865.984

## ERRÁTA

| Strana | Riadok | Chybne | Správne |
|---|---|---|---|
| 48 | 24 | $[\log_2(c^+ - c^-)]$ | $[\log_2(c^+ - c^-)]$ |
| 54 | 17, 18 | $\min\{z_1 + 2z_2 + 4z_3 + \ldots$ | $\min\{z_0 + 2z_1 + 4z_2 + \ldots$ |
| | | $\Sigma a_i x_i + y_i + 2y_2 + \ldots$ | $\Sigma a_i x_i + y_0 + 2y_1 + \ldots$ |
| 83 | 10 | $m(Q_1')$ | $m(Q_1)$ |
| 123 | 19 | $p_{ij}^{(k)} = p_{ij}^{(k-1)}$ | $p_{ij}^{(k)} := p_{ij}^{(k-1)}$ |
| | | $p_{ij}^{(k)} = p_{kj}^{(k-1)}$ | $p_{ij}^{(k)} := p_{kj}^{(k-1)}$ |
| 197 | 30 | $T' = T + s_i t_{j^-} - s_i t_{j^-}$ | $T' := T + s_i t_{j^-} - s_i t_{j^-}$ |
| 206 | 24 | $\hat{d}_i := d_i + \lvert E(\bar{G})\rvert (i = 1, 2)$ | $\hat{d}_i := d_i + \lvert E(\bar{G})\rvert (i = 1, 2)$ |
| 259 | 33 | $\lfloor r_i/2 \rfloor$ a $r_i$ | $\lceil r_i/2 \rceil$ a $r_i$ |
| | 36 | $\lfloor n/2 \rfloor, n$ | $\lceil n/2 \rceil, n$ |
| 326 | 18 | Plesník, J. (1982a) | Plesník, J. (1982) |
| | 20 | Plesník, J. (1982) | Plesník, J. (1982a) |

**Prosíme čitateľa, aby si opravy urobil ihneď, lebo viaceré z nich sú závažné.**

ERRÁTA

Časť KROK 2 algoritmu na strane 118 má byť správne:

KROK 2: Pre každé $j \in \bar{S} \cap V^+(k)$ urobíme:
[ Ak $d_j > d_k + c_{kj}$, tak $(p_j, d_j) := (k, d_k + c_{kj})$ a inak značku nemeníme.
Ideme na krok 1. ●

Časť KROK 1 algoritmu v hornej časti strany 120 má byť správne:

KROK 1: ($k$-tá iterácia): Ak $k = n - 1$, alebo ak $a_i \neq k$, pre každé $i \in V_G$, STOP.
Inak, pre každé $i \in V_G$ také, že $a_i = k$, urobíme:
[ Pre každé $j \in V^+(i)$ urobíme:
[ [ Ak $d_j > d_i + c_{ij}$, tak urobíme:
[ $d_j := d_i + c_{ij}$,
$p_j := i$,
$a_j := a_i + 1$,
$k := k + 1$ a ideme na krok 1. ●

Časť 1(c) algoritmu na strane 248 má byť správne:

1(c) (*prieskum vrcholu i so značkou S*):
Pre každú nepáriacu hranu $ij$ urobíme:
[ Ak vrchol $j$ má značku $S$, tak podľa druhých zložiek značiek nájdeme korene pre $i$ a $j$. Ak majú rôzne korene, tak ideme na krok 2. Ak majú spoločný koreň, tak sme objavili kvet, ktorý stiahneme na jediný vrchol so značkou bázy kvetu: považujeme ho za nepreskúmaný a ideme na 1(b).
Ak vrchol $j$ má značku $T$, tak ju ponecháme a ak nemá žiadnu značku, tak mu dáme „$T{:}i$“.
Tým je vrchol $i$ preskúmaný a ideme na 1(b).

**Prosíme čitateľa, aby si opravy urobil ihneď, lebo viaceré z nich sú závažné.**

MZK – UK Brno
2610596193

MORAVSKÁ ZEMSKÁ KNIHOVNA
sign. 2-865.987

© JÁN PLESNÍK 1983

z. mjt. + přít.

# OBSAH

PREDHOVOR . . . . . 9

1 ÚVOD DO TEÓRIE GRAFOV . . . . . 12

1.A História a motivácia . . . . . 12
1.B Rôzne druhy grafov . . . . . 17
1.C Cestovanie . . . . . 24
1.D Stupne vrcholov . . . . . 28
1.E Niektoré špeciálne grafy a digrafy . . . . . 30
1.F Ohodnotené grafové štruktúry — siete . . . . . 34
1.G Zadávanie grafových štruktúr . . . . . 35
1.Z Cvičenia a poznámky . . . . . 38

2 ÚLOHY A ALGORITMY . . . . . 40

2.A O algoritmoch . . . . . 40
2.B Polynomiálne transformácie . . . . . 44
2.C NP-ťažké a NP-ekvivalentné úlohy . . . . . 49
2.D Niektoré základné NP-ťažké úlohy . . . . . 54
2.E O navrhovaní a posudzovaní algoritmov . . . . . 67
2.Z Cvičenia a poznámky . . . . . 69

3 DOSIAHNUTEĽNOSŤ . . . . . 72

3.A Prieskum grafov . . . . . 72
3.B Bloky a listy . . . . . 79
3.C Najmenšie doplnenie grafu na list alebo blok . . . . . 86
3.D Rôzne druhy súvislosti digrafov . . . . . 91
3.E Prieskum digrafov . . . . . 93
3.F Najmenšie doplnenie na silne súvislý digraf . . . . . 99
3.G Tranzitívny uzáver a redukcia . . . . . 101
3.Z Cvičenia a poznámky . . . . . 108

4 OPTIMÁLNE SLEDY . . . . . 111

4.A Príklady na problém optimálnej cesty . . . . . 111
4.B Vzdialenosti z jedného vrcholu do ostatných . . . . . 116

4.C Vzdialenosti z každého vrcholu do každého . . . . . 121
4.D Najkratší sled s ohraničeniami . . . . . 124
4.E Generovanie zoznamu najkratších sledov . . . . . 127
4.F Niektoré ďalšie optimálne sledy . . . . . 131
4.Z Cvičenia a poznámky . . . . . 135

5 STROMY A KOSTRY . . . . . 137

5.A Kostra a nezávislé polocykly . . . . . 137
5.B Najlacnejšia kostra . . . . . 139
5.C Najlacnejšia kostra s predpísaným stupňom jedného vrcholu . . . . . 144
5.D Kostra s minimálnym pomerom dvoch ohodnotení . . . . . 150
5.E Generovanie zoznamu K najlacnejších kostier . . . . . 153
5.F Najlacnejšia zdrojová kostra digrafu . . . . . 157
5.G Najlacnejší steinerovský strom . . . . . 163
5.Z Cvičenia a poznámky . . . . . 170

6 TOKY . . . . . 173

6.A Úloha o maximálnom toku . . . . . 173
6.B Polynomiálne algoritmy pre maximálny tok . . . . . 180
6.C Variácie úlohy o maximálnom toku . . . . . 184
6.D Najlacnejší maximálny tok . . . . . 192
6.E Viacproduktové toky v grafoch . . . . . 201
6.F Polynomiálne transformácie úlohy lineárneho programovania na tokové úlohy v digrafoch . . . . . 208
6.Z Cvičenia a poznámky . . . . . 215

7 MIERY SÚVISLOSTI . . . . . 219

7.A Hranová a vrcholová súvislosť . . . . . 219
7.B Hranové súvislosti medzi každými dvoma vrcholmi grafu . . . . . 224
7.C Maximálny počet disjunktných zdrojových kostier . . . . . 232
7.D Najobjemnejší systém $k$ kostier . . . . . 235
7.Z Cvičenia a poznámky . . . . . 238

8 PÁRENIA . . . . . 240

8.A Alternujúce cesty . . . . . 240
8.B Hľadanie najpočetnejšieho párenia . . . . . 243
8.C Hľadanie najcennejšieho párenia . . . . . 249
8.D Faktory s predpísanými stupňami . . . . . 256
8.Z Cvičenia a poznámky . . . . . 259



9 POCHÔDZKY . . . . . 261

9.A Eulerovské ťahy . . . . . 261
9.B Úloha čínskeho poštára . . . . . 264
9.C Úloha obchodného cestujúceho . . . . . 273
9.Z Cvičenia a poznámky . . . . . 278

10 PLANÁRNE GRAFY . . . . . 281

10.A Eulerov vzorec . . . . . 281
10.B Rôzne rovinné vnorenia . . . . . 284
10.C Testovanie planarity . . . . . 288
10.Z Cvičenia a poznámky . . . . . 294

11 IZOMORFIZMUS . . . . . 296

11.A Izomorfizmus stromov . . . . . 296
11.B Izomorfný podstrom . . . . . 298
11.C Izomorfizmus 3-súvislých planárnych grafov . . . . . 300
11.D Izomorfizmus planárnych grafov . . . . . 306
11.Z Cvičenia a poznámky . . . . . 311

LITERATÚRA . . . . . 312
PREHĽAD OZNAČENÍ . . . . . 331
РЕЗЮМЕ . . . . . 333
SUMMARY . . . . . 335
REGISTER . . . . . 337

# PREDHOVOR

Účelom predkladanej knihy je oboznámiť čitateľa so základnými grafovými algoritmami, a tak aspoň trochu prispieť k doterajšej československej literatúre v tomto smere.

V Československu má teória grafov bohatú tradíciu a existujú viaceré knižné publikácie, napr. [Sedláček (1964), (1977, 1981)], [Zelinka (1977)], [Nečas (1978)], [Nešetřil (1979)], [Bosák (1980)], [Jucovič (1981)]. V ďalších sa teória grafov aplikuje a čitateľ sa môže s grafmi oboznámiť aspoň čiastočne: [Kotzig (1961)], [Čulík, Doležal, Fiedler (1967)], [Nebeský (1969)], [Brunovský, Černý (1980)], [Fiedler (1981)] a i. (ďalšie domáce a zahraničné publikácie pozri napr. v [Bosák (1980)]). V uvedených publikáciách sa čitateľ dočíta o „čistej" teórii grafov, ale aj o mnohých aplikáciách. Známe sú jednoduché aplikácie, kedy sa grafy používajú len ako zobrazovacie prostriedky, ale aj dômyselnejšie aplikácie, kedy sa používajú tiež aj výsledky teórie grafov. (Niektorí autori [Bondy, Murty (1976)] považujú za aplikácie len tie druhé.) V spomínaných knihách sa čitateľ dozvie, že nejakú praktickú úlohu možno sformulovať v reči teórie grafov, bližší opis algoritmu pre riešenie grafovej úlohy sa však uvádza len zriedka. Výnimkou je [Čulík et. al. (1967)], kde sa autori grafovým algoritmom venujú pomerne podrobne. Odvtedy sa však objavili nové algoritmy, ktorých výklad často vyžaduje mnoho miesta, a preto sa do kníh dostávajú len sporadicky. Táto kniha by mala sčasti vyplniť túto medzeru.

Publikácia Grafové algoritmy obsahuje viaceré (podľa mienky autora) základné algoritmy a zachytáva aj časť teórie, ktorá sa v poslednom desaťročí okolo grafových algoritmov vybudovala. Hlavný dôraz sa kladie na princípy algoritmov a ich zdôvodnenie. Na viacerých miestach sa diskutuje aj o výpočtovej zložitosti. Kniha je písaná jednoducho, bez väčších nárokov na matematické znalosti, a preto by mala byť dostupná pomerne širokému okruhu čitateľov (matematikom, technikom, ekonómom a i.). Všetky potrebné grafové pojmy sú v knihe definované, takže

v tomto smere nie je potrebné obracať sa na inú literatúru. Pravda, predbežné znalosti z teórie grafov budú rozhodne užitočné. Tu však musíme čitateľa upozorniť na to, že teória grafov nemá zatiaľ ustálenú jednotnú slovenskú, resp. českú terminológiu, a to isté možno povedať napr. o ruskej, resp. anglickej grafovej terminológii. V tejto knihe vychádzame z istého smeru v anglickej terminológii. Väčšina označení má tiež pôvod v angličtine (čo čitateľovi uľahčí štúdium anglickej literatúry). Pre viaceré pojmy bolo potrebné nájsť slovenské názvy. Preto sa na súbor použitých názvov treba pozerať ako na návrh slovenskej terminológie z teórie grafov, resp. grafových algoritmov. Hoci pre komunikovanie je otázka terminológie nesporne dôležitá, treba si uvedomiť, že podstatné sú samotné pojmy (ich obsah), a preto by čitateľ mal myslieť skôr v pojmoch ako v ich názvoch alebo označeniach. Teória grafov toto umožňuje.

Čitateľ v knihe nenájde žiadne hotové algoritmy súce hneď do počítača, ale len základné myšlienky a niekedy aj zhrnutia algoritmov, na základe ktorých už čitateľ-programátor bude vedieť urobiť programy. Pravda, niektoré algoritmy si v tomto smere môžu ešte vyžiadať mnoho práce. Na niekoľkých miestach sú uvedené aj výsledky nealgoritmickej povahy, umožňujúce lepšie pochopiť problematiku. Tomuto cieľu slúžia aj cvičenia na konci každej kapitoly. Časť z nich má ilustračný charakter a čitateľ ovládajúci základný text ich ľahko zvládne. Mnohé cvičenia sú však náročnejšie, a preto by čitateľ mal počítať aj s tým, že niektoré nebude vedieť vyriešiť. Viaceré z takýchto cvičení sú obsahom vedeckých článkov (teda nie sú triviálne) a mali by čitateľa viesť k samostatnej výskumnej práci. Pri niektorých (náročnejších) cvičeniach uvádzame stručný návod, resp. odkaz na literatúru. Na uľahčenie čítania knihy je text doplnený mnohými obrázkami. Kniha, resp. jej časti, budú užitočné aj študentom vysokých škôl.

Text je rozdelený na 11 kapitol a každá kapitola ešte na niekoľko častí (A, B atď.). Posledná časť, ktorú označujeme vždy Z, obsahuje cvičenia a poznámky. Vnútri každej časti je samostatné číslovanie viet, obrázkov, vzorcov a pod. Pri odvolávkach vo vnútri časti píšeme napr. obr. 3, ale ak sa obrázok nachádza v inej časti, napr. v časti D kapitoly 6, tak napíšeme obr. 6.D.3. Na konci každého dôkazu je značka ■. Algoritmy (resp. zhrnutia algoritmov) zvyčajne začínajú hlavičkou *Algoritmus* a končia





značkou ●. Všetky závažnejšie výsledky majú odkazy na literatúru dané menom autora a rokom publikovania. V zozname literatúry je potom v hranatých zátvorkách uvedená časť knihy, v ktorej sa príslušný prameň cituje. Mnohé jednoduché výsledky patria k „folklóru", a preto tam odkazy na literatúru neuvádzame. Takéto výsledky čitateľ nájde napr. v knihách citovaných v príslušnej časti. V zozname literatúry uvádzame aj niekoľko necitovaných prameňov. Zväčša ide o pramene, ktoré sú dostupné nášmu čitateľovi a súvisia s grafovými algoritmami. Všetky knihy sú označené symbolom *. Hoci zoznam literatúry obsahuje okolo 300 prameňov, je to len malá časť existujúcej literatúry. Ďalšie pramene nájde čitateľ citované v uvedených článkoch, ale najmä v knihách. Pre obsiahlejšie a klasifikované zoznamy odkazujeme čitateľa na referátové časopisy: Mathematical Reviews, Referativnyj žurnal — matematika a Zentralblatt für Mathematik.

Na uľahčenie orientácie je na konci knihy register obsahujúci všetky dôležité pojmy. Väčšina označení je v súlade s ustálenou matematickou symbolikou. Prehľad najčastejšie používaných označení uvádzame na konci knihy.

Recenzenti knihy RNDr. Jiří Sedláček, CSc. a prof. RNDr. Štefan Znám, DrSc. a vedecký redaktor knihy RNDr. Juraj Bosák, DrSc. vyjadrili k rukopisu početné pripomienky, ktoré pomohli podstatne zlepšiť text pri jeho konečných úpravách. Patrí im moje poďakovanie.

J. Plesník

# 1
# ÚVOD DO TEÓRIE GRAFOV

Hlavným a najčastejšie používaným pojmom pre nás bude *graf*. Najprv načrieme trochu do histórie vzniku tohto pojmu. Uvedieme niekoľko jednoduchých úloh, ktoré „zdôvodňujú" existenciu grafových štruktúr. Potom uvedieme definície základných pojmov a aj niekoľko tvrdení. Tieto tvrdenia možno nájsť v každej úvodnej knihe do teórie grafov. Napriek tomu ani pokročilejší čitateľ by kapitolu nemal preskočiť, lebo naša terminológia, žiaľ, tiež prispieva k doterajšiemu terminologickému chaosu v teórii grafov a čitateľ by si ju mal osvojiť.

## 1. A
## História a motivácia

Počiatky teórie grafov sú pomerne skromné. Na rozdiel od mnohých iných matematických disciplín, ktoré vznikli z významných (väčšinou fyzikálnych) problémov, pri vzniku teórie grafov často stáli úlohy z rekreačnej matematiky (hlavolamy).

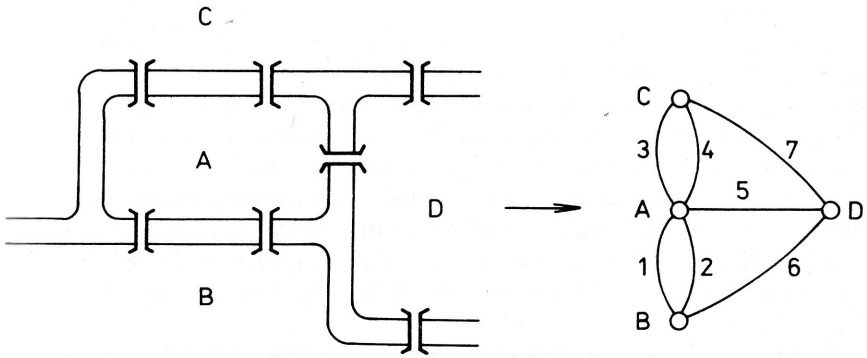
### PROBLÉM MOSTOV

Za otca teórie grafov sa pokladá vynikajúci matematik 18. storočia L. Euler, ktorý v roku 1736 vyriešil nasledujúci problém (hlavolam): Cez mesto Königsberg (dnešný Kaliningrad) tiekla rieka Pregel (dnešná Pregoľa), ktorá tam vytvárala dva ostrovy (obr. 1). Tým rozdeľovala mesto na štyri časti, ktoré však boli pospájané siedmimi mostami. Úlohou bolo navrhnúť okružnú prechádzku po meste, pri ktorej sa prejde cez všetky mosty, ale po každom len raz.

Euler (1736) vyriešil tento problém (ukázal, že takáto prechádzka neexistuje) a vytvoril aj teóriu riešiacu problémy takéhoto druhu. V jeho úvahách je implicitne obsiahnutá myšlienka zobrazenia úlohy graficky tak ako na obr. 1, kde každej časti $A$, $B$, $C$, $D$ zodpovedá jeden bod a každému mostu 1, 2, ..., 7 jedna čiara. Pritom dva takéto body sú spojené toľkými čiarami, koľko je mostov medzi príslušnými časťami. Prenechávame čitateľovi sformulovať úlohu pre pravú časť obr. 1 a dokázať, že požadovaná prechádzka neexistuje. So všeobecným výsledkom Eulera sa oboznámime až neskôr.



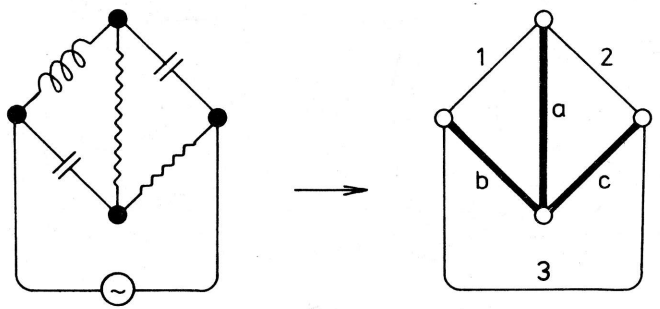





Obr. 1. Problém 7 mostov.

### ELEKTRICKÉ SIETE

Ďalší podnet, ktorý spomenieme, prišiel z fyziky. Čitateľ pravdepodobne pozná Kirchhoffove zákony pre elektrické siete. Tieto umožňujú napísať k danej sieti sústavu rovníc. Takáto sústava bude však často obsahovať „zbytočné" rovnice (lineárne závislé), keďže jeden z týchto zákonov nedáva žiadne obmedzenie na to, cez ktoré „okruhy" máme písať rovnice, a preto ich píšeme cez všetky. Kirchhoff (1847) dokázal, že stačí uvažovať len „nezávislé okruhy" a uviedol, ako ich treba hľadať. Na tento účel priradil k danej elektrickej sieti grafické vyjadrenie ignorujúce fyzikálnu povahu prvkov (obr. 2) a ukázal, že stačí v ňom nájsť nejakú „kostru" (hrubo vyznačená) a potom k tej kostre vždy pridať jednu zo zvyšných hrán a uvažovať vzniknutý „okruh". Takto sa získajú „lineárne nezávislé okruhy" a pre ne napísané rovnice budú tiež lineárne nezávislé. (Na obr. 2 dostaneme tri takéto „okruhy": $a1b$, $ac2$, $b3c$.)



