$S_k^{j+1} := \{T \mid T \in S_k^j,\ e_k^j \in T\}$,

$S_{j+1}^{j+1} := \{T \mid T \in S_k^j,\ e_k^j \notin T\}$.

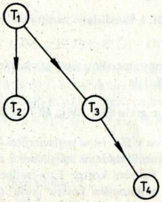
Ak definujeme $IN_k^{j+1} := IN_k^j \cup \{e_k^j\}$, $OUT_k^{j+1} := OUT_k^j$
a $IN_{j+1}^{j+1} := IN_k^j$, $OUT_{j+1}^{j+1} := OUT_k^j \cup \{e_k^j\}$,
tak S_k^{j+1} a S_{j+1}^{j+1} budú mať tvar (*). Tiež vidíme, že $T_k \in S_k^{j+1}$ a $T_{j+1} \in S_{j+1}^{j+1}$. Množiny S_i^{j+1} ($i = 1, \ldots, j$: $j \neq k$) a s nimi aj príslušné *IN* a *OUT* „zdedíme" ($S_i^{j+1} := S_i^j$). Tým máme urobený požadovaný rozklad všetkých kostier do $(j+1)$-ej úrovne a nájdených $j+1$ najlacnejších kostier. (Príklad takého rozkladania až do štvrtej úrovne je na obr. 1 v tvare stromu rozkladu.) ●

Tým je daný princíp algoritmu a jeho zdôvodnenie.

POZNÁMKY K IMPLEMENTÁCII

Pri počítaní je výhodné mať pri každom S_i^j ešte číslo t_i^j udávajúce cenu druhej najlacnejšej kostry v S_i^j a príslušné hrany e_i^j a f_i^j. Explicitne nepoznáme množiny S_i^j, ale iba istú informáciu o nich (to nám stačí), a menovite: t_i^j, e_i^j, f_i^j, T_i, IN_i^j, OUT_i^j (ale budeme hovoriť o S_i^j). Množiny S_i^j udržiavame usporiadané (v prioritnom fronte) podľa veľkosti čísel t_i^j. Takto rozkladáme vždy prvú množinu S_k^j, vynecháme ju a dve vzniknuté množiny S_k^{j+1} a S_{j+1}^{j+1} posúvame podľa čísel t_k^{j+1} a t_{j+1}^{j+1}, čím získame nové usporiadanie. Toto vsúvanie možno urobiť v čase $O(\log K)$ (metódou bisekcie), a teda v čase $O(m)$, lebo kostier nie je viac ako 2^m.

Uvedené zakódovanie množín S_i^j je však dosť objemné – vyžaduje priestor veľkosti rádove m, a teda celkove mK, čo je nepríjemné. Toto však možno zredukovať takto: Definujme orientovaný strom D takto: Vrcholmi sú kostry T_i (doteraz nájdené) a šíp z T_i do T_j existuje práve vtedy, ak kostra T_j vznikla z T_i



Obr. 2. Strom *D* pre príklad z obr. 1.

výmenou jednej hrany (nech $T_j = T_i - e_j + f_j$). Budeme predpokladať, že všetci synovia (nasledovníci) T_j toho istého vrcholu sú na diagrame stromu D usporiadaní tak, že j narastá zľava doprava. Príklad stromu pre obr. 1 je na obr. 2. Predpokladajme, že máme doteraz nájdené kostry $T_1, T_2, \ldots, T_p$, takže poznáme príslušný strom D. Potom objemnú informáciu o S_i^k (t. j. T_i, IN_i^k, OUT_i^k) získame takto:

Nech P je T_1—T_i cesta v D. Uvažujme množiny:
$O_1 := \{e_j \mid T_j$ leží na P a $j > 1\}$,
$O_2 := \{f_j \mid T_j$ leží na P a $j > 1\}$,
$T := (T_1 \cup O_2) - O_1$,
$I_1 := \{e_j \mid T_j$ je „ľavý brat" nejakého T_q z $P\}$,
$I_2 := \{e_j \mid T_j$ je synom kostry $T_i\}$.

Potom možno ukázať, že $T_i = T$, $IN_i^k = I_1 \cup I_2$, $OUT_i^k = O_1$ (5.Z.8). Takto vystačíme s priestorom $O(1)$ na každé S_i^j a sumárne stačí priestor $O(K+m)$ na celý algoritmus (kostry $T_1, T_2, \ldots$, postupne dávame na výstup).

