáza: transponovane ten istý postup $\Rightarrow$ z matice rozmerov $n_{k-1} \times n_k$ dostaneme maticu susednosti rozmerov $n_k \times n_k$ – hrany s minimálnou cenou z každého nového p-vrcholu do nových p-vrcholov

Celkom: čas $O(\log^2 n)$ s $O(\frac{n^2}{\log^2 n})$ procesormi

ľubovoľná kostra – analogicky

2-súvislé komponenty

- hrany e, f patria do tej istej komponenty 2-súvislosti neorientovaného grafu $\equiv_{df}$ existuje prostý cyklus obsahujúci e, f



graf sa dá rozložiť na strom 2-súvislých komponent

Vstup: $G = (V, E)$ neorientovaný graf zadaný maticou susednosti

Výstup: pole $B[e] = B[f]$, pre hrany $e, f \in E$, $\Leftrightarrow$ e, f ležia v tej istej 2-súvislej komponente grafu G

Idea: prevodom na výpočet komponent súvislosti

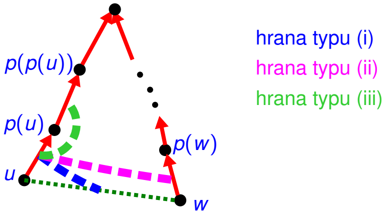
- nájdeme nejakú kostru $T = (V, E')$ grafu G – je to strom, zakoreníme ho, vrcholy T očíslujeme podľa PREORDER, spočítame $nd(v) =$ počet potomkov uzlu v vrátane v.
- každá hrana, ktorá nie je v kostre, generuje cyklus – *bázický cyklus*

  $e_1 R_c e_2 \Leftrightarrow$ existuje bázický cyklus, ktorý obsahuje e_1 aj e_2
  Tranzitívny uzáver relácie R_c je rozklad na 2-súvislé komponenty

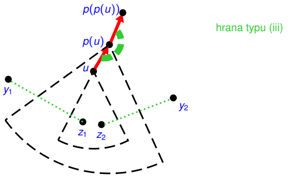
Pomocný graf

$G' = (E, \overline{E})$ – vrcholy sú hrany grafu G,
$T = (V, E')$ je kostra, $p(v)$ je otec vrcholu v; do $\overline{E}$ dáme hrany tvaru

(i) $\{\{u, p(u)\}, \{u, w\}\}$, ak $\{u, w\} \in E \setminus T$, $u < w$ a u nie je koreň
(ii) $\{\{u, p(u)\}, \{w, p(w)\}\}$, ak $\{u, w\} \in E \setminus T$, u a w sú neporovnateľné (nie je jeden následníkom druhého)
(iii) $\{\{w, p(w)\}, \{p(w), p(p(w))\}\}$, ak ani w, ani $p(w)$ nie je koreň a existuje hrana $\{z, y\} \in E \setminus T$ taká, že z je následníkom w a y nie je následníkom $p(w)$



Ako určiť hrany typu (iii)?



(iii) $\{\{w, p(w)\}, \{p(w), p(p(w))\}\}$, ak ani w, ani $p(w)$ nie je koreň a existuje hrana $\{z, y\} \in E \setminus T$ taká, že z je následníkom w a y nie je následníkom $p(w)$

- $low(v) =$ najmenšie číslo vrcholu, do ktorého vedie hrana z nejakého potomka vrcholu v
- $high(v) =$ najväčšie číslo vrcholu, do ktorého vedie hrana z nejakého potomka vrcholu v
- podmienka (iii) $\equiv (low(w) < p(w)) \quad \&$
  $(high(w) \geq preorder(p(w)) + nd(p(w)))$
- $low(v)$ a $high(v)$ pre všetky vrcholy sa dá spočítať metódou miním na intervaloch dokonca na EREW PRAM v čase $O(\log n)$ s $O(n)$ procesormi

2-súvislé komponenty
(dokončenie)

G'' je podgraf G' obsahujúci len hrany typov (ii) a (iii)

- G'' má maximálne $n - 1$ uzlov (hrany z T)
- G'' má menej než $(n - 1)^2$ hrán
- nájdeme komponenty súvislosti grafu G''
- pre každú nekostrovú hranu nájdeme nejakú kostrovú hranu, ku ktorej je pripojená (hrany typu (i)) a pridáme ju do jej komponenty súvislosti

Veta

Nech algoritmus $\mathcal{A}$ nájde komponenty súvislosti grafu (V, E) v čase $f(|V|, |E|)$ s $g(|V|, |E|)$ procesormi na modele M ($M \in \{EREW, CREW, CRCW\}$). Potom existuje algoritmus $\mathcal{B}$, ktorý nájde 2-súvislé komponenty grafu (V, E) v čase $O(f(|V|, |E|) + \log |V|)$ s $O(g(|V|, |E|) + |E| + |V|)$ procesormi na modele M.