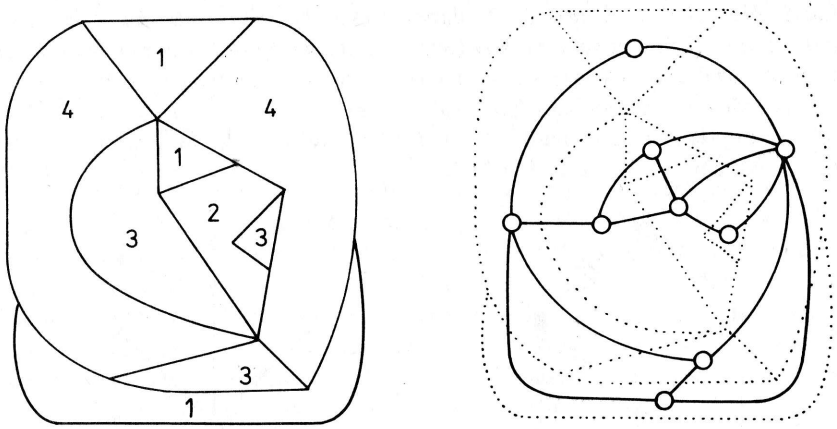

Obr. 2. Určovanie nezávislých okruhov v elektrickej sieti.

## FARBENIE MÁP

Geografické (politické) mapy sa často kvôli prehľadnosti vyfarbovali. Aký je najmenší počet farieb, ktorý vždy stačí na zafarbenie mapy v rovine, resp. na guľovej ploche? Prv než možno na otázku odpovedať, úlohu treba upresniť. Predovšetkým pôjde o také mapy, kde každý štát predstavuje súvislú oblasť. Od farbenia budeme žiadať, aby každé dva rôzne štáty mali rôznu farbu, ak majú spoločnú hranicu na nejakom úseku nenulovej dĺžky (ak majú iba konečný počet bodov spoločných, tak môžu mať tú istú farbu). Tento problém vznikol okolo r. 1850, odkedy sa mnohí pokúšali dokázať tzv. ***hypotézu o štyroch farbách***, v ktorej sa hovorí, že štyri farby nám vždy stačia. Obr. 3 ukazuje príklad, kedy sú štyri farby potrebné.

K danej mape možno zostrojiť inú grafickú reprezentáciu tak, že v každom štáte zvolíme jeden bod a dva rôzne body spojíme čiarou práve vtedy, ak príslušné štáty majú spoločný úsek hranice. Tak je to urobené na obr. 3. V tejto reprezentácii sa stačí zaoberať farbením bodov: pritom dva body spojené čiarou musia mať rôzne farby.

Tento problém je veľmi slávny, stimuloval veľkú časť teórie grafov a podarilo sa ho vyriešiť len nedávno (prvé oznámenie bolo [Appel, Haken (1976)]). Bližšie o dôkaze pozri napr. [Bosák (1979)], alebo v [Behzad et al. (1979)]. V našej knihe sa problémami farbenia takmer nezaoberáme.



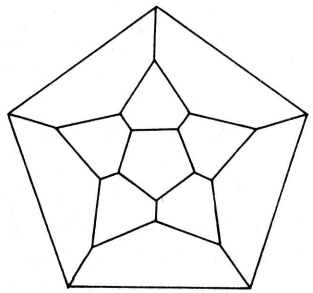

Obr. 3. K farbeniu máp.





## HAMILTONOVA HRA

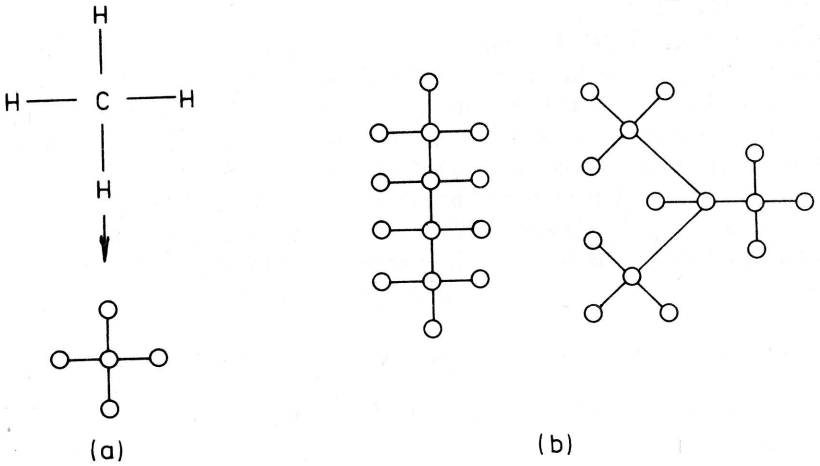
Írsky matematik R. W. Hamilton v súvislosti s objavom nekomutatívnych algebier vynašiel v r. 1859 hru na pravidelnom dvanásťstene (obr. 4). Úlohou bolo „cestovať" z vrcholu do vrcholu po hranách tohto polyédra za predpísaných podmienok. Takto vzniklo niekoľko problémov. Jeden z nich bol problém nájdenia „okružnej cesty" obsahujúcej každý vrchol práve raz. (Čitateľovi odporúčame jednu takú „okružnú cestu" nájsť.) Z tejto úlohy sa zrodil jeden smer výskumu v teórii grafov (problematika hamiltonovských grafov). Hoci vidno analógiu s problémom mostov (Euler), predsa sú takéto úlohy omnoho ťažšie.



Obr. 4. Pravidelný dvanásťsten.

## CHEMICKÉ GRAFY

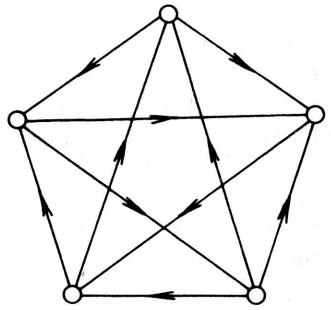
Chemické zlúčeniny sa často znázorňujú graficky. K danému „sumárnemu vzorcu" priraďujeme tzv. „štruktúrny vzorec". Tieto štruktúrne vzorce môžeme často zjednodušiť tak ako na obr. 5(a), kde je $CH_4$ (metán). Na obr. 5(b) sú dva rôzne štruktúrne vzorce pre $C_4H_{10}$. Koľko existuje štruktúrnych vzorcov k danému sumárnemu vzorcu, je vo všeobecnosti ťažká otázka a pre uhľovodíky $C_nH_{2n+2}$ sa ňou zaoberal už Cayley v r. 1874. Hoci dosiahol len čiastočný úspech, grafické zobrazovanie sa ukázalo dobrou pomôckou. Práve na tomto základe Sylvester (1878) používa prvýkrát termín *graf* v dnešnom zmysle teórie grafov.




Obr. 5. Grafy chemických zlúčenín.

## TURNAJE

Uvažujme taký športový turnaj, kde hral každý s každým a vždy bol víťaz aj porazený (napr. volejbalový turnaj). Graficky možno výsledok turnaja zobraziť napr. tak ako na obr. 6 pre 5 účastníkov turnaja. (Šípka smeruje od víťaza k porazenému.) Pomocou takýchto grafických reprezentácií možno potom zaviesť rôzne (netradičné) kritériá na celkové umiestnenie účastníkov turnaja. Tiež možno získať rôzne zaujímavosti. Jedna z nich, ktorú ukázal Rédei (1934), je takáto: Po skončení takéhoto turnaja vždy možno $n$ účastníkov usporiadať do postupnosti $a_1$, $a_2$, ..., $a_n$ tak, že $a_1$ zvíťazil nad $a_2$, $a_2$ zvíťazil nad $a_3$ atď., až $a_{n-1}$ nad $a_n$.



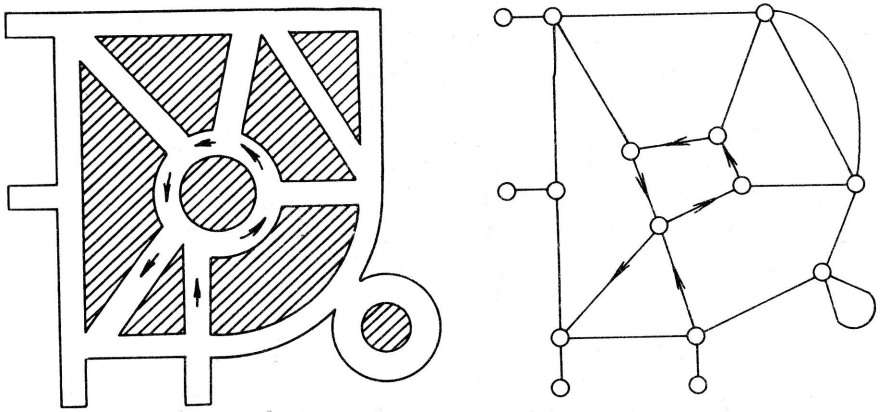
Obr. 6. Turnaj.





## CESTNÁ SIEŤ

Podobnými obrázkami možno zobrazovať aj časti cestnej siete (obr. 7). Tu vyznačíme križovatky, rázcestia, konce slepých uličiek a iné dôležité miesta, ktoré potom spájame čiarami reprezentujúcimi komunikácie, pričom šípka na čiare označuje jednosmernú komunikáciu. Takéto zobrazovanie je veľmi jednoduché a možno predpokladať, že vzniklo už v počiatkoch mapovania. Pre viaceré úlohy nám stačí uvažovať len grafické vyjadrenie. Neskoršie sa stretneme s viacerými praktickými úlohami z cestnej siete (nájsť najkratšiu cestu a pod.).



Obr. 7. Zobrazenie cestnej siete.

## PRVÁ KNIHA Z TEÓRIE GRAFOV

Definitívny vznik teórie grafov sa viaže na rok 1936, kedy maďarský matematik D. König publikoval prvú monografiu z teórie grafov. Odvtedy sa táto disciplína začína prudko rozvíjať jednak z praktických podnetov a jednak z vnútorných potrieb teórie.

„Starovek" teórie grafov, t.j. od Eulera až po Königovu knihu, je veľmi pekne opísaný v knihe [Biggs, Lloyd a Wilson (1976)].

# 1. B
# Rôzne druhy grafov

V predchádzajúcej časti sme ilustrovali vznik istých obrázkov z praktických úloh. Všetky tieto obrázky majú viaceré spoločné vlastnosti: obsahujú konečnú množinu význačných bodov (zakreslené ako krúžky), ktoré budeme nazývať



MORAVSKÁ ZEMSKÁ BRNO KNIHOVNA

*vrcholy* a konečnú množinu čiar spájajúcich niektoré vrcholy: tieto čiary budeme nazývať *hrany*. (Tieto pojmy sú prevzaté z geometrie polyédrov.) Takéto štruktúry budeme nazývať *grafy*, resp. slovo *graf* dostane ešte istú predponu.

Naše obrázky sa však líšia aj v niektorých detailoch: niektoré dvojice vrcholov sú spojené aj dvoma, resp. viacerými hranami (tzv. *rovnobežné hrany*), niekde je vrchol spojený sám so sebou (príslušná hrana je tzv. *slučka*), niektoré hrany sú bez šípiek (tzv. *rebrá*) a niektoré majú šípku (tzv. *šípy*).

Zatiaľ nie dosť presné pojmy si najprv objasníme na obrázkovej klasifikácii grafových štruktúr a potom uvedieme definíciu abstraktných „grafov".

## ZÁKLADNÁ KLASIFIKÁCIA

Niektoré štruktúry sa vyskytujú pomerne zriedkavo, a preto si slovo „graf" nechávame na najčastejšie situácie. Obr. 1 ukazuje základnú klasifikáciu. Najvšeobecnejšou grafovou štruktúrou bude pre nás *pseudomigraf*, z ktorého obmedzeniami získame rôzne menšie triedy. Dôležitú úlohu tu hrá pojem násobnej hrany. Rovnobežné hrany sa nazývajú *násobné*, ak sú buď všetky rebrá, alebo sú všetky šípy a majú ten istý začiatok aj ten istý koniec. Na obr. 2 sú rôzne druhy hrán.

*Graf* neobsahuje šípy, slučky a násobné hrany. *Multigraf* neobsahuje šípy a slučky. *Pseudograf* neobsahuje šípy.

*Digraf* (z anglického digraph = directed graph) neobsahuje rebrá, slučky a násobné hrany (*nesúhlasne rovnobežné šípy*, tzv. *protišípy*, môže obsahovať). *Multidigraf* neobsahuje rebrá a slučky. *Pseudodigraf* neobsahuje rebrá.

*Migraf* (z anglického mixed graph) neobsahuje slučky a násobné hrany. *Multimigraf* neobsahuje slučky. *Pseudomigraf* môže mať všetky vyššie zakazované hrany.

## ABSTRAKTNÉ GRAFOVÉ ŠTRUKTÚRY

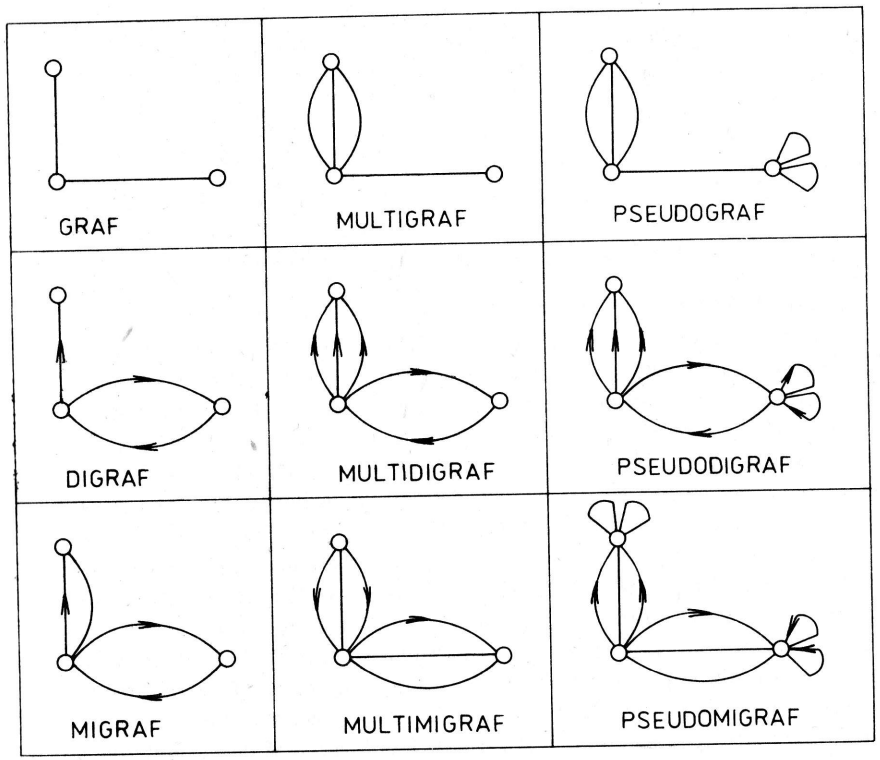
Teraz vyslovíme presné definície doteraz zavedených pojmov. Nech zápis $(a, b)$ označuje usporiadanú dvojicu prvkov, zatiaľ čo $[a, b]$ je neusporiadaná dvojica prvkov. Takto $[a, b] = [b, a]$. V oboch zápisoch môže byť aj $a = b$. Ak $A$ a $B$ sú množiny, tak píšeme $A \circ B = \{[a, b] | a \in A, b \in B\}$ a $A \times B = \{(a, b) | a \in A, b \in B\}$.

*Pseudomigraf* (alebo *grafová štruktúra*) je usporiadaná trojica $(V, E, \varphi)$, kde $V$ je konečná neprázdna množina, $E$ je konečná množina s $E \cap V = \emptyset$ a $\varphi$ je zobrazenie: $E \to (V \circ V) \cup (V \times V)$. Prvky z $V$ nazývame *vrcholy* a prvky z $E$ *hrany*. (Symboly $V$ a $E$ zodpovedajú anglickým slovám vertex — vrchol a edge — hrana.)

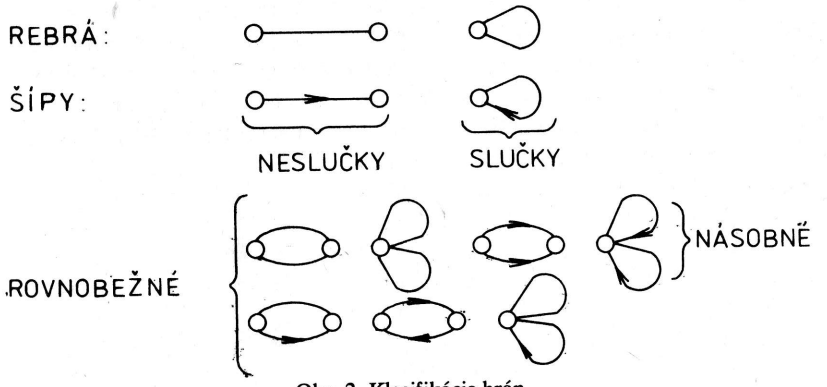
Uvedieme príklad. Nech $V = \{1, 2, 3\}$, $E = \{a, b, c, d, e\}$ a nech $\varphi(a) = [1, 2]$,





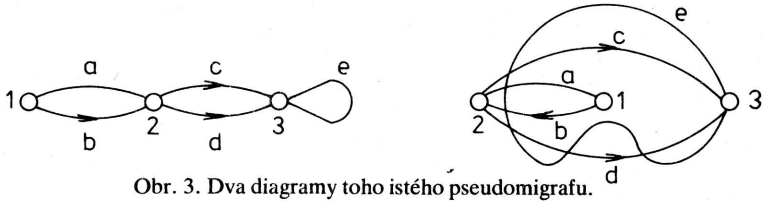



Obr. 1. Základné druhy grafových štruktúr.




Obr. 2. Klasifikácia hrán.

$\varphi(b) = (1, 2)$, $\varphi(c) = \varphi(d) = (2, 3)$ a $\varphi(e) = [3, 3]$. Tento pseudomigraf možno zobraziť tzv. *diagramom* tak ako na obr. 3, kde sú dva jeho diagramy.




Obr. 3. Dva diagramy toho istého pseudomigrafu.

## DIAGRAMY

Vo všeobecnosti (abstraktný) pseudomigraf $G$ často reprezentujeme graficky a príslušný obrázok $D$ nazývame *diagram* v rovine (na guľovej ploche, na anuloide, v 3-rozmernom priestore a pod.), ak spĺňa nasledujúce podmienky:

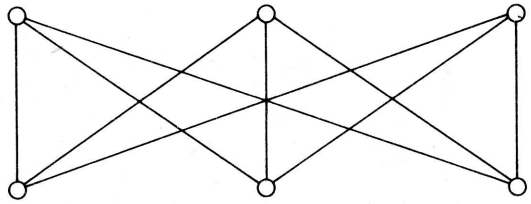
(i) každému vrcholu $x_G$ v $G$ zodpovedá bod $x_D$ v $D$ (kvôli prehľadnosti zakreslený ako krúžok, štvorček a pod.), pritom $x_D \neq y_D$, ak $x_G \neq y_G$;

(ii) každej hrane $e_G$ v $G$ zodpovedá čiara $e_D$ v $D$, pritom $e_D \neq f_D$, ak $e_G \neq f_G$;

(iii) ak $\varphi(e_G) = [x_G, y_G]$ alebo $(x_G, y_G)$, tak $e_D$ spája body $x_D$, $y_D$ a ak $\varphi(e_G) = (x_G, y_G)$, tak naviac $e_D$ je označená šípkou smerujúcou z $x_D$ do $y_D$;

(iv) každá čiara je jednoznačne identifikovateľná. (Z toho vyplýva, že čiara $e_D$ spájajúca $x_D$ a $y_D$ neobsahuje žiaden iný bod priradený vrcholu a že dve rôzne čiary môžu mať iba konečne mnoho spoločných bodov. Z toho potom vyplýva, že sa môžeme obmedziť na diagramy, kde čiara sama seba nepretína a dve rôzne čiary majú nanajvýš dva spoločné body a ak zodpovedajú nerovnobežným hranám, tak nanajvýš jeden spoločný bod.) Takto z dvoch diagramov na obr. 3 uprednostníme ľavý diagram pred pravým.