Na vyrobenie informácie o S_i^j stačí $O(m)$ operácií.

Zložitá úloha v algoritme je hľadať najlacnejšiu susednú kostru. Výhodné je na začiatku usporiadať hrany grafu G od najlacnejšej po najdrahšiu do postupnosti $h_1, h_2, \ldots, h_m$ (v čase $O(m \log m)$). Potom sa dobre hľadajú dvojice zámenných hrán (g, h) pre danú kostru T takto:

Ak $h_1 \in G - T - OUT$, nájdeme všetky vhodné hrany $g_1 \in T - IN$ a nazveme ich potom nevhodnými. Prejdeme na h_2 atď. Takto (g_1, h_1) je najlepšia dvojica pre g_1, lebo h_1 je najlacnejšia hrana. Podobne (g_2, h_2) je najlepšia dvojica pre každé g_2 (lebo s h_1 sú vymeniteľné už nájdené hrany g_1, ktoré sme pri hľadaní g_2 vylúčili). Takto prejdeme všetky hrany h_i a nájdeme požadovanú najlepšiu dvojicu (g, h).

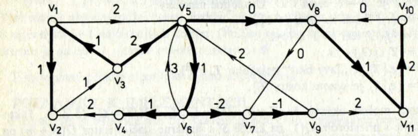
Gabow (1977) dáva podrobnú implementáciu celého algoritmu a ukazuje, že vystačíme s časom $O(Km \log m)$ (ba ešte trochu lepším). Tiež uvádza modifikáciu poskytujúcu všetky kostry grafu (nech je ich N) od najlacnejšej po najdrahšiu v čase $O(Nm)$ a v priestore $O(N+m)$.

5. F
Najlacnejšia zdrojová kostra digrafu

K úlohe o najlacnejšej kostre v grafe možno „prirodzene" priradiť viac úloh v digrafe. Jednou z nich je nasledujúca.

Daný je digraf G a každej jeho hrane e je priradené reálne číslo (cena) $c(e)$. Úlohou je nájsť najlacnejšiu zdrojovú kostru digrafu G.

Pripomíname, že v úlohe nie je predpísaný vrchol, ktorý má byť zdrojom (vrchol, z ktorého je každý iný dosiahnuteľný). Viaceré modifikácie tejto úlohy možno previesť na ňu (pozri cvičenia). Hneď vidieť, že úloha najlacnejšej kostry v grafe je špeciálnym prípadom (rebro nahradíme dvojicou protišípov). Na obr. 1 je príklad digrafu a v ňom hrubými čiarami je vyznačená jedna najlacnejšia zdrojová kostra T ($c(T)=6$ a zdrojom je v_3).



Obr. 1. Najlacnejšia zdrojová kostra (hrubé čiary).

Ako prví túto úlohu formulovali a vyriešili Chu a Liu (1965). Nezávisle túto, resp. nepodstatne zmenenú úlohu vyriešili Edmonds (1967) a Bock (1971). Všetky tri algoritmy sú takmer rovnaké, avšak Chu a Liu dávajú zdôvodnenie teóriou grafov, zatiaľ čo dôkazy Edmondsa a Bocka sa opierajú o teóriu lineárneho programovania. Karp (1971) dáva grafový dôkaz Edmondsovho algoritmu.

Medzi aplikácie patrí napr. návrh siete zberných kanálov. Kanály sú jednosmerné (vedené dolu svahom) a z veľkého množstva treba vybrať niekoľko tvoriacich sieť so stokom (tu ide o najlacnejšiu stokovú kostru). Edmonds dáva úlohu dopravenia najvieryhodnejšej informácie (v súčte) od riaditeľa (veliteľa) až k jednotlivým pracovníkom (vojakom a pod.). (Tu ide o najcennejšiu zdrojovú kostru s predpísaným zdrojom.)

Algoritmus

Nasledujúci algoritmus sa v základe zhoduje s postupom, ktorý predložili Chu a Liu. Ide o intuitívne prirodzený rekurentný postup, ktorého detaily budú však zrejmé až pri dôkaze.