Spoločný vnútorný bod dvoch rôznych čiar nazývame *priesečníkom*. Zvyčajne uprednostňujeme diagramy s malým (najlepšie nulovým) počtom priesečníkov. Hovoríme, že pseudomigraf je *planárny*, ak existuje taký jeho diagram v rovine, ktorý nemá priesečníky, tzv. *rovinný diagram*. Žiaľ, nie vždy je to možné. Napr. graf, ktorého diagram je na obr. 4, nemá taký diagram (overte!). Planárnym grafom sa budeme venovať neskôr.



Obr. 4. Neplanárny graf.





Často namiesto diagramu grafu (digrafu atď.) hovoríme len o grafe (digrafe atď.). Tak to robíme vždy, keď nemôže dôjsť k nedorozumeniu (pozri obr. 1 a 4). Takisto na diagramoch často vynechávame označenia vrcholov a hrán.

Diagramy sú veľmi užitočné (väčšina pojmov a tvrdení v teórii grafov vznikla pozorovaním diagramov).

## ĎALŠIE POJMY

Teraz upresníme niektoré intuitívne zavedené pojmy. Hrana $e \in E$ sa nazýva *rebro*, ak $\varphi(e) \in V \circ V$, a *šíp*, ak $\varphi(e) \in V \times V$. Ak $\varphi(e) = [u, u]$, tak $e$ sa nazýva *slučkové rebro* a ak $\varphi(e) = (u, u)$, potom *slučkový šíp*; spoločný názov je *slučka*. Hrane, ktorá nie je slučkou, hovoríme *neslučka*. Namiesto rebra niekedy hovoríme *neorientovaná hrana* a šíp je potom *orientovaná hrana*. Vrcholy a hrany majú spoločný názov – *prvky* pseudomigrafu.

Nech $\varphi(e) = [u, v]$ alebo $(u, v)$. Potom $u$ a $v$ sa nazývajú *kraje (krajné vrcholy)* hrany $e$. Tiež hovoríme, že hrana $e$ *spája* vrcholy $u$ a $v$. Ďalej $u$ a $e$ (a takisto $v$ a $e$) sa nazývajú *incidentné prvky*. Hovoríme, že $u$ je *okolitý vrchol* pre $v$ a ak $u \neq v$, potom $u$ a $v$ sa nazývajú *susedné vrcholy*. Podobne dve rôzne hrany incidentné s tým istým vrcholom sa nazývajú *susedné*. Ak $e$ je šíp, tak používame aj nasledujúce upresnenia. Vrchol $u$ je *začiatok* a $v$ *koniec* hrany $e$; hrana $e$ *odchádza* z $u$ a *prichádza* do $v$; $e$ je *orientovaná* z $u$ do $v$; $v$ je *nasledovník* vrcholu $u$ a $u$ je *predchodca* vrcholu $v$.

Ak pseudomigraf $G = (A, B, \gamma)$ nebude zadaný takouto trojicou explicitne, ale inak, potom často píšeme: $A = V_G$ (alebo aj $V(G)$), $B = E_G$ (aj $E(G)$) a $\gamma = \varphi_G$; ak bude zrejmé o aké $G$ ide, tak budeme písať len $V$, $E$ a $\varphi$.

Pre pseudomigraf $G$ a jeho vrchol $v$ označujeme: $V_G^-(v)$ je množina všetkých predchodcov vrcholu $v$ alebo *prichádzajúce vrcholové okolie*; $V_G^+(v)$ je množina všetkých nasledovníkov vrcholu $v$ alebo *odchádzajúce vrcholové okolie*; $V_G(v)$ je množina všetkých okolitých vrcholov vrcholu $v$, stručne len *vrcholové okolie vrcholu* $v$. Množinu $V_G^0(v) = \{u \in V_G |$ existuje rebro spájajúce $u$ a $v\}$, nazývame *neutrálne vrcholové okolie* vrcholu $v$. Podobne $E_G^-(v)$, resp. $E_G^+(v)$ označujú množiny všetkých šípov prichádzajúcich do $v$, resp. odchádzajúcich z $v$; $E_G^0(v)$ označuje množinu všetkých rebier incidentných s $v$ a $E_G(v)$ množinu všetkých hrán incidentných s vrcholom $v$. Príslušné názvy sú *prichádzajúce šípové, odchádzajúce šípové, rebrové* a *hranové okolie* vrcholu $v$.

Dve alebo viaceré hrany $e_1, e_2, \ldots, e_k$ sa nazývajú *rovnobežné (paralelné)*, ak každá z nich má tie isté kraje a *násobné*, ak $\varphi(e_1) = \varphi(e_2) = \ldots = \varphi(e_k)$.

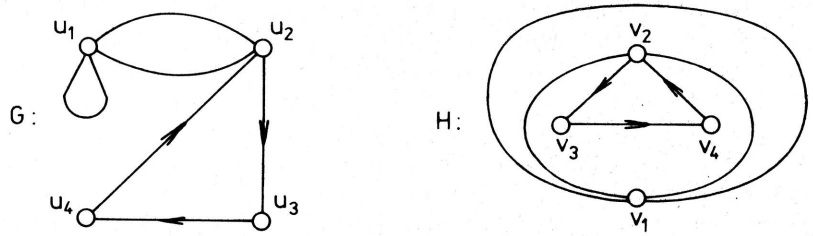
Nech $G = (V, E, \varphi)$ je pseudomigraf a $P_G = (V \circ V) \cup (V \times V)$. Pre dané $y \in P_G$ číslo $|\{e \in E | \varphi(e) = y\}|$ nazývame *násobnosťou dvojice* $y$ (a niekedy tiež *násobnosťou hrany* $f \in E$, ak $\varphi(f) = y$). Teraz už môžeme presne definovať rôzne druhy

grafových štruktúr (grafy, digrafy atď.). Keďže definície zostávajú nezmenené, potom to nebudeme znova robiť.

Nech $G$ je pseudomigraf. Ak v $G$ každé rebro *zorientujeme,* t. j. nahradíme šípom s tými istými krajmi, potom získaný pseudodigraf sa nazýva *orientácia pseudomigrafu* $G$. Na druhej strane, ak v $G$ každý šíp *dezorientujeme,* t. j. nahradíme rebrom s tými istými krajmi, získaný pseudograf sa nazýva *dezorientácia pseudomigrafu* $G$.

## IZOMORFIZMUS A PODGRAFY

Hovoríme, že dva pseudomigrafy $G$ a $H$ sú *izomorfné*, ak existuje bijekcia, tzv. *izomorfizmus* $f: V_G \to V_H$, zachovávajúca násobnosť dvojíc (t. j. násobnosť dvojice $y \in P_G$ musí byť tá istá ako násobnosť dvojice $f(y) \in P_H$). Voľne povedané $G$ a $H$ sú rovnaké,okrem označenia vrcholov a hrán. (Pozri obr. 5.) Izomorfizmus $f$ sa nazýva *automorfizmus,* ak $H = G$.




Obr. 5. $G$ a $H$ sú izomorfné ($f(u_i) = v_i$).

Nech $H$ a $G$ sú pseudomigrafy. Hovoríme, že $H$ je *podgrafom pseudomigrafu* $G$ (píšeme $H \subseteq G$), ak $V_H \subseteq V_G$, $E_H \subseteq E_G$ a $\varphi_H$ je zúženie zobrazenia $\varphi_G$ na množinu $E_H$. Hovoríme, že $H$ je *vlastný podgraf,* ak $H \subseteq G$ a $H \neq G$, a píšeme $H \subset G$.

Hovoríme, že $H$ je *podgraf indukovaný množinou* (resp. *na množine) vrcholov* $S \subseteq V_G$, ak $H \subseteq G$, $V_H = S$ a $H$ je maximálny s týmito vlastnosťami (t. j. pre každé $F \subseteq G$, kde $V_F = S$ platí $F \subseteq H$). Potom píšeme $H = G(S)$.

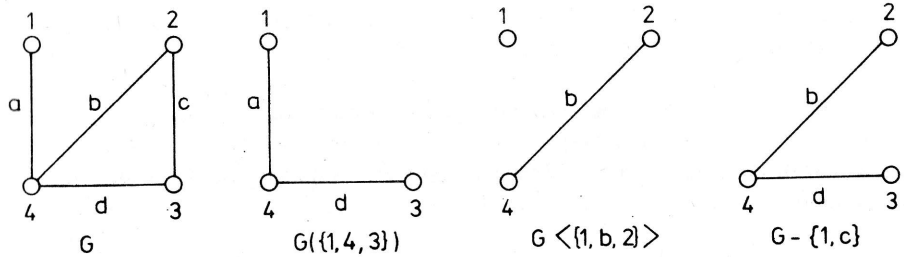
Ak $S \subseteq V \cup E$, potom $G\langle S \rangle$ označuje tzv. *obal množiny* $S$, alebo *hranovo-indukovaný podgraf na* $S$, čo je minimálny podgraf pseudomigrafu $G$ obsahujúci $S$ (t. j. žiadny vlastný podgraf pseudomigrafu $G\langle S \rangle$ už neobsahuje $S$). $H$ sa nazýva *faktor pseudomigrafu* $G$, ak $H \subseteq G$ a $V_H = V_G$.

Ak $S \subset V_G \cup E_G$, *tak* $G - S$ označuje maximálny podgraf neobsahujúci $S$. Hovoríme, že $G_1$, $G_2 \subseteq G$ sú (*vrcholovo*) *disjunktné*, ak $V(G_1) \cap V(G_2) = \emptyset$ a *hranovo disjunktné*, ak $E(G_1) \cap E(G_2) = \emptyset$.

Rôzne typy podgrafov sú ilustrované na obr. 6.








Obr. 6. Indukovaný podgraf, obal a vynechanie prvkov grafu.

## TERMINOLOGICKÉ POZNÁMKY

Z matematického hľadiska by bolo správnejšie používať termín „graf" pre pseudomigraf. Tak je to napr. v knihe [Bosák (1980)]. Potom sa používajú rôzne prívlastky pre špeciálne „grafy". Veľká časť teórie je však urobená len pre grafy a digrafy a tiež väčšina praktických úloh vyžaduje len takéto modely. Z tohto aspektu je potom výhodné používať predloženú terminológiu. Naša terminológia je blízka terminológii z kníh [Harary (1969)], [Sedláček (1977, 1981)] a [Behzad, Chartrand, Lesniak–Foster (1979)].

Pri definícii grafov sme vylúčili prípad $V_G = \emptyset$. V takomto zriedkavom prípade budeme hovoriť o *prázdnom grafe.*

Podobne, ak pripustíme, aby $V_G \cup E_G$ bola nekonečná množina, potom $G$ dostane prívlastok *„nekonečný"*. Nekonečným grafom sa prakticky nevenujeme.

V celej knihe budú mať niektoré písmená stály význam. Tak napr. kladieme počet vrcholov $n(G) = |V(G)|$ a počet hrán $m(G) = |E(G)|$ (stručne len $n$ a $m$).

Pretože graf nemá rovnobežné hrany, môžeme hranu $e$, kde $\varphi(e) = [u, v]$, stotožniť s touto dvojicou a jednoducho hovoriť o hrane $[u, v]$, čo ešte zjednodušíme a píšeme len $uv$. Analogicky v digrafe budeme hovoriť o hrane $(u, v)$, resp. $uv$. Takéto zjednodušenie používame aj v pseudomigrafoch; napr. namiesto „existuje hrana $e$ taká, že $\varphi(e) = [u, v]$ alebo $(u, v)$" píšeme „existuje rebro alebo šíp $uv$", alebo len „existuje hrana $uv$".

Naša terminológia má aj nevýhody. Jedna z nich je tá, že slovo „graf" je niekedy vhodné použiť aj v širšom význame. Napr. *teória grafov, grafové algoritmy, podgraf, úlohy na grafoch* a pod. Avšak z kontextu čitateľ ľahko zistí zamýšľaný význam.

## 1. C
## Cestovanie

Medzi najdôležitejšie pojmy v teórii grafov patria pojmy „cestovateľské", keďže sa o ne opiera veľmi mnoho výsledkov a metód. Ak vrcholy považujeme za križovatky a hrany za komunikácie, budeme cestovať tak, že každú začatú komunikáciu prejdeme vždy až do konca.

### RÔZNE SPÔSOBY CESTOVANIA

V ďalšom uvažujeme ľubovoľný pseudomigraf $G$. Podľa toho, či pri cestovaní rešpektujeme šípky alebo nie, resp. či dovolíme opakované prechádzanie prvkami, rozlišujeme niekoľko pojmov.

*Polosled* v $G$ je alternujúca postupnosť $S = v_0e_1v_1e_2 \dots e_pv_p$ ($p \geq 0$) vrcholov $v_0, \dots, v_p \in V_G$ a hrán $e_1, \dots, e_p \in E_G$, kde pre každé $i = 1, \dots, p$, $v_{i-1}$ a $v_i$ sú kraje hrany $e_i$. Hovoríme, že polosled $S$ *začína vo* $v_0$ a *končí vo* $v_p$. Tieto vrcholy sú jeho *kraje*. Číslo $p$ je jeho *dĺžka*. Stručne takýto polosled nazývame $v_0 - v_p$ *polosled*. Tie hrany polosledu $S$, ktoré sú orientované v smere polosledu (od $v_0$ ku $v_p$) alebo sú neorientované, nazývame *súhlasné hrany*, v opačnom prípade *nesúhlasné hrany polosledu*. Pokiaľ výslovne neuvedieme inak, všetky slučkové šípy v poloslede považujeme za súhlasné. *Obrátený polosled* k polosledu $S$ je polosled $S^{-1} = v_pe_p \dots v_1e_1v_0$. Ak je ešte daný $v_p - v_q$ polosled $T = v_pe_{p+1} \dots e_qv_q$, tak *zreťazenie polosledov* $S$ a $T$ (v tomto poradí) je polosled $ST = v_0e_1 \dots e_pv_pe_{p+1} \dots e_qv_q$. *Úsek polosledu* $S$ je ľubovoľná podpostupnosť po sebe idúcich prvkov z $S$, ktorá je tiež polosledom. Napr. $v_i - v_j$ úsek polosledu $S$ je polosled $v_ie_{i+1}v_{i+1}, \dots, e_jv_j$. Úsek polosledu $S$ rôzny od $S$ je *vlastný úsek*. Polosled $S$ sa nazýva *uzavretý*, ak $v_0 = v_p$, a *otvorený*, ak $v_0 \neq v_p$. Polosled $S$ sa nazýva *triviálny*, ak $p = 0$, inak *netriviálny*. Polosled, ktorý má len orientované (resp. neorientované) hrany, nazýva sa *orientovaný* (resp. *neorientovaný*).

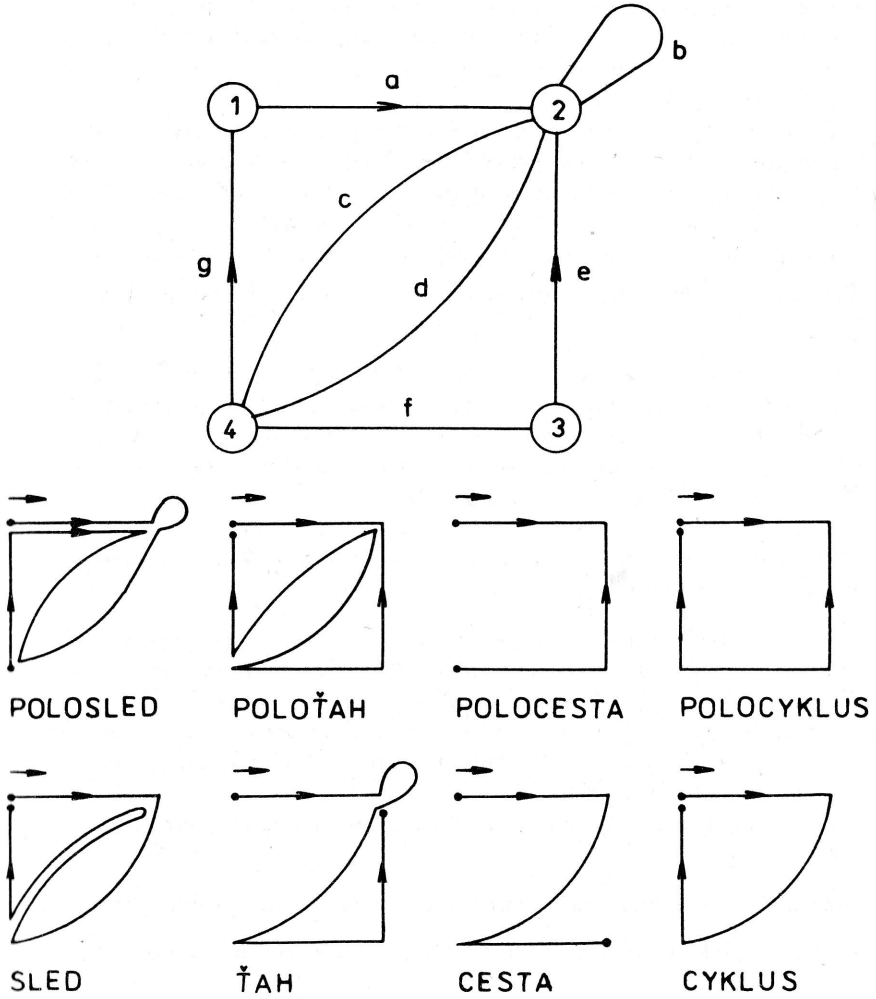
Polosled $S$ sa nazýva *poloťah*, ak sú hrany $e_1, e_2, \dots, e_p$ navzájom rôzne ($e_i \neq e_j$ pre $i \neq j$). Ak sú vrcholy $v_0, v_1, \dots, v_p$ navzájom rôzne, potom $S$ sa nazýva *polocesta*. Poloťah môže byť uzavretý aj otvorený, ale netriviálna polocesta len otvorená. Namiesto „uzavretej polocesty" máme *polocyklus*: uzavretý netriviálny poloťah, ktorého každý vlastný úsek je polocestou.

Doteraz sme cestovali bez rešpektovania šípiek (mohli sme ísť aj proti šípke). Ak dovolíme ísť po orientovanej hrane len v smere šípky, dostávame ďalšie pojmy. *Sled* je taký polosled $S = v_0e_1v_1 \dots e_pv_p$, kde pre každé $i = 1, 2, \dots, p$ je alebo $\varphi_G(e_i) = [v_{i-1}, v_i]$ alebo $\varphi_G(e_i) = (v_{i-1}, v_i)$. Podobne máme: *ťah (cesta, cyklus)* je poloťah (polocesta, polocyklus), ktorý je sledom. Príklady zavedených pojmov sú





na obr. 1. Poznamenajme, že pre pseudografy sú pojmy polosled a sled rovnocenné; preto tam vystačíme s pojmami sled, ťah, cesta a cyklus.




Obr. 1. Príklady cestovateľských pojmov: $1a2b2d4c2a1g4$ — otvorený polosled dĺžky 6, $1a2e3f4d2c4g1$ — uzavretý poloťah, $1a2d4f3$ — polocesta, $1a2e3f4g1$ — polocyklus, $1a2d4c2c4g1$ — uzavretý sled, $1a2b2d4f3e2$ — otvorený ťah, $1a2d4f3$ — cesta, $1a2d4g1$ — cyklus.

Ak $S$ je polocyklus dĺžky $p$, tak obal $G\langle S\rangle$ (myslíme tým obal prvkov polocyklu $S$) tiež nazveme *polocyklus,* prípadne *p-polocyklus* (niekedy aj *p-uholník*). Ak $S$ je cyklus dĺžky $p$, tak $G\langle S\rangle$ sa tiež nazýva *cyklus*, prípadne *p-cyklus*. (V oboch prípadoch vždy bude z kontextu zrejmé, čo máme na mysli. Ak nie, tak na tom nebude záležať.) Analogicky, ak $S$ je polocesta (resp. cesta), tak $G\langle S\rangle$ sa nazýva tiež *polocesta* (resp. *cesta*).

Poznamenajme, že niekedy polosledy zapisujeme aj stručnejšie (ak nevzniká viacznačnosť, resp. ak na tom nezáleží). Napr. namiesto celej postupnosti vypisujeme len postupnosť vrcholov alebo len postupnosť hrán. Napr. prvý 1—4 polosled z obr. 1 zapíšeme takto: 1224214, resp. takto: *abdcag*.

Polosled vo všeobecnosti nepredstavuje „ekonomické" cestovanie, platí však:

**Veta 1.** Z každého $u$—$v$ polosledu $S$ v pseudomigrafe možno vybrať $u$—$v$ polocestu.

DÔKAZ. Nech $S = w_0e_1w_1e_2 \dots e_pw_p$, kde $w_0 = u$, $w_p = v$. Ak vrcholy $w_0, w_1, \dots, w_p$ sú vzájomne rôzne, tak $S$ je polocestou. Nech $w_i = w_j$ pre nejaké $i<j$. Potom postupnosť $w_0e_1w_1 \dots e_iw_ie_{j+1}w_{j+1} \dots w_p$ je $w_0 - w_p$ polosled kratší ako $S$. Získaný polosled je alebo polocestou, alebo ho môžeme opäť skrátiť. Takto po konečnom počte skrátení získame $u$—$v$ polocestu. ■

**Veta 2.** V pseudomigrafe možno z uzavretého polosledu $S$ nepárnej dĺžky vybrať polocyklus nepárnej dĺžky.

DÔKAZ. Nech $S = w_0e_1w_1e_2 \dots e_pw_p$, $p>0$, $w_0 = w_p$. Ak $S$ nie je polocyklus, tak existujú $i, j$, $i<j$, tak, že $w_i = w_j$ a $(i, j) \neq (0, p)$. Potom $S$ hranovo rozložíme do dvoch polosledov: $S_1 = w_0e_1w_1 \dots w_ie_{j+1}w_{j+i} \dots w_n$ a $S_2 = w_ie_{i+1}w_{i+1} \dots e_jw_j$, ktoré sú uzavreté a netriviálne. Práve jeden z nich má nepárnu dĺžku, a teda môžeme o ňom uvažovať ako o $S$. Po konečnom počte takýchto rozkladov získame požadovaný polocyklus. ■

Poznamenajme, že polosled párnej dĺžky nemusí obsahovať polocyklus. (Uvážte polosled $w_0e_1w_1e_1w_0$.)

**Veta 3.** V pseudodigrafe možno z uzavretého sledu nenulovej dĺžky vybrať cyklus.

DÔKAZ je podobný predchádzajúcemu a ponechávame ho čitateľovi. ■

## SÚVISLOSŤ A KOMPONENTY

Nech $G$ je pseudomigraf. Hovoríme, že vrcholy $u$ a $v$ *súvisia*, ak v $G$ existuje $u$—$v$ polocesta. Ak každé dva vrcholy súvisia, tak $G$ sa nazýva *súvislý*. Zrejme binárna relácia „súvisieť" je ekvivalencia na $V(G)$: ak $V_1, V_2, \dots, V_c$ sú príslušné





triedy ekvivalencie, tak $G(V_1), G(V_2), \dots, G(V_c)$ sú všetky maximálne súvislé podgrafy v $G$ a nazývame ich *komponenty* (súvislosti). Pri kreslení diagramu často kreslíme komponenty oddelene (kvôli prehľadnosti). Ak $G$ má jediný komponent, tak sa nazýva *súvislý,* inak *nesúvislý.*

**Veta 4.** Ak pseudomigraf $G$ s $n$ vrcholmi a $m$ hranami je súvislý, potom $m \geq n-1$; naviac, ak neobsahuje polocyklus, tak $m = n-1$.

DÔKAZ. Nech $F$ je minimálny súvislý faktor pseudomigrafu $G$. Zrejme $F$ nemá polocyklus (inak by sme z neho mohli vynechať hranu bez porušenia súvislosti, čo by bol spor s minimalitou $F$). Teraz stačí dokázať, že $F$ má $n-1$ hrán. Pre $n=1$ je to zrejmé. Ak $n>1$, tak vynechaním ľubovoľnej hrany z $F$ získame dva komponenty $F_1, F_2$, z ktorých každý je súvislý a bez polocyklov. Preto podľa indukčného predpokladu platí $m(F_1) = n(F_1) - 1$ a $m(F_2) = n(F_2) - 1$. Keďže $m(F) = 1 + m(F_1) + m(F_2) = n(F_1) + n(F_2) - 1 = n(F) - 1$, veta je dokázaná. ■

Pseudomigraf, ktorý je súvislý a bez polocyklov, nazýva sa *strom*. Faktor pseudomigrafu, ktorý je stromom, nazýva sa *kostra* tohto *pseudomigrafu.* V predchádzajúcom dôkaze $F$ je kostrou. Tam sme ukázali, že strom má $n-1$ hrán. Čitateľ by si mal všímať význam minimálnych (resp. maximálnych) pseudomigrafov určitých vlastností, ktoré často uľahčujú dôkazy. Stromom, resp. kostrám budeme venovať jednu kapitolu.

## VZDIALENOSTI

Ak v pseudomigrafe $G$ existuje $u$—$v$ cesta, hovoríme, že vrchol $v$ je *dosiahnuteľný* z $u$. Potom symbolom $d_G(u, v)$ (stručne len $d(u, v)$) označíme dĺžku najkratšej $u$—$v$ cesty a nazveme ju *vzdialenosťou* z vrcholu $u$ do $v$, resp. vzdialenosť vrcholu $v$ od $u$. Ak žiadna $u$—$v$ cesta neexistuje, kladieme $d(u, v) = \infty$. Ak $A, B \subseteq V(G)$, definujeme ich vzdialenosť $d_G(A, B) = \min\{d(u, v) \mid u \in A, v \in B\}$. Na základe pojmu vzdialenosti definujeme ďalšie pojmy. Pre vrchol $u \in V(G)$ máme:

*odchádzajúca excentricita*

$$\mathrm{ex}^+_G(u) = \max_{v \in V(G)} d_G(u, v),$$

*prichádzajúca excentricita*

$$\mathrm{ex}^-_G(u) = \max_{v \in V(G)} d_G(v, u),$$

*excentricita*

$$\mathrm{ex}_G(u) = \max\{\mathrm{ex}^+_G(u), \mathrm{ex}^-_G(u)\}.$$

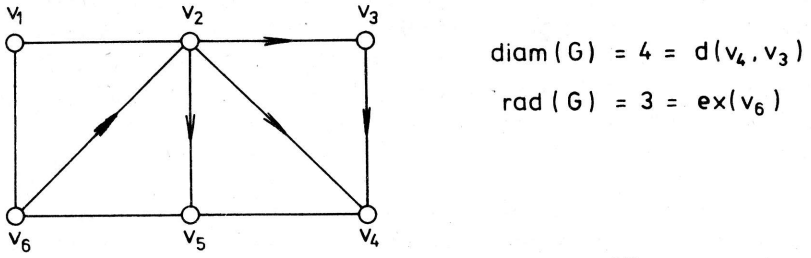
*Diameter* (alebo *priemer*) pseudomigrafu $G$ je

$$\operatorname{diam}(G) = \max_{u \in V(G)} \operatorname{ex}_G(u).$$

*Rádius* (alebo *polomer*)

$$\operatorname{rad}(G) = \min_{u \in V(G)} \operatorname{ex}_G(u).$$

Tieto metrické pojmy sú ilustrované na obr. 2.




Obr. 2. Vzdialenosť, excentricita, diameter a rádius.

**Veta 5.** Pre ľubovoľný pseudomigraf $G$ platí: $\operatorname{rad}(G) \leqslant \operatorname{diam}(G) \leqslant 2 \operatorname{rad}(G)$.

DÔKAZ. Prvá nerovnosť je zrejmá z definície. Uvažujme ľubovoľné dva vrcholy $a$, $b \in V(G)$. Nech $c$ je tzv. *centrálny vrchol,* t. j. $\operatorname{ex}(c) = \operatorname{rad}(G)$. Zreťazením $a$—$c$ cesty dĺžky $d(a, c) \leqslant \operatorname{rad}(G)$ a $c$—$b$ cesty dĺžky $d(c, b) \leqslant \operatorname{rad}(G)$, získame $a$—$b$ sled dĺžky nanajvýš $2 \operatorname{rad}(G)$. Takto $d(a, b) \leqslant 2 \operatorname{rad}(G)$, a teda aj $\operatorname{diam}(G) \leqslant 2 \operatorname{rad}(G)$. ■

## 1. D
## Stupne vrcholov

Stupne vrcholov udávajú počty hrán, ktoré sú s nimi incidentné. Presnejšie, pre vrchol $v$ pseudomigrafu $G$ definujeme niekoľko druhov stupňov. (Označenie zodpovedá anglickému degree — stupeň.) *Prichádzajúci stupeň* $\deg_G^-(v)$ je počet šípov prichádzajúcich do $v$. *Odchádzajúci stupeň* $\deg_G^+(v)$ je počet šípov odchádzajúcich z $v$. *Rebrový stupeň* $\deg_G^0(v)$ je počet rebier incidentných s $v$, pričom slučkové rebrá počítame dvakrát. *Konečne stupeň* (alebo *celkový stupeň*)

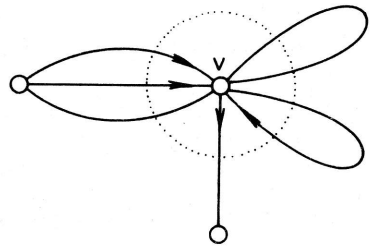
$$\deg_G(v) = \deg_G^-(v) + \deg_G^+(v) + \deg_G^0(v).$$

Obr. 1 ilustruje určovanie stupňov (bodkovaná kružnica je pomôckou).





Obr. 1. Určovanie stupňov: $\deg^-(v) = 3$, $\deg^+(v) = 2$, $\deg^0(v) = 3$, $\deg(v) = 8$.




Ak $\deg(v) = 0$, tak $v$ sa nazýva *nulový vrchol.* Graf, ktorý má len nulové vrcholy, nazýva sa *nulový graf.*

Nech $G$ je pseudomigraf a $v \in V_G$. Ak $\deg_G^-(v) = 0$ a $\deg_G^+(v) > 0$, potom $v$ sa nazýva *prameň:* ak $\deg_G^-(v) > 0$ a $\deg_G^+(v) = 0$, tak $v$ sa nazýva *ústie.* Ak $\deg_G^-(v) = \deg_G^+(v)$, tak $v$ sa nazýva *rovnovážny vrchol.* Hovoríme, že $G$ je *rovnovážny* alebo *rovnovážne orientovaný,* ak má len rovnovážne vrcholy.

Pseudomigraf sa nazýva *pravidelný stupňa k* (*k-pravidelný*), ak každý jeho vrchol má stupeň $k$. Špeciálne pre $k = 1$, 2 alebo 3 hovoríme, že pseudomigraf je *lineárny, kvadratický* alebo *kubický.*

Nasledujúce výsledky patria medzi prvé v teórii grafov vôbec [Euler (1736)]:

**Veta 1.** V ľubovoľnom pseudomigrafe $G$ platí: $\sum_{v \in V(G)} \deg_G(v) = 2m(G)$.

DÔKAZ. V súčte stupňov je každý kraj neslučky započítaný práve raz, a teda každá hrana práve dvakrát. ■

**Dôsledok 1(a).** V ľubovoľnom pseudomigrafe je počet vrcholov nepárneho stupňa párny.

DÔKAZ. Nech $A$ označuje množinu všetkých vrcholov nepárneho stupňa a nech $B = V(G) - A$. Potom podľa vety 1

$$\sum_{v \in A} \deg(v) + \sum_{v \in B} \deg(v) = 2m.$$

Pretože druhá suma je vždy párne číslo, v prvej musí byť párny počet sčítancov. ■

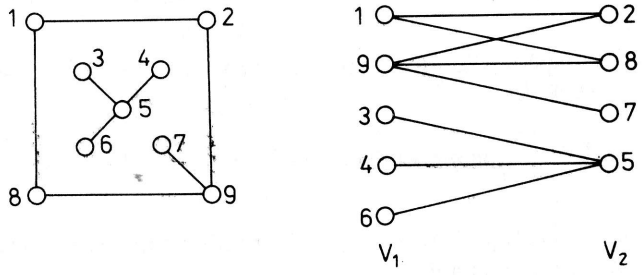
**Dôsledok 1(b).** Pravidelný pseudomigraf nepárneho stupňa má párny počet vrcholov.

DÔKAZ. Ihneď z dôsledku 1(a). ■

## 1. E
## Niektoré špeciálne grafy a digrafy

Podľa našej dohody každý graf, digraf atď. má aspoň jeden vrchol (inak hovoríme o prázdnom grafe). Graf alebo digraf sa nazýva *triviálny*, ak má jediný vrchol (a teda žiadnu hranu). *Kompletný graf* $K_n$ je graf, ktorý má $n$ vrcholov a pritom každé dva rôzne vrcholy sú susedné. Takto $K_n$ má $\frac{1}{2}n(n-1)$ hrán. Ak graf $G$ má $n$ vrcholov, tak ho možno pokladať za faktor grafu $K_n$ (uvažovaného na $V_G$). Potom graf $K_n - E_G$ označíme symbolom $\bar{G}$ a nazývame *komplement* (alebo *komplementárny graf*) grafu $G$. Špeciálne $\bar{K}_n$ je tzv. *nulový graf* (nemá žiadne hrany). *Kompletný digraf* je digraf, v ktorom pre každé dva rôzne vrcholy $u$ a $v$ existuje šíp $uv$ aj šíp $vu$.

Nech je dané číslo $k \geq 2$. Graf sa nazýva *k-partitný*, ak jeho vrcholovú množinu možno rozložiť na $k$ neprázdnych podmnožín (tzv. *partií* alebo *častí*) $V_1, V_2, \ldots, V_k$ tak, že ak $uv \in E(G)$, kde $u \in V_i$, $v \in V_j$, tak $i \neq j$ (teda žiadne dva vrcholy z tej istej partie nie sú spojené hranou). 2-partitný graf sa obvykle nazýva *bipartitný* alebo *párny graf*. Úplne rovnako sa definuje *k-partitný multimigraf*. (Na obr. 1 sú dva diagramy toho istého bipartitného grafu. V pravom diagrame je už urobený požadovaný rozklad.)




Obr. 1. Dva diagramy bipartitného grafu.

Zistiť, či je daný graf $k$-partitný, je ťažké (súvisí to so zafarbením grafu, o ktorom budeme neskoršie diskutovať). Avšak pre $k=2$ je to celkom ľahká úloha, ako vidíme z nasledujúceho dôkazu.

**Veta 1** [König (1936)]. Netriviálny graf $G$ je bipartitný $\Leftrightarrow$ $G$ neobsahuje cyklus nepárnej dĺžky.





DÔKAZ. ($\Rightarrow$) Nech $C = v_0 v_1 \ldots v_s$ je cyklus grafu $G$, kde $v_0 = v_s$. Nech $v_0 \in V_2$. Potom $v_1 \in V_1$, $v_2 \in V_2$, ..., $v_s \in V_2$, a teda $s = m(C)$ je párne číslo.

($\Leftarrow$) Nech $G$ neobsahuje cyklus nepárnej dĺžky. Zrejme stačí dokázať, že každý netriviálny komponent grafu $G$ je bipartitný. Preto predpokladajme, že $G$ je súvislý. Zvoľme ľubovoľne $v_0 \in V(G)$ a definujme rozklad množiny $V(G)$ takto:

$$V_1 = \{v \in V(G) \mid \text{existuje } v_0 - v \text{ sled nepárnej dĺžky}\},$$
$$V_2 = \{v \in V(G) \mid \text{existuje } v_0 - v \text{ sled párnej dĺžky}\}.$$

Dokážme, že $V_1$, $V_2$ sú dobre definované. Ak by existoval $v_0 - v$ sled $S_1$ nepárnej dĺžky aj $v_0 - v$ sled $S_2$ párnej dĺžky, tak $S_1 S_2^{-1}$ je uzavretý sled nepárnej dĺžky, z ktorého možno podľa vety 1.C.2 vybrať cyklus nepárnej dĺžky, čo je spor s predpokladom. Ešte ukážeme, že $V_1$, $V_2$ sú požadované partie v bipartitnom grafe. Nech $uw \in E(G)$. Ak by $u$ a $w$ patrili do toho istého $V_i$, potom existujú $v_0$—$u$ cesta $P$ a $v_0$—$w$ cesta $Q$ s rovnakou paritou dĺžok. Potom však $PuvQ^{-1}$ je uzavretý sled nepárnej dĺžky $m(P)+1+m(Q)$, čo je opäť spor s predpokladom. ■

### ACYKLICKÉ DIGRAFY

V grafe bez cyklov je každý komponent stromom, a preto sa takýto graf nazýva *les*. Analogický pojem v digrafoch je tzv. *acyklický digraf*, t. j. digraf bez cyklov. K acyklickým digrafom sa ešte vrátime. Tu uvedieme len nasledujúce základné pozorovania.

**Lema 2(a).** Ak $G$ je acyklický digraf, potom každý jeho netriviálny komponent má prameň aj ústie.

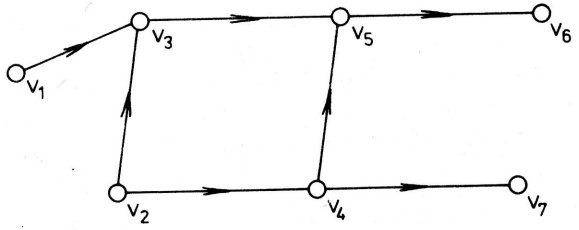
DÔKAZ. Začiatok ľubovoľnej maximálnej cesty v takom komponente je prameňom a jej koniec je ústím. ■

**Veta 2.** Digraf $G$ je acyklický $\Leftrightarrow$ vrcholy vo $V_G$ možno usporiadať do postupnosti $v_1, v_2, \ldots, v_n$ tak, že platí: $v_i v_j \in E_G \Rightarrow i < j$.

DÔKAZ. ($\Rightarrow$) Tvrdenie je zrejmé pre $n=1$. Ak $n \geq 2$, potom podľa lemy 2(a) existuje vrchol (označíme ho) $v_n$, ktorý je nulový alebo je ústím. Indukčný predpoklad zaručuje postupnosť $v_1, v_2, \ldots, v_{n-1}$ vrcholov digrafu $G - v_n$ spĺňajúcu tvrdenie vety. Potom postupnosť $v_1, v_2, \ldots, v_n$ nám zrejme vyhovuje.

($\Leftarrow$) Táto časť je zrejmá. ■

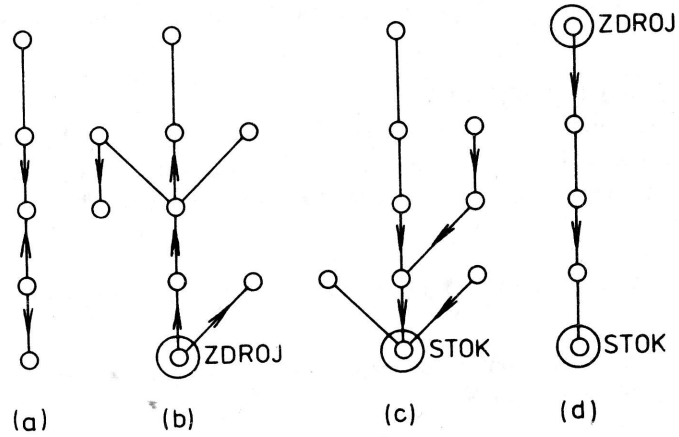
Príklad acyklického digrafu s vrcholmi označenými v súlade s vetou 2 je na obr. 2.




Obr. 2. Acyklický digraf.

## STROMY

Súvislý migraf bez polocyklov sa nazýva *strom*. Stromy patria medzi najjednoduchšie migrafy a vieme ich dobre charakterizovať. Na obr. 3 sú príklady stromov. Často však stačí stromy študovať v grafoch.




Obr. 3. Strom: (a) bez zdroja a stoku, (b) zdrojový, (c) stokový, (d) so zdrojom aj stokom.

**Veta 3.** Pre ľubovoľný graf $G$ sú nasledujúce tvrdenia ekvivalentné:
(1) $G$ je strom.
(2) Pre každé 2 vrcholy $x$, $y$ v $G$ existuje jediná $x$—$y$ cesta.
(3) $G$ je súvislý a $m = n - 1$.
(4) $G$ je bez cyklov a $m = n - 1$.





DÔKAZ. (1) $\Rightarrow$ (2) Strom je súvislý, a preto pre každé dva vrcholy $x$, $y$ existuje nejaká $x$—$y$ cesta $P$. Ak by existovala ešte $x$—$y$ cesta $Q \neq P$, potom uzavretý sled $PQ^{-1}$ by obsahoval cyklus, čo je nemožné.
(2) $\Rightarrow$ (3) Po odstránení ľubovoľnej hrany $e$ bude mať graf $G - e$ práve dva komponenty. Každý z nich bude strom, a preto tvrdenie vyplýva indukciou podľa počtu vrcholov.
(3) $\Rightarrow$ (4) Zrejme $G$ má faktor $F \subseteq G$, ktorý je súvislý a bez cyklov, teda stromom. Preto $m(F) = n - 1$ a keďže aj $m(G) = n - 1$, tak $F = G$, a teda $G$ nemá cykly.
(4) $\Rightarrow$ (1) Každý komponent $G_i$ grafu $G$ je strom, a preto $m(G_i) = n(G_i) - 1$, čo v súčte pre $k$ komponentov dáva: $m(G) = n(G) - k$. Preto musí byť $k = 1$, a teda $G$ je súvislý graf. ■

**Dôsledok 3(a).** Každý netriviálny strom má aspoň dva *koncové vrcholy* (t. j. vrcholy stupňa 1).

DÔKAZ. Inak má aspoň $n$ hrán (podľa vety 1.D.1), čo protirečí vete 3. ■

Ak uvažujeme orientáciu šípov, zavádzame ďalšie pojmy. Vrchol pseudomigrafu, z ktorého je každý vrchol dosiahnuteľný, nazýva sa *zdroj*. Duálny pojem je *stok* — vrchol dosiahnuteľný z každého vrcholu. Strom, ktorý má zdroj (resp. stok), sa nazýva *zdrojový (stokový)*; pozri obr. 3. Tieto pojmy majú prevažne význam len v digrafoch. Ak každá hrana stromu je orientovaná (resp. neorientovaná), *strom* sa nazýva *orientovaný (neorientovaný)*.