KROK 1 (*výber najlacnejších prichádzajúcich šípov*):

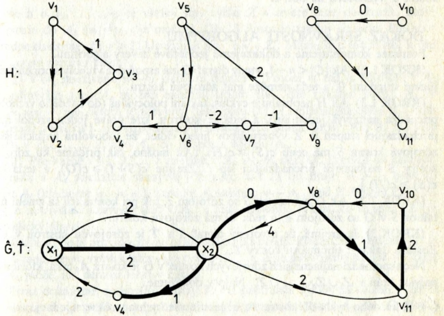
1.1: Pre každý vrchol $u \in V_G$ vyberieme jednu najlacnejšiu prichádzajúcu hranu (ak existuje). Tým sme vybrali nejakú množinu hrán Q s $|Q| \leqslant n$.

1.2: Ak $|Q| < n-1$, tak STOP: digraf G nemá zdrojovú kostru.

1.3: Utvoríme faktor H digrafu G pozostávajúci z $n-1$ najlacnejších hrán v Q. Ak H neobsahuje cyklus, tak STOP: H je najlacnejšia zdrojová kostra v G.

KROK 2 (*kontrakcie cyklov v H*):

2.1: Keďže $\deg_H^-(u) \leqslant 1$ pre každý vrchol u, tak každé dva cykly v H sú vrcholovo-disjunktné. Nech $Z_1, \ldots, Z_s$ sú všetky cykly v H, nech $Z_t = u_{t1}u_{t2} \ldots u_{tp}u_{t1}$ a $e_t^{\max}$ nech je najdrahšia hrana v Z_t ($t = 1, 2, \ldots, s$). Každý cyklus Z_t stiahneme na jediný vrchol x_t. Pritom každý šíp $vw \in E(G)$ sa zmení na šíp $\hat{v}\hat{w}$ s novou cenou takto: Ak v a w patria do toho istého cyklu Z_t, tak hrana $\hat{v}\hat{w}$ nebude existovať. V ostatných prípadoch $\hat{v}\hat{w}$ existuje a kladieme: ak vw je nejaký šíp e_w prichádzajúci zvonku do cyklu Z_t, tak $c(\hat{e}_w) := c(e_w) - c(e'_w) + c(e_t^{\max})$, kde e'_w je šíp cyklu Z_t prichádzajúci do w: inak $c(\hat{v}\hat{w}) := c(vw)$. Týmto získame z digrafu G nejaký multidigraf $\hat{G}$ s menším počtom vrcholov



Obr. 2. Ilustrácia algoritmu na digrafe z obr. 1.

a hrán ako má G, a preto môžeme predpokladať, že v ňom vieme nájsť najlacnejšiu zdrojovú kostru $\hat{T}$ (ak existuje).

2.2: Ak $\hat{G}$ nemá zdrojovú kostru, tak ju nemá ani G a STOP.

KROK 3 (*roztiahnutie kostry* $\hat{T}$): Pri návrate od $\hat{G}$ ku G roztiahneme kostru $\hat{T}$ a získame kostru T digrafu G takto:

3.1: Každý vrchol x_t, ktorý vznikol kontrakciou cyklu Z_t, nahradíme:

(i) Ak x_t je zdrojom v $\hat{T}$, tak namiesto x_t dáme cestu $Z_t - e_t^{\max}$, kde $e_t^{\max}$ je nejaká najdrahšia hrana cyklu Z_t.

(ii) Nech x_t nie je zdrojom v $\hat{T}$, nech yx_t je (jediný) šíp v $\hat{T}$ prichádzajúci do x_t a nech šíp yx_t vznikol pri kontrakcii zo šípa e_w. Nech e'_w je šíp cyklu Z_t prichádzajúci do vrcholu w. Potom namiesto x_t dáme cestu $Z_t - e'_w$.

3.2: Každý šíp kostry $\hat{T}$ nahradíme šípom digrafu G, z ktorého pri kontrakcii vznikol.

STOP: T je najlacnejšou zdrojovou kostrou digrafu G. ●

Na obr. 2 ilustrujeme uvedený algoritmus pre digraf z obr. 1. Výsledkom je kostra T vyznačená na obr. 1.

DÔKAZ SPRÁVNOSTI ALGORITMU

Postupne komentujeme a dokazujeme jednotlivé závery algoritmu.

(KROK 1.2) Ak $|Q| < n-1$, vtedy digraf G má aspoň dva vrcholy s prichádzajúcimi stupňami 0, a teda nemôže mať zdrojovú kostru.