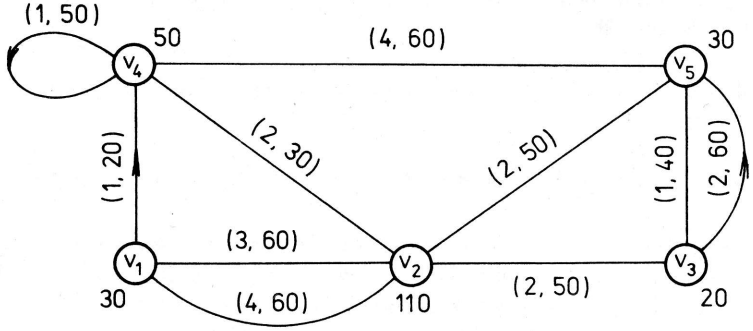
**Veta 4.** Nech $G$ je digraf a $v \in V(G)$. Potom nasledujúce tvrdenia sú ekvivalentné:
(1) $G$ je strom so zdrojom $v$.
(2) Pre každý vrchol $u$ existuje v $G$ jediný $v$—$u$ sled.
(3) Vrchol $v$ je zdroj v $G$, $\deg^-(v) = 0$ a pre každé $u \neq v$ je $\deg^-(u) = 1$.
(4) Vrchol $v$ je zdroj v $G$, ale vynechaním ľubovoľnej hrany sa táto vlastnosť poruší.
(5) $G$ je súvislý, $\deg^-(v) = 0$ a pre každý iný vrchol $u$ je $\deg^-(u) = 1$.

DÔKAZ. Prenechávame čitateľovi overiť platnosť implikácií (1) $\Rightarrow$ (2) $\Rightarrow$ (3) $\Rightarrow$ (4) $\Rightarrow$ (5) $\Rightarrow$ (1). ■

## 1. F
## Ohodnotené grafové štruktúry — siete

V aplikáciách teórie grafov sa často vyskytujú grafy sprevádzané ďalšou informáciou, ktorá nám opisuje realitu dokonalejšie ako samotný graf. Takto sa možno dívať aj na digrafy. Podstatnejšie zväčšenie informácie prináša pribranie číselných údajov na hrany, resp. vrcholy.

Ak každý vrchol $v$ pseudomigrafu $G$ má priradenú $p$-ticu reálnych čísel a každá hrana $e$ má priradenú $q$-ticu reálnych čísel, kde $p+q \geq 1$, potom hovoríme, že pseudomigraf $G$ je *ohodnotený*. Inak sa $G$ nazýva *neohodnotený pseudomigraf*.




Obr. 1. Ohodnotený pseudomigraf.

V konkrétnych situáciách ohodnotenia vrcholov a hrán nazývame rôzne; napr. *dĺžka hrany, priepustnosť vrcholu* alebo *hrany, priorita vrcholu* a pod. Na obr. 1 je príklad ohodnoteného pseudomigrafu $G$, kde $V(G) = \{v_1, v_2, v_3, v_4, v_5\}$. Tu $G$ predstavuje cestnú sieť; vrcholy sú križovatky, hrany sú ulice. Každý vrchol $v_i$ je ohodnotený jedným číslom $t(v_i)$, ktoré značí priemerný čas čakania na križovatke $v_i$. Každá hrana $e$ má dve čísla: prvé je $s(e)$, ktoré vyjadruje dĺžku ulice $e$ a druhé je $r(e)$ a označuje maximálnu povolenú rýchlosť na ulici $e$. (Napr. $t(v_4)=50$, $s([v_3, v_5])=1$ a $r((v_3, v_5)) = 60$.)

V pseudomigrafoch, kde každá hrana má predpísanú dĺžku, rozumieme pod *dĺžkou polosledu* súčet dĺžok jeho hrán (dĺžka hrany sa započítava toľkokrát, koľkokrát sa hrana v poloslede vyskytuje). Týmto zovšeobecňujeme pojem dĺžky polosledu pre neohodnotené pseudomigrafy, kde je vlastne dĺžka každej hrany rovná 1. Formálne úplne rovnako ako v časti 1.C sa definujú pojmy vzdialenosť, excentricita, diameter a i. Takto na obr. 1 je vzdialenosť $d(v_4, v_1)=5$ a $d(v_1, v_4)=1$.





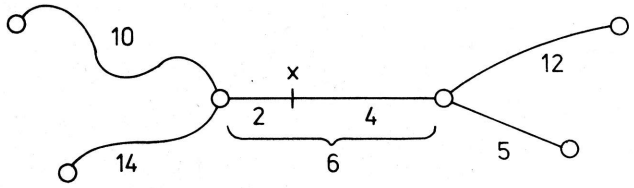
Ohodnotené pseudomigrafy sa často nazývajú *siete*. Niekedy sa stretávame aj s prívlastkami (napr. dopravná alebo transportná sieť). My však nebudeme slovo „sieť" používať v tomto zmysle.

### BODY NA HRANÁCH

V niektorých modeloch, kde sa vyskytujú ohodnotenia hrán, ktoré možno interpretovať ako dĺžky ulíc a pod., je výhodné neuvažovať graf ako abstraktnú štruktúru, ale skôr jeho diagram, pričom budú dôležité aj tie body na hranách, ktoré nie sú vrcholmi. Takto každá hrana je reprezentovaná krivkou zodpovedajúcej dĺžky. Potom má význam hovoriť o vzdialenosti ľubovoľných dvoch bodov $x$, $y$ grafu (presnejšie diagramu). Ak sa body $x$ alebo $y$ nachádzajú mimo vrcholu, prehlásime ho za vrchol (stupňa 2) a problém *vzdialenosti bodov* $x$, $y$ sa redukuje na problém vzdialenosti vrcholov $x$, $y$ v novom grafe.

Napr. pre komunikačnú sieť z obr. 2 s vyznačenými dĺžkami hrán je bod $x$ „absolútnym centrom", t. j. na ňom sa minimalizuje excentricita bodov.

Je zrejmé, že vo všeobecnosti možno takto uvažovať aj o pseudomigrafoch.




Obr. 2. Absolútne centrum komunikačnej siete.

## 1. G
## Zadávanie grafových štruktúr

Doteraz sme zadávali grafové štruktúry pomocou diagramov. Ľudia totiž uprednostňujú obrázky. Avšak tento spôsob zatiaľ nie je vhodný pre počítače. Z tohto dôvodu tu uvedieme niekoľko spôsobov pre zadávanie grafových štruktúr. Základom opäť budú grafy, resp. digrafy.

### SUSEDSKÉ MATICE

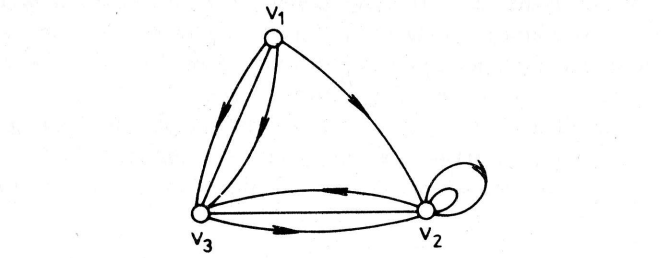
*Susedská matica A(G)* grafu $G$ s vrcholmi $v_1, v_2, \ldots, v_n$ je štvorcová matica typu $(n, n)$, ktorej prvok $a_{ij}=0$, ak $v_iv_j \notin E_G$ a $a_{ij}=1$, ak $v_iv_j \in E_G$. $A(G)$ je zrejme symetrická.

Ak $G$ je pseudograf, potom kladieme $a_{ij} = |\varphi_G^{-1}([v_i, v_j])|$.

Analogicky definujeme susedskú maticu digrafu alebo aj pseudodigrafu $G$: $a_{ij} = |\varphi_G^{-1}((v_i, v_j))|$.

Ak je daný pseudomigraf $G$, hranovo ho rozložíme na faktory $G_1$, $G_2$, pričom $G_1$ obsahuje len neorientované a $G_2$ len orientované hrany. Potom definujeme $A(G) = A(G_1) + A(G_2)$ (pozri obr. 1). Matica $A(G)$ však potom už neurčuje $G$ jednoznačne na rozdiel od $A(G_1)$ a $A(G_2)$.

Ak by sme chceli zapísať ohodnotenie hrán, tak by sme potrebovali ďalšiu maticu. Tento spôsob sa málo používa.




| | $v_1$ | $v_2$ | $v_3$ |
|---|---|---|---|
| $v_1$ | 0 | 0 | 1 |
| $v_2$ | 0 | 1 | 1 |
| $v_3$ | 1 | 1 | 0 |

$A(G_1)$

| | | | |
|---|---|---|---|
| | 0 | 1 | 2 |
| | 0 | 1 | 1 |
| | 0 | 1 | 0 |

$A(G_2)$

| | | | |
|---|---|---|---|
| | 0 | 1 | 3 |
| | 0 | 2 | 2 |
| | 1 | 2 | 0 |

$A(G)$

Obr. 1. Susedské matice.

**Veta 1.** Nech $G$ je pseudomigraf, nech $A = A(G)$ a nech $a_{ij}^{(k)}$ označuje prvok matice $A^k$ na mieste $(i, j)$. Potom v $G$ sa počet $v_i - v_j$ sledov dĺžky $k$ rovná $a_{ij}^{(k)}$ $(k = 0, 1, 2, \ldots)$.

DÔKAZ. Pre $k = 0$ je $A^k$ jednotková matica a $A^1 = A$. V oboch prípadoch je tvrdenie zrejmé. Keďže

$$a_{ij}^{(k)} = \sum_{s=1}^{n} a_{is}^{(k-1)} a_{sj}^{(1)},$$

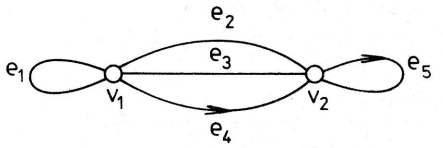
tak tvrdenie vyplýva z použitia indukcie. ■





## INCIDENČNÉ MATICE

Iný spôsob je zadať $G$ pomocou tzv. *incidenčnej matice B(G),* ktorá má $n(G)$ riadkov (zodpovedajúcich vrcholom) a $m(G)$ stĺpcov (hrany). Ak $G$ je pseudomigraf, tak prvok $b_{ij}$ incidenčnej matice $B(G)$ kladieme rovný 1, ak hrana $e_j$ je neorientovaná neslučka a inciduje s $v_i$, alebo ak $e_j$ je orientovaná neslučka odchádzajúca z $v_i$: $b_{ij} = -1$ ak $e_j$ je orientovaná a prichádza do $v_i$: $b_{ij} = 2$ ak $e_j$ je neorientovaná slučka a $b_{ij} = -2$ ak $e_j$ je orientovaná slučka pri $v_i$; v ostatných prípadoch je $b_{ij} = 0$. Na obr. 2 je príklad.




| | $e_1$ | $e_2$ | $e_3$ | $e_4$ | $e_5$ |
|---|---|---|---|---|---|
| $v_1$ | 2 | 1 | 1 | 1 | 0 |
| $v_2$ | 0 | 1 | 1 | -1 | -2 |

Obr. 2. Incidenčná matica.

Incidenčná matica má v každom stĺpci nanajvýš dva nenulové prvky a súčet absolútnych hodnôt týchto prvkov je 2. Takáto matica má však *nm* prvkov, a preto sa zriedka používa.

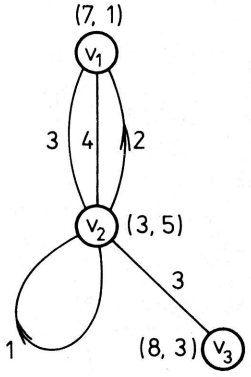
## ZOZNAMY HRÁN

Úspornejší spôsob zadávania je pomocou *zoznamov hrán*. Hrany pseudomigrafu rozdelíme na dve skupiny: rebrá a šípy. Potom napíšeme zoznam vrcholov, zoznam rebier (ako dvojice vrcholov) a zoznam šípov (usporiadané dvojice vrcholov). Ak je pseudomigraf ohodnotený, tak tu možno jednoducho pripísať aj ohodnotenia, ako ilustruje príklad na obr. 3.

Tento zápis je veľmi úsporný oproti predchádzajúcim, lebo mnohé úlohy z praxe vedú na grafové modely s pomerne malým počtom hrán.

## ZOZNAMY OKOLITÝCH VRCHOLOV

Podobný a ešte častejší spôsob zadávania pseudomigrafov je pomocou *okolí*. Pre každý vrchol $v$ sa napíšu najprv jeho ohodnotenia, potom vrcholy z neutrálneho okolia $V^0(v)$, pričom za každým $w \in V^0(v)$ napíšeme ohodnotenia rebra $vw$ a konečne vrcholy z odchádzajúceho okolia $V^+(v)$, pričom za každým $w \in V^+(v)$ napíšeme ohodnotenia šípa $vw$. Poznamenajme, že okolia $V^-(v)$ už sú týmto určené a netreba ich zadávať. Zrejme namiesto okolí $V^+(v)$ možno dať len okolia $V^-(v)$. Na obr. 3 je uvedený príklad.




| VRCHOLY | REBRÁ | ŠÍPY |
|---|---|---|
| $v_1$: 7, 1 | $v_1, v_2$ : 3 | $v_2, v_1$ : 2 |
| $v_2$: 3, 5 | $v_1, v_2$ : 4 | $v_2, v_2$ : 1 |
| $v_3$: 8, 3 | $v_2, v_3$ : 3 | |

| v | $V^o(v)$ | $V^+(v)$ |
|---|---|---|
| $v_1$: 7, 1 | $v_2$: 3; $v_2$: 4 | |
| $v_2$: 3, 5 | $v_1$ : 3; $v_1$ : 4; $v_3$ : 3 | $v_1$ : 2; $v_2$ : 1 |
| $v_3$: 8, 3 | $v_2$ : 3 | |

Obr. 3. Zoznamy hrán a zoznamy okolitých vrcholov.

## POZNÁMKY

Čitateľovi by malo byť zrejmé, ako previesť jedny dáta na iné (napr. ako zo susedskej matice vytvoriť zoznam hrán a pod.).

Existuje mnoho ďalších spôsobov, ktoré sa hodia najmä na špeciálne grafové štruktúry (napr. stromy). Vhodne zakódovaná informácia často uľahčuje spracovanie a vedie k efektívnejším algoritmom. Čitateľovi odporúčame knihu [Aho, Hopcroft a Ullman (1974)].

## 1. Z
## Cvičenia a poznámky

1. Dokážte, že každý turnaj (pozri 1.A) má tzv. *hamiltonovskú cestu* (t. j. cestu obsahujúcu všetky vrcholy) [Rédei (1934)].
2. (a) Dokážte vetu 1.C.3.
   (b) Dokážte, že ak uzavretý polosled $S$ obsahuje hranu $e$ nepárny počet ráz, tak z $S$ možno vybrať polocyklus obsahujúci $e$.
3. Hovoríme, že postupnosť nezáporných celých čísel $d_1, \ldots, d_n$ sa realizuje pseudografom, ak existuje pseudograf so stupňami vrcholov $d_1, \ldots, d_n$. Dokážte:
   (a) Postupnosť $d_1, \ldots, d_n$ sa realizuje pseudografom $\Leftrightarrow$ $d_1 + \ldots + d_n$ je párne číslo.





   (b) Postupnosť $d_1 \geq d_2 \ldots \geq d_n$ sa realizuje multigrafom $\Leftrightarrow$ $d_1 + \ldots + d_n$ je párne číslo a $d_2 + \ldots + d_n \geq d_1$ [Hakimi (1962)].
   (c) Postupnosť $d_1 \geq d_2 \geq \ldots \geq d_n$ sa realizuje grafom $\Leftrightarrow$ postupnosť $d_2 - 1$, $d_3 - 1$, ..., $d_{d_1+1} - 1$, $d_{d_1+2}$, ..., $d_n$ sa realizuje grafom [Havel (1955), Hakimi (1962)].
   (d) Postupnosť kladných čísel $d_1 \geq \ldots \geq d_n$ sa realizuje súvislým multigrafom $\Leftrightarrow$ $d_1 + \ldots + d_n$ je párne číslo, $d_2 + \ldots + d_n \geq d_1$ a $d_1 + \ldots + d_n \geq 2(n-1)$ [Hakimi (1962)].
4. Dokážte vetu 1.E.4.
5. Množina vrcholov grafu s minimálnou excentricitou sa nazýva *centrum* grafu. Dokážte, že centrum neorientovaného stromu pozostáva alebo z jediného vrcholu alebo z dvoch susedných vrcholov.
6. *s-rozmerná kocka* ($s \geq 1$) je graf s $2^s$ vrcholmi reprezentovaný všetkými možnými boolovskými vektormi (pozostávajú z 0 a 1) dĺžky $s$, pričom dva vrcholy sú spojené hranou práve vtedy, keď sa príslušné vektory líšia práve v jednej súradnici.
   Dokážte, že $s$-rozmerná kocka je pravidelný bipartitný graf stupňa $s$ a priemeru $s$. Navrhnite nejakú grafovú operáciu, pomocou ktorej možno získať z $s$-rozmernej kocky $(s+1)$-rozmernú kocku.
   Na kockách sa dobre študujú náhodné boolovské funkcie v súvislosti s pravdepodobnostnými algoritmami (napr. [Toman (1979)]).
7. Ukážte, že každý graf možno reprezentovať ako tzv. *prienikový graf* vhodného systému množín $\{S_1, \ldots, S_n\}$, kde každé $S_i$ zodpovedá vrcholu $v_i$ a dva vrcholy $v_i$, $v_j$ sú spojené hranou práve vtedy, keď $S_i \cap S_j \neq \emptyset$.
   [Návod: Za $S_i$ zvoľte $E(v_i) \cup \{v_i\}$.] Tiež ukážte, že pre súvislý graf s $n \geq 3$ možno žiadať, aby $|S_1 \cup \ldots \cup S_n| \leq m$ [Harary (1969)].
   Existuje mnoho ďalších reprezentácií grafov. Z výpočtového hľadiska je snahou nájsť úspornú reprezentáciu grafu. Ukazuje sa však, že tieto problémy patria medzi ťažké úlohy (tzv. NP-ťažké, pozri kap. 2) ([Nešetřil, Pultr (1977)], [Kučera, Nešetřil, Pultr (1980)], [Poljak, Rödl, Turzík (1981)]).

# 2
# ÚLOHY A ALGORITMY

V tejto kapitole stručne približujeme pojem algoritmu a niektoré s tým súvisiace otázky, ako je napr. zložitosť algoritmov a úloh. V matematike sa veľmi často transformuje jedna úloha na druhú. Tu nás budú zaujímať transformácie, ktoré nevyžadujú „príliš mnoho" krokov. Tieto tzv. polynomiálne transformácie ilustrujeme na viacerých úlohách, pričom sa oboznámime s úlohou lineárneho programovania a s viacerými diskrétnymi úlohami. Niektoré problémy už majú svoju históriu, ale stále sa pre ne nedarí nájsť „dobré" algoritmy. V posledných rokoch sa podarilo ukázať, že mnohé z takýchto úloh patria do istej triedy navzájom rovnocenných úloh v tom zmysle, že z existencie „dobrého" algoritmu pre jednu z nich vyplýva existencia „dobrých" algoritmov pre ostatné. Ukážeme, že mnoho kombinatorických a obzvlášť grafových úloh patrí do tejto triedy.

## 2. A
## O algoritmoch

Intuitívne pod *algoritmom* rozumieme postup, ktorý nás dovedie k riešeniu úlohy. Pritom tento postup musí byť *determinovaný* (t. j. zadaný vo forme konečného počtu jednoznačných pravidiel). Inou črtou algoritmov je ich *hromadnosť* (t. j. jedným algoritmom možno riešiť celú triedu úloh rovnakého druhu). Keďže tento postup má byť *efektívny* (t. j. má zaručiť vyriešenie úlohy po konečnom počte krokov), tak nepripúšťame v algoritme také pravidlá, ako napr.: „Ak existujú mimozemské civilizácie, tak choď domov, inak zostaň tu." (Keďže doteraz nevieme, či existujú mimozemské civilizácie, tak túto inštrukciu nemožno realizovať.)

Úlohy sa vyskytovali od nepamäti, a preto aj algoritmy na ich riešenie. Okrem takých úloh, ako uloviť medveďa, ako založiť oheň a pod., vyskytovali sa aj úlohy súvisiace s počítaním. S rozvojom počítania sa objavujú algoritmy pre sčitovanie, násobenie a pod. Snáď najznámejší algoritmus, v názve ktorého slovo *algoritmus* vystupuje, je Euklidov algoritmus pre nájdenie najväčšieho spoločného deliteľa dvoch prirodzených čísel.

S intuitívnym pojmom algoritmu sa často vystačí, avšak hlbšie skúmania vyžadujú jeho formalizáciu. V tejto súvislosti uveďme, že na začiatku 20. storočia





sa Hilbert (1926) a iní zástancovia formalizmu domnievali, že možno nájsť formálny uzavretý systém pre zapisovanie a dokazovanie všetkých matematických poznatkov. Gödel (1931) však dokázal, že to nie je možné („mechanicky dokazovať" všetky vety elementárnej aritmetiky). K dôkazu jeho „vety o neúplnosti" použil metódu diagonalizácie, založenú na starovekých paradoxoch. (Použil ju aj Cantor (1874) na dokázanie toho, že reálnych čísel je viac ako celých.)

Upresnenie pojmu „mechanické dokazovanie", viedlo k formálnej definícii algoritmu (napr. Turing (1936), Church (1936) a i.). Všetky navrhnuté definície algoritmu sa ukázali ekvivalentné, čo je obsiahnuté v tzv. *Churchovej téze*: Každý algoritmus (v neformálnom zmysle) možno reprezentovať tzv. Turingovým počítačom a obrátene.

*Turingov počítač* má obojstranne nekonečnú pásku rozdelenú na políčka a pohyblivú hlavicu, ktorá môže čítať aj zapisovať na políčko pásky jeden symbol z danej konečnej (aspoň 2-prvkovej) abecedy $S_1, \ldots, S_m$. Riadiaca jednotka počítača sa v každom časovom okamihu 1, 2, 3, ... nachádza v jednom z konečného počtu stavov $q_1, \ldots, q_n$. Vo všeobecnosti, v nejakom časovom okamihu $t$ sa hlavica nachádza na nejakom políčku $P$, kde číta symbol $S_i$ a riadiaca jednotka je v stave $q_j$. Tým je činnosť počítača určená a urobí jeden zo štyroch krokov: (1) hlavica vymazáva z políčka $P$ symbol $S_i$ a zapisuje tam nejaký symbol $S_k$, (2) hlavica sa posunie o jedno políčko vľavo, (3) hlavica sa posunie o jedno políčko vpravo, (4) počítač sa zastaví. *Program* pre Turingov počítač pozostáva z definovania jednej z uvedených štyroch možností a nového stavu riadiacej jednotky pre každú dvojicu $(S_i, q_j)$. Úlohou počítača je spracovať danú postupnosť symbolov (vstup) zapísanú na páske.

Formalizácia pojmu algoritmu sa ukázala plodná. Turing diagonalizáciou dokázal, že neexistuje algoritmus pre takýto problém: *Určiť pre daný Turingov počítač s programom a pre daný vstup, či sa počítač zastaví.* Takýchto nerozhodnuteľných problémov (presnejšie tried problémov) je známych mnoho (pozri napr. [Jones (1974)]). Pravdepodobne najkrajší výsledok dáva Matijasevič (1970), ktorý dokázal, že *problém určiť, či daná polynomická rovnica viacerých premenných s celočíselnými koeficientmi má celočíselné riešenie* (tzv. desiaty Hilbertov problém), *je algoritmicky nerozhodnuteľný.* (Každú konkrétnu rovnicu môžeme vedieť vyriešiť, ale neexistuje jediný algoritmus alebo pevný konečný počet algoritmov, ktorým by sa dala vyriešiť každá polynomická rovnica.) Z grafových problémov je algoritmicky nerozhodnuteľný napr. *problém zistiť, či pre daný graf $G_0$ existuje graf alebo nekonečný graf $G$ tak, že pre každé $v \in V_G$ je podgraf $G(V_G(v))$* (indukovaný vrcholovým okolím vrcholu $v$) *izomorfný s $G_0$* [Bulitko (1973)]. Algoritmicky nerozhodnuteľný je tak isto *problém zistiť, či pre danú konečnú množinu grafov $M$ existuje graf $G$ tak, že pre každé $v \in V_G$ je graf*

*$G(V_G(v))$ izomorfný s niektorým z grafov množiny $M$* [Bulitko (1973a)].

Pre skutočné výpočty je však Turingov počítač ťažkopádny, a preto pri štúdiu zložitosti algoritmov, resp. úloh, uvažuje sa hypotetický počítač blízky skutočným počítačom. Najčastejšie sa ako referenčný uvažuje tzv. *počítač* RAM (random access machine) [Cook, Reckhow (1973)], ktorý má tieto vlastnosti: Pamäť počítača pozostáva z nekonečného počtu buniek, očíslovaných číslami 1, 2, ..., čo sú adresy buniek. Počítač má ešte pevný konečný počet registrov. V každej bunke a v každom registri možno uložiť jedno celé číslo. (Pri úlohách s reálnymi číslami sa predpokladá, že tam možno uložiť aj reálne číslo.) V jednom kroku môže počítač (1) preniesť obsah nejakého registra do bunky, ktorej adresa je v druhom registri, alebo (2) preniesť obsah bunky, ktorej adresa je v registri, do druhého registra, alebo (3) urobiť nejakú aritmetickú operáciu na číslach v dvoch registroch, alebo (4) urobiť porovnanie čísel z dvoch registrov. Program počítača má pevnú konečnú dĺžku a udáva postupnosť operácií, ktoré sa majú urobiť. Predpokladá sa, že počiatočný obsah pamäti je vstup a konečný obsah je výstup.

V nami uvažovaných úlohách bude počítač pracovať sekvenčne. Teoreticky by sa mohol prejaviť paralelizmus tak, že by sme do jednej bunky uložili vhodným zakódovaním mnoho malých čísel tvoriacich jedno veľké číslo a jednou operáciou na veľkom čísle uskutočnili vlastne operácie na všetkých malých. Tomuto problému sa možno vyhnúť, ak predpokladáme, že čas potrebný na manipuláciu s číslom je úmerný dĺžke jeho binárnej reprezentácie [Aho et al. (1974)].

Na tomto mieste poznamenajme, že v ďalšom predpokladáme, že vstup do počítača je „rozumne zakódovaný". Týka sa to najmä celých čísel, ktoré nezapisujeme unárne, ale binárne (resp. v desiatkovej sústave a pod.). Potom na zapísanie čísla $n$ netreba $n$ symbolov, ale stačí $\lfloor \log_2 n \rfloor + 1$ binárnych symbolov (0 a 1), resp. $\lfloor \log_{10} n \rfloor + 1$ desiatkových symbolov. Celková dĺžka dát opisujúcich nejakú úlohu, sa nazýva *dĺžka vstupu* pre túto úlohu alebo jednoducho *dĺžka úlohy*.

## ZLOŽITOSŤ ALGORITMOV A PROBLÉMOV

Uviedli sme, že niektoré úlohy sú nerozhodnuteľné. Existencia algoritmu pre niektoré úlohy je celkom zrejmá, lebo pre vyriešenie stačí preskúšať všetky možnosti (tak je to v mnohých kombinatorických a grafových úlohách). Žiaľ, takých možností býva veľmi mnoho, a preto má zmysel hľadať lepší algoritmus. Ukázalo sa rozumné posudzovať algoritmy podľa počtu elementárnych operácií (času), ktorý potrebujú na vyriešenie problému. Keďže málokedy sa podarí tento počet presne vyčísliť, tak sa uspokojujeme s odhadmi. Dolné odhady sa ťažko získavajú (pozri napr. [Morávek (1969)] pre niekoľko diskrétnych problémov: novšie úspechy pozri napr. v [Aho et al. (1974)] a [Gruska (1975)].

Algoritmy a úlohy často porovnávame *asymptoticky:* Pre dve kladné funkcie





$g(x)$ a $h(x)$ píšeme $g(x) = O(h(x))$ a $h(x) = \Omega(g(x))$, ak existujú $c_0 > 0$ a $x_0$ tak, že $g(x) \leq c_0 h(x)$ pre každé $x \geq x_0$.

Takto hovoríme, že algoritmus má *zložitosť* $O(f(L))$, ak pre každé dostatočne veľké $L$ vyrieši každú úlohu dĺžky $L$ v čase $O(f(L))$. Analogicky hovoríme, že algoritmus má *zložitosť* $\Omega(f(L))$, ak pre každé číslo $c$ existuje príklad úlohy dĺžky $L \geq c$, na vyriešenie ktorej potrebuje čas $\Omega(f(L))$. Jednoducho hovoríme o $O(f(L))$ *algoritme*, resp. o $\Omega(f(L))$ *algoritme*.

Podobne možno hovoriť o úlohách: Nejaká úloha má *zložitosť* $O(f(L))$, ak pre ňu existuje $O(f(L))$ algoritmus a má *zložitosť* $\Omega(f(L))$, ak každý algoritmus pre ňu je $\Omega(f(L))$ algoritmus.

Na tomto mieste poznamenajme, že obyčajne namiesto vyššie definovanej „jemnej" dĺžky úlohy uvažujeme len „hrubú" dĺžku úlohy získanú tým, že namiesto dĺžky $\lfloor \log c \rfloor + 1$ prirodzeného čísla $c$ počítame (ako v počítači RAM) dĺžku 1. Ako príklad uvažujme *úlohu triedenia* (usporiadania do neklesajúcej postupnosti) prirodzených čísel $c_1, c_2, \ldots, c_n$. Jemná dĺžka tejto úlohy je $L = \Sigma(\lfloor \log c_i \rfloor + 1)$ a hrubá dĺžka je iba $n$. Keďže na vyhľadanie minimálneho $c_i$ stačí (v počítači RAM) $O(n)$ operácií, tak pre úlohu triedenia máme $O(n^2)$ algoritmus. Poznamenávame, že dômyselnejšie postupy viedli k vytvoreniu $O(n \log n)$ algoritmov a je ukázané, že lepší algoritmus už neexistuje, t. j. že táto úloha má zložitosť $\Omega(n \log n)$ (pozri napr. [Knuth (1973)]). Nech $K = \max(\lfloor \log c_i \rfloor + 1)$. Potom pri jemnej reprezentácii úlohy vystačíme s $O(K)$ jemnými operáciami namiesto jednej hrubej, a preto máme algoritmy zložitosti $O(K n^2) \leq O(L^3)$, resp. $O(K n \log n) \leq O(L^2 \log L)$. Aj vo všeobecnosti sa tak isto zaručí k hrubému algoritmu jemný algoritmus, a teda polynomiálnosť sa zachová. Obrátený prechod nie je vždy možný. V ďalšom budeme hovoriť len o dĺžke úlohy, ale z kontextu bude vždy zrejmé, o akú dĺžku ide.

V grafových algoritmoch budeme obyčajne uvažovať (hrubú) dĺžku úlohy $m + n$, kde $m$ je počet hrán a $n$ počet vrcholov, aj keď pôjde o ohodnotené grafové štruktúry. Toto umožňuje uvažovať funkciu $f$ (z definície zložitosti) ako funkciu dvoch premenných a jemnejšie odhadovať zložitosť. Potom sa však môže stať, že dva algoritmy (napr. zložitosti $O(m n^4)$ a $O(m^3 n)$) sú neporovnateľné (pre niektoré hodnoty $m$ a $n$ je lepší jeden a pre iné zasa druhý).

Edmonds (1965) bol prvý, ktorý si uvedomil rozdiel medzi *polynomiálnymi algoritmami* (t. j., kde za $f$ môžeme zobrať polynóm) a *nepolynomiálnymi* (ak $f$ nemôže byť polynómom). (Prvé algoritmy nazýva „dobré".) Podľa toho možno rozdeliť aj problémy na také, ktoré majú polynomiálne algoritmy a také, pre ktoré polynomiálne algoritmy neexistujú, a takto aspoň hrubo rozdeliť problémy na „zvládnuteľné" a „beznádejné". Túto ideu sa zatiaľ nepodarilo úplne realizovať, ale istý pokrok tu už je a čiastočne sa s ním oboznámime v nasledujúcich častiach.

## 2. B
## Polynomiálne transformácie

Nech je daný problém $P_1$ so vstupnými dátami $D_1$. Problém $P_1$ môžeme riešiť aj tak, že ho *pretransformujeme* na nejaký iný problém $P_2$, t. j. dáta $D_1$ prerobíme na dáta $D_2$ zodpovedajúce problému $P_2$. Ak riešenie $R_2$ problému $P_2$ vieme prerobiť na riešenie $R_1$ problému $P_1$, tak transformácia je hotová. Ak prerábanie dát $D_1$ na $D_2$ a riešenia $R_2$ na $R_1$ vyžaduje len polynomiálne mnoho elementárnych operácií vzhľadom na dĺžku dát $D_1$, tak uvedená transformácia sa nazýva *polynomiálna*. (Presnejšie, ak existuje polynóm $p$ tak, že pre každý príklad problému $P_1$ s dĺžkou vstupných dát $L_1$ možno transformáciu urobiť vykonaním nanajvýš $p(L_1)$ operácií.)

Pri pretransformovaní $P_1$ na $P_2$ stačí pre jeden príklad problému $P_1$ vyriešiť jeden príklad problému $P_2$. Ak pripustíme riešiť viac príkladov problému $P_2$, tak hovoríme o *redukcii* $P_1$ na $P_2$. (Napr. problém násobenia dvoch čísel sa redukuje na problém sčítania dvoch čísel.) Ak sumárny počet operácií je polynomiálny vzhľadom na $L_1$, tak redukcia sa nazýva *polynomiálna*. Ak $P_1$ možno polynomiálne redukovať na $P_2$ a obrátene, tak hovoríme, že $P_1$ a $P_2$ sú *polynomiálne ekvivalentné.*

V ďalšom uvedieme viaceré polynomiálne transformácie a redukcie rôznych úloh. Väčšina z týchto úloh je kvôli stručnosti formulovaná v tvare: „Treba nájsť $X$ spĺňajúce predpísané podmienky". Pod tým vždy myslíme úlohu zistiť, či také $X$ existuje a ak existuje, tak ho aj nájsť.

### ÚLOHA LINEÁRNEHO PROGRAMOVANIA

Nech sú dané (reálne) čísla $a_{ij}$, $b_i$, $c_j$ $(i = 1, \ldots, m; j = 1, \ldots, n)$. Úlohou je nájsť čísla $x_1, \ldots, x_n$, ktoré

minimalizujú $c_1x_1 + \ldots + c_nx_n$

pri *ohraničeniach*

$$\begin{aligned} a_{11}x_1 + \ldots + a_{1n}x_n &= b_1, \\ &\vdots \\ a_{m1}x_1 + \ldots + a_{mn}x_n &= b_m, \\ x_1, x_2, \ldots, x_n &\geqslant 0. \end{aligned}$$

V maticovom tvare úlohu zapíšeme stručne takto: $\min \{c^T x | Ax = b,\ x \geqslant 0\}$ (symbol T označuje transponovanie matice: vektory považujeme za stĺpcové, a preto $c^T$ je riadkový vektor).

Lineárne programovanie vzniklo v 40. rokoch nášho storočia a stalo sa významným nástrojom najmä v ekonómii. Jeho históriu a úspechy možno nájsť napr. v [Dantzig (1962)].





K uvedenej úlohe sa definuje tzv. *duálna úloha*: $\max \{b^T u | u^T A \leqslant c^T\}$. Vektor $x$ spĺňajúci ohraničenia sa nazýva *prípustné riešenie primárnej úlohy*: analogicky hovoríme o prípustnom riešení $u$ duálnej úlohy a o *optimálnych riešeniach.* $c^T x$ a $u^T b$ sú *účelové funkcie* úloh. Z teórie duality vyberáme:

**Veta 1.** Pre primárnu a duálnu úlohu platí:

(a) (*Slabá veta o dualite*): Pre ľubovoľné prípustné riešenia $x$ a $u$ je $c^T x \geqslant b^T u$.

(b) (*Silná veta o dualite*): Ak má jedna z úloh optimálne riešenie, tak má aj druhá a optimálne hodnoty účelových funkcií sa rovnajú $(c^T x^* = b^T u^*)$.

(c) (*Podmienky ortogonality*): Prípustné riešenia $x$ a $u$ sú optimálne $\Leftrightarrow$ $(u^T a_j - c_j)x_j = 0$ pre každé $j = 1, \ldots, n$, kde $a_j$ je $j$-tý stĺpec matice $A$.

Dôkaz časti (a) je bezprostredný: $u^T b \leqslant u^T A x \leqslant c^T x$. Dôkaz časti (b) je zložitejší, a tu ho neuvádzame (pozri napr. [Dantzig (1962)], [Hu (1969)]). Dôkaz časti (c) vyplýva z dôkazu časti (a) a z tvrdenia (b). ■

Poznamenajme, že existujú aj iné tvary úlohy lineárneho programovania, napr. $\min \{c^T x | Ax \geqslant b,\ x \geqslant 0\}$. Potom duálna úloha je $\max \{b^T u | u^T A \leqslant c^T,\ u \geqslant 0\}$ a platí analógia vety 1. Rôzne tvary sa ľahko prevádzajú jeden na druhý: rovnosť môžeme nahradiť dvoma nerovnosťami; nerovnosť zasa možno previesť na rovnosť pridaním nezápornej doplnkovej premennej; premennú neohraničenú na znamienko nahradíme rozdielom dvoch nezáporných premenných. Namiesto maximalizácie funkcie $b^T u$ možno tiež minimalizovať funkciu $-b^T u$ a takto možno duálnu úlohu previesť na tvar primárnej úlohy. Preto možno hovoriť, že duálna úloha k duálnej je rovná pôvodnej primárnej úlohe.

Pri konštrukcii duálnej úlohy možno však postupovať aj bez transformácie na základný tvar riadiac sa pravidlom, že rovnostiam primáru zodpovedajú v duáli premenné bez ohraničenia na znamienko a obrátene. Nerovnosti musia byť také, aby sme jednoducho vynásobením získali požadovanú nerovnosť v slabej vete o dualite ako hore.

Klasická metóda riešenia úlohy lineárneho programovania je tzv. *simplexová metóda*, ktorá predstavuje cestovanie po susedných vrcholoch polyedrickej množiny určenej ohraničeniami $Ax = b$, $x \geqslant 0$. Každý vrchol zodpovedá tzv. *bázickému riešeniu* systému $Ax = b$ a prechádza sa od jedného bázického riešenia k druhému, až sa nájde optimálne riešenie, alebo sa zistí, že neexistuje.

Z výpočtového hľadiska môžeme predpokladať, že všetky vstupné údaje $(a_{ij}, b_i, c_j)$ sú racionálne čísla, čo však možno upraviť na celé čísla. Potom simplexová metóda poskytuje racionálne riešenie, a teda takéto môžeme žiadať vždy.

Hoci v praxi je so simplexovou metódou veľká spokojnosť, boli zostrojené príklady ukazujúce jej exponenciálny charakter dokonca pre špeciálnu, tzv.

*dopravnú úlohu* [(Zadeh (1973)] (pozri 6.D). Nedávno však Chačijan (1979) našiel polynomiálny algoritmus. Hoci v praxi zatiaľ nemôže konkurovať simplexovej metóde, táto tzv. *elipsoidová metóda* poskytla niektoré nové polynomiálne algoritmy [Lovász (1980)], [Grötschel, Lovász, Schrijver (1981)], [Nešetřil, Poljak (1980)]. (Využíva sa možnosť charakterizovania niektorých úloh systémom lineárnych nerovností.)

Nasledujúci výsledok [Itai (1978)] použijeme v kap. 6.

**Veta 2.** Nasledujúce 3 úlohy sú polynomiálne ekvivalentné.

(1) *Úloha lineárneho programovania*: Nájsť racionálne riešenie úlohy $\min\{c^T x | Ax = b,\ x \geqq 0\}$, kde všetky $a_{ij}$, $b_i$ a $c_j$ sú celé čísla.

(2) *Nezáporné riešenie lineárnych rovníc:* Nájsť racionálne riešenie systému $Ax = b$, $x \geqq 0$, kde všetky $a_{ij}$ a $b_i$ sú celé čísla.

(3) *Nezáporné racionálne riešenie lineárnych rovníc s malými koeficientmi*: Úloha (2), kde všetky $a_{ij}$, $b_i \in \{-1, 0, +1\}$.

DÔKAZ. (1) pretransformujeme na (2). Podľa vety 1 je $x$ optimálnym riešením úlohy (1) práve vtedy, ak existuje optimálne riešenie $u$ duálnej úlohy $\max\{b^T u | A^T u \leqq c\}$. Potom je $c^T x = b^T u$ a môžeme žiadať, aby $u$ bol racionálny vektor. Teda získame úlohu nájsť racionálne riešenie $(x^T, u^T)$ systému

$$Ax = b,\ x \geqq 0,\ A^T u \leqq c,\ c^T x = b^T u.$$

Systém nerovností $A^T u \leqq c$ zmeníme na rovnosti zavedením doplnkových premenných $v \geqq 0$. Substitúciou $u = u' - u''$, kde $u' \geqq 0$, $u'' \geqq 0$, získame požadovaný tvar úlohy: $Ax = b$, $A^T u' - A^T u'' + v = c$, $c^T x - b^T u' + b^T u'' = 0$ $x \geqq 0$, $u' \geqq 0$, $u'' \geqq 0$, $v \geqq 0$.