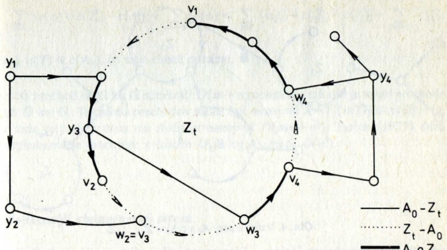
(KROK 1.3) Ak H neobsahuje cyklus, tak ani polocyklus (do každého vrcholu prichádza nanajvýš jeden šíp), a teda je kostrou, kde práve jeden vrchol má prichádzajúci stupeň 0. Z výberu šípov hneď vidieť, že ľubovoľná najlacnejšia zdrojová kostra S má cenu $c(S) \geq c(H)$. (Ak možno, tak pridáme ku zdroju kostry S najlacnejší prichádzajúci šíp f. Zrejme $c(S+f) \geq c(Q)$, a teda aj $c(S) \geq c(H)$.)

(KROK 2.2) Ak G má kostru S so zdrojom z, tak pri kontrakcii sa zmení na faktor $\hat{S}$ v $\hat{G}$ so zdrojom z, a teda $\hat{G}$ má zdrojovú kostru.

(KROK 3) Je zrejmé, že utvorený digraf T z $\hat{T}$ je zdrojovou kostrou v G. Zostáva dokázať optimalitu kostry T.

Nech spomedzi najlacnejších zdrojových kostier v G je kostra A_0 taká, ktorá má najviac hrán v $Z_1 \cup Z_2 \cup \ldots \cup Z_s$. Najprv dokážeme:

(*) Z každého cyklu Z_t kostra A_0 obsahuje všetky hrany okrem jednej.



Obr. 3. Príklad kostry A_0 a cyklu Z_t.

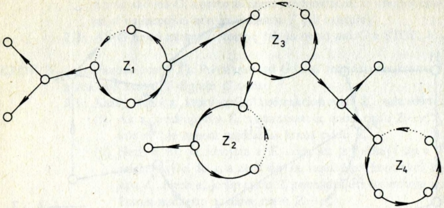
Nech $v_1w_1, \ldots, v_rw_r$ sú všetky šípy cyklu Z_t v smere jeho obiehania, ktoré nepatria do A_0 ($w_i = v_{i+1}$ je možné). Obr. 3 ilustruje takúto situáciu. Pre spor predpokladajme, že $r \geq 2$. Uvažujme šípy $v_iw_i \in Z_t$. Máme dve možnosti, ktoré postupne rozoberieme.

(1) V A_0 neexistuje w_i—v_i sled pre nejaké i. Potom w_i nie je zdrojom v A_0 a jediný šíp $y_iw_i \in A_0$ možno nahradiť šípom v_iw_i, čím získame kostru $A_1 := A_0 - y_iw_i + v_iw_i$, ktorá je zdrojová (lebo platí (1)) a najlacnejšia, lebo v_iw_i bola vybraná v kroku 1 ako najlacnejšia prichádzajúca hrana do w_i. Ale A_1 má viac hrán v $Z_1 \cup \ldots \cup Z_s$ ako A_0, čo je spor s voľbou A_0.

(2) Pre každé $i = 1, 2, \ldots, r$ existuje v A_0 w_i—v_i sled Q_i. Pretože do každého vrcholu v A_0 prichádza nanajvýš jeden šíp, tak Q_i obsahuje w_{i-1}—v_i úsek zo $Z_t \cap A_0$ (kladieme $w_0 := w_r$), a teda v A_0 existuje w_i—w_{i-1} sled P_i. Potom $P_rP_{r-1} \ldots P_1$ je netriviálny uzavretý sled v A_0, a teda obsahuje cyklus, čo je spor (A_0 je acyklický digraf).

Keďže v oboch prípadoch sme prišli do sporu s predpokladmi, tak tvrdenie (*) je dokázané. Tento vzťah kostry A_0 a cyklov $Z_1, \ldots, Z_s$ je ilustrovaný na obr. 4 (zdroj v A_0 môže, ale nemusí ležať v cykle).

Teraz dokážeme optimalitu kostry T. Pri kontrakcii cyklov sa G zmení na $\hat{G}$ a kostra A_0 sa podľa (*) zmení na nejakú zdrojovú kostru $\hat{A}_0$ pre $\hat{G}$. Podľa predpokladu o $\hat{T}$ je $c(\hat{T}) \leq c(\hat{A}_0)$.