Teraz (2) pretransformujeme na (3). Zrejme môžeme predpokladať, že $b \geqq 0$. Každú rovnicu systému $Ax = b$ nahradíme rovnocenným systémom rovníc, ktorý získame vyjadrením čísel v dvojkovej sústave, ako ukazuje nasledujúci príklad. Rovnicu

(i) $7x_1 - 10x_2 + 2x_3 = 12$

môžeme prepísať takto:

(ii) $(1.x_1 - 0.x_2 + 0.x_3)2^0 + (1.x_1 - 1.x_2 + 1.x_3)2^1 +$
$(1.x_1 - 0.x_2 + 0.x_3)2^2 + (0.x_1 - 1.x_2 + 0.x_3)2^3 =$
$0.2^0 + 0.2^1 + 1.2^2 + 1.2^3.$

Pomocou nových premenných $u_i$, $v_i \geqq 0$ zabezpečíme prenos z nižších jednotiek na vyššie, ako pri počítaní v dvojkovej sústave: z (ii) získame:





(iii)
$$\begin{aligned}
&1.x_1 - 0.x_2 + 0.x_3 && -2(u_1 - v_1) = 0,\\
&1.x_1 - 1.x_2 + 1.x_3 + u_1 - v_1 - 2(u_2 - v_2) = 0,\\
&1.x_1 - 0.x_2 + 0.x_3 + u_2 - v_2 - 2(u_3 - v_3) = 1,\\
&0.x_1 - 1.x_2 + 0.x_3 + u_3 - v_3 && = 1.
\end{aligned}$$

Ukážme, že systém (iii) je ekvivalentný s (ii), a tým aj s (i). Ak $x$, $u$, $v$ je riešenie systému (iii), tak vynásobme prvú rovnicu v (iii) číslom $2^0$, druhú číslom $2^1$ atď. a spočítajme. Získame práve (ii), a preto $x$ je riešením (ii).

Obrátene, ak $x$ je racionálne riešenie rovnice (ii), tak zrejme existujú také racionálne čísla $y_1, y_2, y_3, y_4$, že koeficient pri $2^p$ na ľavej strane, zväčšený o $y_{p+1}$, je rovný koeficientu pri $2^p$ na pravej strane pre každé $p = 0, 1, 2, 3$. Ďalej k číslam $y_1, \ldots, y_4$ existujú racionálne čísla $u_i$, $v_i \geqq 0$ tak, že $y_1 = -2(u_1 - v_1)$, $y_2 = (u_1 - v_1) - 2(u_2 - v_2)$, .... Pritom $y_4$ je už určené podmienkou (ii) (t. j. $y_1 + 2y_2 + 4y_3 + 8y_4 = 0$), a preto $y_4 = u_3 - v_3$. Teda systém (iii) má racionálne riešenie.

Vo všeobecnosti sa konštrukcia systému (iii) a dôkaz správnosti robí tak isto. Ak $K = 1 + \lfloor \log_2 \max\{|a_{ij}|, |b_i|\} \rfloor$, tak celkove zo systému $m$ rovníc $Ax = b$ s $n$ neznámymi vznikne systém nanajvýš $mK$ rovníc s počtom neznámych nanajvýš $n + 2(K-1)m$.

V získanom systéme však sú niektoré koeficienty 2 alebo $-2$. To možno odstrániť takto: namiesto $\pm 2x_j$ píšeme $\pm(x_j + x_j')$ a pridávame rovnicu $x_j = x_j'$. Tým získame problém (3).

Keďže uvedené transformácie sú zrejme polynomiálne a (3) je špeciálnym prípadom (1), veta je dokázaná. ■

## ÚLOHA BINÁRNEHO LINEÁRNEHO PROGRAMOVANIA

Ak v úlohe lineárneho programovania žiadame, aby všetky $x_i$ boli celé čísla, tak hovoríme o *úlohe celočíselného lineárneho programovania.* Takéto úlohy sú veľmi ťažké (pozri napr. [Hu (1969)], [Taha (1975)]). Často však vystačíme s úlohami, kde je dané číslo $K$ a žiadame, aby každé $x_i \leqq K$. Tieto úlohy možno ešte zjednodušiť:

**Veta 3.** Nasledujúce úlohy sú polynomiálne ekvivalentné:

(1) *Úloha celočíselného lineárneho programovania s ohraničenými premennými*: $\min\{c^T x | Ax = b,\ x \geqq 0,\ x_i \leqq K,\ x_i$ celé pre všetky $i\}$, kde $a_{ij}$, $b_i$, $c_j$ a $K$ sú dané celé čísla.

(2) *Úloha binárneho lineárneho programovania:* $\min\{c^T x | Ax = b,$ každé $x_i \in \{0, 1\}\}$, kde všetky čísla $a_{ij}$, $b_i$, $c_j$ sú celé.

(3) *Binárne riešenie systému rovníc:* Nájsť $x_1, \ldots, x_n \in \{0, 1\}$ tak, aby $Ax = b$, kde všetky $a_{ij}$, $b_i$ sú celé.

(4) *Binárne riešenie systému rovníc s malými koeficientmi:* Úloha (3), kde všetky $a_{ij}$, $b_i \in \{-1, 0, 1\}$.

(5) *Binárne riešenie lineárnej rovnice:* Dané sú prirodzené čísla $a_1, \ldots, a_n, b$ a treba nájsť $x_1, \ldots, x_n \in \{0, 1\}$ tak, aby $a_1x_1 + \ldots + a_nx_n = b$.

(6) *Rozdelenie multimnožiny čísel:* Dané prirodzené čísla $a_1, \ldots, a_n$ treba rozdeliť do dvoch tried s rovnakým súčtom ($\Sigma a_i/2$).

(7) *Úloha o binárnom batohu:* $\max \{c_1x_1 + \ldots + c_nx_n | a_1x_1 + \ldots + a_nx_n \leqslant b$, všetky $x_i \in \{0, 1\}\}$, kde všetky $c_j$ a tak isto všetky $a_j$ a číslo $b$ sú prirodzené čísla.

(8) *Nezáporné celočíselné riešenie lineárnej rovnice:* Dané sú prirodzené čísla $a_1, \ldots, a_n, b$: treba nájsť nezáporné celé čísla $x_1, \ldots, x_n$, tak, aby $a_1x_1 + \ldots + a_nx_n = b$.

(9) *Úloha o celočíselnom batohu:* Úloha (7), kde namiesto $x_i \in \{0, 1\}$ žiadame, aby všetky $x_i$ boli nezáporné celé čísla.

DÔKAZ. Využívame známe techniky z lineárneho a celočíselného programovania (doplnkové premenné, metóda bisekcie, agregácia rovníc: pozri napr. [Taha (1975)]).

(1) → (2) (šípka označuje, že úlohu (1) redukujeme na úlohu (2)). Nech $s = \lfloor \log_2 K \rfloor$. Potom namiesto každého $x_i$ dosadíme $2^0y_{i0} + 2^1y_{i1} + \ldots + 2^sy_{is}$, kde $y_{ij} \in \{0, 1\}$ sú nové premenné. Ak si uvedomíme, že zápis čísla $K$ má dĺžku $\Omega(\log_2 K)$, tak hneď vidíme, že uvedená transformácia je polynomiálna.

(2) → (3). Nech súčet všetkých záporných $c_j$ je $c^-$ a kladných $c^+$. Potom pre každé riešenie $x$ úlohy (2) platí: $c^- \leqslant c^Tx \leqslant c^+$. Takto pre nájdenie minima pre $c^Tx$ stačí vyriešiť $\lceil \log_2 (c^+ - c^-) \rceil$ úloh tvaru: $c^Tx \leqslant d$, $Ax = b$, $x_i \in \{0, 1\}$. Skutočne stačí použiť tzv. *metódu bisekcie* na interval $[c^-, c^+]$. (Najprv zvolíme $d = \lfloor (c^+ - c^-)/2 \rfloor$: ak úloha má riešenie, tak ďalej skúmame len interval $[c^-, d]$, inak len interval $[d, c^+]$. V novom intervale opäť nájdeme „stred" atď.) V každej takejto úlohe možno nerovnosť $c^Tx \leqslant d$ nahradiť rovnosťou $c^Tx + 2^0z_0 + 2^1z_1 + \ldots + 2^tz_t = d$, kde $t = \lfloor \log_2 d \rfloor$ a $z_i$ sú nové binárne premenné. Tým získame úlohu tvaru (3).

(3) → (4). Dôkaz je úplne analogický ako pre vetu 2.

(4) → (1). Zrejmé.

(3) → (5). Nech pre každé $i$ súčet záporných $a_{ij}$ je $a_i^-$ a kladných $a_i^+$ a nech $f_i(x) := a_{i1}x_1 + \ldots + a_{in}x_n - b_i$. Potom $a_i^- - b_i \leqslant f_i(x) \leqslant a_i^+ - b_i$. Preto, ak položíme: $q_i := 1 + \max \{|a_i^- - b_i|, |a_i^+ - b_i|\}$, tak $q_i > |f_i(x)|$ a systém rovníc:

(i) $f_1(x) = 0, \ldots, f_m(x) = 0$





môžeme agregovať do jednej rovnice

(ii) $f_1(x) + q_1f_2(x) + q_1q_2f_3(x) + \ldots + q_1q_2 \ldots q_{m-1}f_m(x) = 0$.

Tvrdíme, že ak má rovnica (ii) riešenie, tak má aj systém (i) riešenie a zrejme aj obrátene. (Vidíme, že $f_1(x)$ musí byť deliteľné $q_1$, ale $q_1 > |f_1(x)|$, a preto $f_1(x) = 0$. Po vydelení rovnice číslom $q_1$ zistíme, že $f_2(x) = 0$ atď.) Rovnica (ii) má tvar $a_1x_1 + \ldots + a_nx_n = b$. Ak niektoré $a_i < 0$, tak namiesto $x_i$ dáme $1 - x_i'$. Ak $a_i = 0$, tak rovnicu jednoducho zmenšíme. Ak po týchto úpravách je $b \leqslant 0$, tak rovnica zrejme nemá netriviálne riešenie. Inak máme požadovaný tvar úlohy (5) ekvivalentný so systémom (i). Poznamenajme, že hoci je číslo $q_1q_2 \ldots q_{m-1}$ obrovské, predsa má dĺžku $O(\log q_1 + \log q_2 + \ldots + \log q_{m-1}) \leqslant O(\Sigma\Sigma \log|a_{ij}| + \Sigma \log|b_i|)$, t. j. polynomiálnu vzhľadom na dĺžku zápisu úlohy (3).

(5) → (6) [Karp (1972)]. Stačí uvažovať čísla $a_1, a_2, \ldots, a_n, \Sigma a_i + b, 2\Sigma a_i - b$ (ich súčet je $4\Sigma a_i$). Dôkaz vyplýva z toho, že posledné 2 čísla nemôžu byť v tej istej triede.

(6) → (5). Stačí položiť $b = \Sigma a_i/2$.

(5) → (7). Úloha (5) má riešenie práve vtedy, ak $\max \{a_1x_1 + \ldots + a_nx_n | a_1x_1 + \ldots + a_nx_n \leqslant b\} = b$.

(7) → (2). Úloha (7) je rovnocenná úlohe $\min \{-(c_1x_1 + \ldots + c_nx_n) | a_1x_1 + \ldots + a_nx_n + 2^0y_0 + \ldots + 2^sy_s = b$, všetky $x_i, y_j \in \{0, 1\}\}$, kde $s = \lfloor \log_2 b \rfloor$.

(5) → (8). Úloha (5) je rovnocenná úlohe: Nájsť celé čísla $x_i, y_i \geqslant 0$ spĺňajúce rovnice $a_1x_1 + \ldots + a_nx_n = b$, $x_1 + y_1 = 1, \ldots, x_n + y_n = 1$. Agregovaním týchto $n + 1$ rovníc do jednej (ako v dôkaze (3) → (5)), získame úlohu tvaru (8).

(8) → (9). Analogicky ako (5) → (7).

(9) → (1). Úloha (9) je rovnocenná úlohe: $\min \{-(c_1x_1 + \ldots + c_nx_n) | a_1x_1 + \ldots + a_nx_n + y = b$; $x_1, x_2, \ldots, x_n, y$ sú nezáporné celé čísla nepresahujúce $b\}$. ■

Poznamenajme, že veta 3 platí aj vtedy, ak v (1) nežiadame ohraničenosť premenných [Borosh, Treybing (1976)]. Ak však pripustíme kvadratické ohraničenia (napr. $3x_1x_2 + x_2^2 - x_3 + 20 = 0$), tak je tu kvalitatívny skok: Takéto celočíselné úlohy sú algoritmicky neriešiteľné [Jeroslow (1973)].

## 2. C
## NP-ťažké a NP-ekvivalentné úlohy

V lineárnom, resp. v celočíselnom lineárnom programovaní, sa vyskytovali redukcie úloh dávnejšie, ale často boli obmedzené na úzky okruh problémov, resp. boli len „jednosmerné". Veľký rozvoj polynomiálnych redukcií sa začína prácou

Cooka (1971) a následnou prácou Karpa (1972). Cook založil tzv. triedu NP *rozhodovacích problémov*, t. j. problémov, ktorých riešenie je alebo „áno", alebo „nie" (napr. „má rovnica $3x_1+4x_2+x_3 = 5$ binárne riešenie?") a ktorých „áno" možno verifikovať v polynomiálnom čase (trojica $x_1=0$, $x_2=1$, $x_3=1$ dáva odpoveď „áno", čo možno v polynomiálnom čase overiť). Formálne sa zavádza tzv. *nedeterministický polynomiálny* (skrátene NP) Turingov počítač, ktorý uhádne prípustnú štruktúru (u nás trojicu (0, 1, 1)) a potom ju overí v polynomiálnom čase. Ak je riešenie „nie", tak počítač dáva alebo odpoveď „nie", alebo sa vôbec nezastaví. (Pre rigorózne definície odkazujeme čitateľa napr. na [Aho et al. (1974)], alebo [Garey, Johnson (1979)]). Potom Cook dokázal, že každý problém z triedy NP možno polynomiálne transformovať na problém splniteľnosti (boolovského výrazu: pozri ďalej), ktorý je tiež z NP, a preto dostáva pomenovanie NP-*úplný*. Karp (1972) už uvádza 21 NP-úplných problémov.

Ak pre problém $Q$ existuje NP-úplný problém $P$ tak, že $P$ možno polynomiálne redukovať na $Q$, tak $Q$ sa nazýva NP-*ťažký* problém. Ak pre problém $Q$ existuje problém $P \in$ NP taký, že $Q$ možno polynomiálne redukovať na $P$, tak $Q$ sa nazýva NP-*ľahký*. Taký problém $Q$, ktorý je aj NP-ťažký aj NP-ľahký, nazýva sa *NP-ekvivalentný*.

Pretože problémy v triede NP sú len rozhodovacie, nebudeme sa zaoberať s NP-úplnými problémami, ale skôr s NP-ekvivalentnými problémami, ktoré zahŕňajú priamo aj optimalizačné problémy. Zvlášť nás bude zaujímať, či je nejaký problém NP-ťažký. Takto vystačíme s nasledujúcimi definíciami, kde za referenčný problém sa kladie binárna úloha lineárneho programovania (stručne: 0—1 *úloha* LP). Problém $Q$ sa nazýva NP-*ťažký*, ak 0—1 úlohu LP možno polynomiálne redukovať na $Q$ a NP-*ľahký*, ak $Q$ možno polynomiálne redukovať na 0—1 úlohu LP. Potom NP-*ekvivalentný problém* je problém polynomiálne ekvivalentný s 0—1 úlohou LP. Všetky známe NP-ťažké problémy sa vyznačujú tým, že pre žiadny z nich nie je doteraz známy polynomiálny algoritmus. Na druhej strane nie je dokázané, že pre NP-ekvivalentné problémy také algoritmy neexistujú.

## PROBLÉM SPLNITEĽNOSTI

Takto už poznáme 9 NP-ekvivalentných problémov (podľa vety 2.B.3). Nasledujúca veta nám umožní dokázať, že mnohé ďalšie problémy sú NP-ťažké, keďže problém splniteľnosti sa často dá polynomiálne pretransformovať na daný problém. Najprv uveďme definíciu.

*Problém splniteľnosti:* Daný je boolovský výraz v konjunktívnej normálnej forme $F = C_1 \wedge C_2 \wedge \ldots \wedge C_m$ nad premennými $x_1, \ldots, x_n$, t. j. každé $C_j$ je elementárnou disjunkciou $d_{j1} \vee d_{j2} \vee \ldots \vee d_{js(j)}$, kde každé $d_{jk} \in \{x_1, \ldots, x_n, \bar{x}_1, \ldots, \bar{x}_n\}$ a $\bar{x}_i$





označuje negáciu premennej $x_i$. Úlohou je zistiť, či existujú pravdivostné hodnoty premenných tak, aby $F$ bolo splnené (pravdivé). Elementárne disjunkcie sú nazývané *klauzulami* a premenné a ich negácie *symbolmi*. Pravdivú hodnotu budeme označovať 1 a nepravdivú 0.

**Veta 1.** Problém splniteľnosti je NP-ťažký.

DÔKAZ. Dáme polynomiálnu redukciu úlohy binárnej lineárnej rovnice $a_1x_1 + \ldots + a_nx_n = b$ ($a_1, \ldots, a_n, b$ sú prirodzené čísla) na problém splniteľnosti, čo podľa vety 2.B.3 stačí. Môžeme predpokladať, že $b \geqslant \max a_i$ (inak úlohu možno zmenšiť vynechaním tých $a_i$, ktoré sú väčšie ako $b$). Nech $t = \lfloor \log_2 b \rfloor$ a nech binárne zápisy čísel $a_1, \ldots, a_n, b$ sú: $a_i = a_i^0 2^0 + \ldots + a_i^t 2^t$ ($i = 1, \ldots, n$), $b = b^0 2^0 + \ldots + b^t 2^t$, kde $a_i^k, b^k \in \{0, 1\}$. Potom zrejme $n$-tica $(x_1, x_2, \ldots, x_n)$ (ide o usporiadanú $n$-ticu, ale to je už vyjadrené okrúhlymi zátvorkami) je riešením rovnice $a_1x_1 + \ldots + a_nx_n = b$ práve vtedy, keď je riešením nasledujúceho systému $t+1$ kongruencií (mod 2) a jednej rovnice. (Systém simuluje počítanie v dvojkovej sústave.)

$$
\text{(i)} \quad \begin{cases}
S^0 := a_1^0x_1 + \ldots + a_n^0x_n & \equiv b^0, \\
S^1 := a_1^1x_1 + \ldots + a_n^1x_n + (S^0 - b^0)/2 & \equiv b^1, \\
S^2 := a_1^2x_1 + \ldots + a_n^2x_n + (S^1 - b^1)/2 & \equiv b^2, \\
\vdots & \\
S^t := a_1^tx_1 + \ldots + a_n^tx_n + (S^{t-1} - b^{t-1})/2 & \equiv b^t, \\
(S^t - b^t)/2 & = 0.
\end{cases}
$$

Pre vyriešenie systému (i) bude užitočné vyriešiť úlohu tohto typu:

(ii) Pre dané $p_1^k, \ldots, p_{2n}^k$, $q^k \in \{0, 1\}$ treba zistiť, či $\Sigma\, p_i^k \equiv q^k$ (mod 2) a ak áno, tak spomedzi nových $n$ premenných $v_1^k, \ldots, v_n^k$ zvoliť $(\Sigma\, p_j^k - q^k)/2$ jednotiek a ostatné nuly.

Skutočne, ak uvažujeme kongruenciu $S^k \equiv b^k$, tak môžeme klásť $p_j^k = a_j^k x_j$ pre $j = 1, \ldots, n$. Keďže $(S^{k-1} - b^{k-1})/2 \leqslant n$, tak toto číslo možno reprezentovať tak, že spomedzi $p_{n+1}^k, \ldots, p_{2n}^k$ zvolíme práve $(S^{k-1} - b^{k-1})/2$ jednotiek. K tomu stačí položiť $p_{n+1}^k = v_1^{k-1}, \ldots, p_{2n}^k = v_n^{k-1}$, kde $v_1^{k-1}, \ldots, v_n^{k-1}$ je výstup z úlohy (ii) pre predchádzajúcu kongruenciu.