Obr. 4. Vzťah kostry A_0 a cyklov $Z_1, \ldots, Z_s$.

Vzťah kostry T a cyklov $Z_1, \ldots, Z_s$ je podobný ako pri A_0 (pozri obr. 4), lebo T sme tak konštruovali v kroku 3. Preto v T prichádza do každého cyklu Z_t jedna alebo žiadna hrana: nech je to e_t ($t \in S := \{1, 2, \ldots, s\}$). Ďalej, nech e'_t označuje hranu cyklu Z_t prichádzajúcu do toho istého vrcholu ako hrana e_t: ak e_t neexistuje (také t môže byť nanajvýš jedno), tak kladieme: $e'_t := e_t^{\max}$, čo je najdrahšia hrana v Z_t a $c(e_t) := 0$.

Nech pri prechode od G ku $\hat{G}$ prejde hrana e_t na hranu $\hat{e}_t$. Vieme, že $c(\hat{e}_t) = c(e_t) - c(e'_t) + c(e_t^{\max})$ (podľa kroku 2). Kostra T obsahuje teda hrany e_t ($t \in S$) a nejaké ďalšie hrany $e_i \notin Z_1 \cup \ldots \cup Z_t$ pre $i \in I - S$, kde I je vhodná množina indexov. Pre kostru A_0 analogicky definujeme hrany f_t, f'_t, $f_t^{\max} := e_t^{\max}$ pre $t \in S$ a f_j pre $j \in J - S$. Potom vzťah $c(\hat{T}) \leq c(\hat{A}_0)$ môžeme písať takto:

$$\sum_{i \in I} c(\hat{e}_i) \leq \sum_{j \in J} c(\hat{f}_j), \text{ t. j.}$$

$$\sum_{i \in S} [c(e_i) - c(e'_i) + c(e_i^{\max})] + \sum_{i \in I-S} c(e_i) \leq \sum_{j \in S} [c(f_j) - c(f'_j) + c(f_j^{\max})] + \sum_{j \in J-S} c(f_j).$$

Ak ku každej strane pripočítame výraz

$$\sum_{i \in S} [c(Z_i) - c(e_i^{\max})],$$

získame:

$$\sum_{i \in S} [c(e_i) + c(Z_i) - c(e'_i)] + \sum_{i \in I-S} c(e_i) \leq \sum_{j \in S} [c(f_j) + c(Z_j) - c(f'_j)] + \sum_{j \in J-S} c(f_j),$$

t. j. $c(T) \leq c(A_0)$, čo sme chceli dokázať. ■

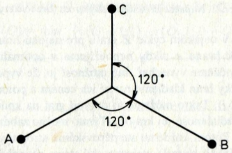
Na prechod od G ku $\hat{G}$ nám stačí $O(m+n)$ operácií a tak isto je to pri prechode od $\hat{G}$ ku G. Takýchto prechodov môže byť nanajvýš $n-1$ ($n(\hat{G}) \leq n(G) - 1$), a teda celý algoritmus má zložitosť nanajvýš $O(mn + n^2)$. Tarjan (1977) dáva implementácie pracujúce v časoch $O(m \log n)$, resp. $O(n^2)$.

5. G
Najlacnejší steinerovský strom

Začiatkom 19. storočia J. Steiner postavil a vyriešil nasledujúci problém: Pre dané 3 body A, B, C v rovine treba nájsť spojovaciu sieť najkratšej možnej celkovej dĺžky (pozri obr. 1). Zovšeobecnenie tohto problému pre viac bodov študovali Jarník a Kössler (1934) a teraz je známy ako Steinerov problém v rovine (resp. všeobecnejšie: v n-rozmernom euklidovskom priestore). Každá optimálna spojovacia sieť má tvar „stromu", a preto sa často hovorí o steinerovskom strome.

Iná verzia tohto problému je známa pod názvom rektilineárny Steinerov problém. Ide o úlohu v rovine, kde môžeme používať iba horizontálne a vertikálne čiary a navrhol ju Hanan (1966) v súvislosti s elektronickými súčiastkami (tlačené obvody).

„Najdiskrétnejšiu" verziu, Steinerov problém v grafoch, ktorému sa tu



Obr. 1. Steinerov problém v rovine a jeho riešenie.