Riešenie úlohy (ii) simulujeme v dvoch fázach. V prvej fáze uložíme $(\Sigma\, p_j^k - q^k)/2$ výstupných jednotiek do vektora dĺžky $2n$ (boolovské premenné $u_1^k$, $u_2^k, \ldots, u_{2n}^k$) a potom v druhej fáze tieto jednotky dáme do požadovaného vektora $(v_1^k, \ldots, v_n^k)$ dĺžky $n$.

1. Fáza: Zavedieme boolovské premenné $y_0^k, y_1^k, \ldots, y_{2n}^k$. Pre každé $i = 1, \ldots, 2n$ požadujeme splnenie vzťahov:

$$\text{(ii.a)}\quad \begin{cases} y_0^k = q^k, \\ [(y_{i-1}^k = 0) \wedge (p_i^k = 1)] \Rightarrow [(y_i^k = 1) \wedge (u_i^k = 0)], \\ [(y_{i-1}^k = 1) \wedge (p_i^k = 1)] \Rightarrow [(y_i^k = 0) \wedge (u_i^k = 1)], \\ [p_i^k = 0] \Rightarrow [(y_i^k = y_{i-1}^k) \wedge (u_i^k = 0)]. \end{cases}$$

Ľahko sa ukáže, že $y_i^k = 0$ práve vtedy, keď rozdiel $\sum_{j=1}^{i} p_j^k - q^k$ dá sa deliť dvoma a že potom tento podiel je rovný

$$\sum_{j=1}^{i} u_j^k.$$

Takto celá kongruencia je splnená práve vtedy, keď $y_{2n}^k = 0$.

2. Fáza: Vektor $(u_1^k, \ldots, u_{2n}^k)$ prerobíme na vektor $(v_1^k, \ldots, v_n^k)$ s tým istým počtom jednotiek. Zavedieme pomocné boolovské premenné $w_{ij}^k$ $(i = 1, \ldots, n: j = n+1, \ldots, 2n)$ a pre každé $g, i \in \{1, \ldots, n\}$ a $j, h \in \{n+1, \ldots, 2n\}$, $g \neq i$, $j \neq h$, žiadame splnenie vzťahov:

$$\text{(ii.b)}\quad \begin{cases} [(u_i^k = 1) \vee (u_j^k = 0)] \Rightarrow [w_{ij}^k = 0], \\ [u_j^k = 1] \Rightarrow [w_{1j}^k \vee w_{2j}^k \vee \ldots \vee w_{nj}^k = 1], \\ w_{ij}^k \wedge w_{ih}^k = 0, \quad w_{gj}^k \wedge w_{ij}^k = 0, \\ v_i^k = u_i^k \vee w_{i,n+1}^k \vee \ldots \vee w_{i,2n}^k. \end{cases}$$

Možno ukázať, že premenné $w_{ij}^k$ zabezpečujú prenos jednotiek $u_{n+1}^k, \ldots, u_{2n}^k$ len na tie miesta $v_i^k$, pre ktoré sú $u_i^k = 0$, a že týmto získame požadovaný výstup $(v_1^k, \ldots, v_n^k)$.

Na vyriešenie systému (i) stačí požadovať splnenie vzťahov (ii.a) a (ii.b) pre každé $k = 0, 1, \ldots, t$, pričom pre $i = 1, \ldots, n$ všade namiesto $p_i^k$ dosadíme $a_i^k \wedge x_i$ a namiesto $p_{n+i}^k$ dosadíme $v_i^{k-1}$, pričom kladieme $v_1^{-1} = v_2^{-1} = \ldots = v_n^{-1} = 0$. Konečne, na splnenie posledného vzťahu v systéme (i) žiadame, aby $v_1^t = v_2^t = \ldots = v_n^t = 0$.

Ak všetky premenné $x_i$, $y_i^k$, $u_i^k$, $v_i^k$, $w_{ij}^k$ a čísla $a_i^k \in \{0, 1\}$ považujeme za boolovské a (ii.a), (ii.b) upravíme podľa známych vzťahov ($A \Rightarrow B$ nahradíme výrazom $\bar{A} \vee B$, $\overline{A \wedge B}$ výrazom $\bar{A} \vee \bar{B}$, $(A \wedge B) \vee C$ výrazom $(A \vee C) \wedge (B \vee C)$ a pod.), tak zrejme získame nejakú konjunktívnu normálnu formu $F$. Ľahko možno overiť, že konštrukcia formy $F$ vyžaduje len polynomiálne mnoho krokov (vzhľadom na dĺžku rovnice $a_1x_1 + \ldots + a_nx_n = b$). Takže tým sme úlohu nájsť binárne riešenie rovnice polynomiálne pretransformovali na úlohu nájsť hodnoty boolovských premenných, ktoré spĺňajú formu $F$. Ešte treba poslednú úlohu redukovať na (rozhodovací) problém splniteľnosti. To však je ľahké. (Zistíme, či $F$ je splniteľná, ak áno, zvolíme hodnotu jednej premennej a znova testujeme: ak je splniteľná, tak zafixujeme hodnotu tej premennej, inak ju zmeníme na opačnú





a zafixujeme: pokračujeme tým, že zvolíme hodnotu druhej premennej atď. Ak $n$ je počet premenných, tak v čase $O(n)$ zistíme ich hodnoty.) ■

V ďalšom uvidíme, že splniteľnosť možno polynomiálne redukovať na mnohé úlohy, čím sa dokáže, že sú NP-ťažké. S tým sa zvyčajne uspokojíme, ale ak chceme ukázať, že sú aj NP-ľahké, tak ich zapíšeme v tvare 0—1 úlohy LP. Napr. v probléme splniteľnosti klauzulu $x_1 \vee \bar{x}_2 \vee \bar{x}_3 \vee x_4$ zapíšeme takto: $x_1 + (1 - x_2) + (1 - x_3) + x_4 \geq 1$, kde už pod $x_i$ myslíme binárne premenné. Ľahšia cesta je zistiť, že príslušná rozhodovacia verzia problému patrí do triedy NP, ale tu uprednostníme redukcie na 0—1 úlohu LP, čo má ten význam, že často sa buduje na riešení úlohy LP, kde namiesto ohraničenia $x_i \in \{0, 1\}$ dáme $0 \leq x_i \leq 1$. Toto nás však väčšinou nebude zaujímať a prenechávame čitateľovi, aby zostrojil príslušné 0—1 úlohy LP. Istou pomôckou môže byť to, že najprv zostrojíme len binárnu úlohu polynomického programovania: min $\{f(x) | h_1(x) = 0, \ldots, h_p(x) = 0$, všetky $x_i \in \{0, 1\}\}$, kde $f, h_1, \ldots, h_p$ sú nejaké polynómy premenných $x_1, \ldots, x_n$, pričom každá premenná tam vystupuje len lineárne, lebo $x_i^2 = x_i$. (Príkladom takej funkcie je $5x_1x_2x_4 - 3x_2x_3 + 4x_2x_4 - 1$.) Potom známym spôsobom (pozri napr. [Taha (1975)]) nahradíme súčiny. Namiesto súčinu $x_1x_2 \ldots x_k$ dáme novú premennú $y$ a pripojíme ohraničenia: $ky \leq x_1 + \ldots + x_k \leqq y + k - 1$, čo je zrejme polynomiálna transformácia.

Poznamenajme, že nie o každom NP-ťažkom probléme vieme dokázať, že je aj NP-ľahký. Napr. úlohu binárnej rovnice $a_1x_1 + \ldots + a_nx_n = b$ možno pretransformovať na spojitú úlohu $\langle a_1x_1 + \ldots + a_1x_n = b, x_1^2 = x_1, \ldots, x_n^2 = x_n \rangle$. Preto úloha *nájsť racionálne (alebo aj reálne) riešenie systému nelineárnych rovníc je* NP-*ťažká*, ale nevieme či je NP-ľahká.

## APROXIMÁCIA

V *optimalizačných úlohách* (t. j. takých, kde minimalizujeme alebo maximalizujeme účelovú funkciu pri istých ohraničeniach) má zmysel hľadať aspoň aproximáciu optima. Existujú rôzne kritériá pre aproximáciu. Za predpokladu, že optimálna hodnota $f^*$ účelovej funkcie $f$ je nenulová, tak snahou je získať také prípustné riešenie $x$, pre ktoré je $f(x)/f^*$ čo najbližšie k hodnote 1. Takto k danej minimalizačnej úlohe s $f^* > 0$ a číslu $\varrho \geq 1$ môžeme priradiť tzv. *$\varrho$-aproximačnú úlohu:* Nájsť prípustné riešenie $x$ tak, aby $f(x)/f^* \leq \varrho$. Analogicky pre maximalizačnú úlohu s $f^* > 0$ a číslo $\varrho \leq 1$ staviame *$\varrho$-aproximačnú úlohu* takto: Nájsť prípustné riešenie $x$ tak, aby $f(x)/f^* \geq \varrho$. (Tieto definície možno zjednotiť pod tzv. $\varepsilon$-aproximačnú úlohu, kde $\varepsilon \geq 0$: zaručiť, aby $|f(x) - f^*|/|f^*| \leq \varepsilon$. My však budeme používať len $\varrho$-aproximáciu.)

*Aproximačný algoritmus* sa niekedy nazýva aj *heuristický* alebo stručne len

*heuristika*. Všeobecnejšie pod pojmom *heuristika* pre daný problém rozumieme postup, ktorý nám nezaručuje vyriešenie každého príkladu problému. Kým aproximačný algoritmus musí vždy dať prípustné, hoci nie optimálne riešenie, heuristika nemusí dať ani prípustné riešenie.

V praxi je nepríjemné stretnúť NP-ťažký problém, avšak niektoré úlohy majú dokonca príslušný $\varrho$-aproximačný problém NP-ťažký aj pre $\varrho \neq 1$. Ilustrujeme to na 0—1 úlohe LP [Sahni, Gonzalez (1976)].

**Veta 2.** Pre každé $\varrho \geq 1$ je $\varrho$-aproximačná 0—1 úloha LP (s kladným minimom) NP-ťažká.

DÔKAZ. Na takúto úlohu pretransformujeme úlohu binárnej rovnice $a_1x_1 + \ldots + a_nx_n = b$ ($a_1, a_2, \ldots, b > 0$). Nech $k \geq \varrho$ je celé číslo. Postavme úlohu min $\{1 + k(b - \Sigma\, a_ix_i) | \Sigma\, a_ix_i \leq b,\ x_i \in \{0, 1\}\}$. Táto úloha má minimum rovné 1 práve vtedy, keď má rovnica $\Sigma\, a_ix_i = b$ riešenie, inak je vždy minimum aspoň $1 + k$. V prvom prípade $\varrho$-aproximačný algoritmus dáva hodnotu nanajvýš $\varrho \leq k$ a v druhom vždy aspoň $1 + k$, a teda podľa toho vieme zistiť, či má rovnica riešenie alebo nie.

Zostrojenú úlohu možno ľahko previesť na základný tvar 0—1 úlohy: min $\{z_1 + 2z_2 + 4z_3 + \ldots + 2^tz_t | \Sigma\, 2^iz_i = 1 + k(b - \Sigma\, a_ix_i) : \Sigma\, a_ix_i + y_i + 2y_2 + \ldots + 2^sy_s = b :$ $x_i, y_i, z_i \in \{0, 1\}\}$, kde $t, s$ sú dostatočne veľké čísla. ■

Na druhej strane, niektoré úlohy majú polynomiálne $\varrho$-aproximačné algoritmy pre ľubovoľné predpísané $\varrho \neq 1$ (ako napr. problém binárneho batohu), resp. len pre nejaké $\varrho \neq 1$ (pozri napr. [Garey, Johnson (1979)]). S niekoľkými sa stretneme aj pri grafových úlohách.

## 2. D
## Niektoré základné NP-ťažké úlohy

Tu uvedieme niekoľko ďalších úloh, z ktorých sa často vychádza pri dokazovaní, že nejaká úloha je NP-ťažká. Najprv uvedieme negrafové a potom grafové úlohy.

### 3-SPLNITEĽNOSŤ

Ide o problém splniteľnosti konjunktívnej normálnej formy, kde každá klauzula má práve 3 symboly. (Napr. $(x_1 \vee \bar{x}_2 \vee \bar{x}_2) \wedge (x_1 \vee x_2 \vee \bar{x}_3)$.) Už Cook (1971) dokazuje:

**Veta 1.** Problém 3-splniteľnosti je NP-ťažký.





DÔKAZ. Dáme transformáciu z problému splniteľnosti. Ak nejaká klauzula má menej symbolov ako 3, tak jednoducho nejaký zopakujeme. Ťažšie je znížiť počet symbolov. To urobíme zavedením nových premenných. Uvažujme nejakú klauzulu $C = a_1 \vee a_2 \vee \ldots \vee a_k$, kde každé $a_i$ je niektorá premenná $x_1, \ldots, x_n$ alebo jej negácia a $k > 3$. Potom namiesto $C$ napíšeme nasledujúcich $k - 2$ klauzúl dĺžky 3, kde $y_1, \ldots, y_{k-3}$ sú nové premenné:

$$(a_1 \vee a_2 \vee y_1),\ (\bar{y}_1 \vee a_3 \vee y_2), \ldots,$$

$$(\bar{y}_{k-4} \vee a_{k-2} \vee y_{k-3}),\ (\bar{y}_{k-3} \vee a_{k-1} \vee a_k).$$

Ak každá z týchto $k - 2$ klauzúl je splnená, tak aspoň jedno $a_i = 1$, a teda $C$ je splnené. Obrátene, ak $C$ je splnené, potom možno zvoliť $y_1, \ldots, y_{k-3}$ tak, že aj každá z $k - 2$ klauzúl bude splnená: Ak $a_1 \vee a_2 = 1$, tak stačí zvoliť $y_1 = y_2 = \ldots$ $y_{k-3} = 0$: ak $a_{k-1} \vee a_k = 1$, tak stačí zvoliť $y_1 = y_2 = \ldots y_{k-3} = 1$: ak nejaké $a_i = 1$, kde $3 \leq i \leq k - 2$, tak stačí zvoliť $y_1 = y_2 = \ldots y_{i-2} = 1$ a $y_{i-1} = \ldots = y_{k-3} = 0$.

Ak takto nahradíme každú klauzulu $C$, tak získame požadovanú formu. Transformácia je zrejme polynomiálna. ■

### 3-ROZMERNÉ PÁRENIE

Nech $X, Y, Z$ sú navzájom disjunktné množiny tej istej mohutnosti $q$ a nech $M \subseteq X \times Y \times Z$ je daná množina trojíc. Úlohou je vybrať 3-*rozmerné párenie* $\hat{M} \subseteq M$, t. j. $q$ trojíc, z ktorých žiadne dve trojice nemajú spoločnú zložku. (Neskôr uvidíme, že „2-rozmerné párenie" je už polynomiálne zvládnuteľné.) Nasledujúci výsledok dokazuje Karp (1972), ale tu sledujeme dôkaz z [Garey, Johnson (1979)].

**Veta 2.** Úloha 3-rozmerného párenia je NP-ťažká.

DÔKAZ. Na túto úlohu pretransformujeme problém 3-splniteľnosti, ktorého premenné sú $x_1, \ldots, x_n$ a klauzuly $C_1, \ldots, C_m$. Množina trojíc $M$ bude pozostávať z troch množín: $M^1$ zabezpečí, že každá premenná $x_i$ bude mať vo všetkých svojich výskytoch v $C_1, \ldots, C_m$ tú istú hodnotu a analogicky aj $\bar{x}_i$: množina $M^2$ zabezpečuje splnenie všetkých klauzúl $C_j$ (ak je to možné); konečne množina $M^3$ obsahuje zvyšné trojice, zabezpečujúce existenciu 3-rozmerného párenia, ak je forma splniteľná. Množiny $M^1$, $M^2$, $M^3$ a $X$, $Y$, $Z$ definujeme postupne.

Pre každú premennú $x_i$ a každú klauzulu $C_j$ zavedieme prvok $x_{ij}$ a analogicky pre $\bar{x}_i$ a $C_j$ je prvok $\bar{x}_{ij}$. Pomocou nich vyjadríme výskyt symbolu v klauzule. Nech

$$X = \{x_{ij}, \bar{x}_{ij} | 1 \leq i \leq n,\ 1 \leq j \leq m\}.$$

Ďalej zavádzame množiny $A$, $B$, $S$ a $T$, pomocou ktorých zostrojíme požadované množiny $M^i$ (množiny $A$, $B$ sa uplatňujú v $M^1$, $S$ v $M^2$ a $T$ v $M^3$):

$$A=\{a_{ij}\,|\,1\leq i\leq n,\ 1\leq j\leq m\},$$
$$B=\{b_{ij}\,|\,1\leq i\leq n,\ 1\leq j\leq m\},$$
$$S=\{s_j\,|\,1\leq j\leq m\},\ S'=\{s'_j\,|\,1\leq j\leq m\},$$
$$T=\{t_k\,|\,1\leq k\leq m(n-1)\},\ T'=\{t'_k\,|\,1\leq k\leq m(n-1)\}.$$

Teraz kladieme:

$$Y=A\cup S\cup T,$$
$$Z=B\cup S'\cup T'.$$

Množina $M=M^1\cup M^2\cup M^3$, kde $M^i\cap M^j=\emptyset$ ak $i\neq j$. Množiny $M^i$ definujeme postupne.

$$M^1=M^1_1\cup M^1_2\cup\ldots\cup M^1_n,$$

kde

$$M^1_i=\{(x_{ij},\,a_{ij},\,b_{ij})\,|\,1\leq j\leq m\}\cup\{(\bar{x}_{ij},\,a_{i,j+1},\,b_{ij})\,|\,1\leq j\leq m\},$$

pričom $a_{i,m+1}=a_{i1}$.

Z trojíc v $M^2\cup M^3$ už žiadna nebude obsahovať $a_{ij}$ ani $b_{ij}$. Každé $a_{ij}$ a tiež $b_{ij}$ je práve v dvoch trojiciach (z $M^1_i$) a v $M^1_k$ pre $k\neq i$ sa nevyskytujú. Potom z definície $M^1_i$ je vidieť, že pre dané $i$, každé 3-rozmerné párenie $\hat{M}$ bude obsahovať alebo všetky trojice $(x_{ij},\,a_{ij},\,b_{ij})$, $1\leq j\leq m$ alebo všetky trojice $(\bar{x}_{ij},\,a_{ij},\,b_{ij})$, $1\leq j\leq m$. To zodpovedá tomu, že každá premenná $x_i$ môže mať iba jedinú hodnotu.

Ďalej definujeme

$$M^2=M^2_1\cup M^2_2\cup\ldots\cup M^2_m,$$

kde

$$M^2_j=\{(x_{ij},\,s_j,\,s'_j)\,|\,x_i\in C_j\}\cup\{(\bar{x}_{ij},\,s_j,\,s'_j)\,|\,\bar{x}_i\in C_j\}.$$

Keďže každá klauzula obsahuje 3 symboly, tak každé $M^2_j$ má nanajvýš 3 trojice. Z týchto trojíc bude v $\hat{M}$ práve jedna (aby sme pokryli $s_j$ treba aspoň jedna, lebo $s_j$ nebude v trojiciach z $M^3$). Trojica $(x_{ij},\,s_j\,s'_j)$ môže byť v $\hat{M}$ iba vtedy, ak sa $x_{ij}$ nevyskytuje v trojiciach z $\hat{M}\cap M^1$ a vtedy je $x_i=1$: analogicky pre $\bar{x}_i=1$.

Doteraz je v $\hat{M}$ $mn+m$ prvkov množiny $X$. Zvyšné prvky, ktorých je $m(n-1)$, pokryjeme množinou $M^3$, kde

$$M^3=\{(x_{ij},\ t_k,\ t'_k),\ (\bar{x}_{ij},\ t_k,\ t'_k)\,|\,1\leq i\leq n,\ 1\leq j\leq m,\ 1\leq k\leq m(n-1)\}.$$

Keďže sme už ukázali, že 3-rozmerné párenie $\hat{M}$ existuje práve vtedy, keď sú všetky klauzuly súčasne splniteľné, tak dôkaz je hotový. ■





## PRESNÉ POKRYTIE 3-MNOŽINAMI

Daná je množina $U$ s $3p$ prvkami a systém $S$ niektorých jej 3-prvkových podmnožín. Úlohou je nájsť podsystém $S'\subseteq S$ taký, že každý prvok $x\in U$ sa vyskytuje práve v jednej podmnožine z $S'$.

Nasledujúce pozorovanie uvádza už Karp (1972).

**Veta 3.** Úloha presného pokrytia 3-množinami je NP-ťažká.

DÔKAZ. Úloha 3-rozmerného párenia sa transformuje ihneď na túto úlohu: $S=\{\{x,\,y,\,z\}\,|\,(x,\,y,\,z)\in M\}$, $U=X\cup Y\cup Z$ a $S'$ potom zodpovedá $\hat{M}$. ■

## 2-ZAFARBENIE HYPERGRAFU (ROZŠTIEPENIE MNOŽÍN)

Daná je konečná množina $U$ a systém neprázdnych podmnožín $S_1$, $S_2$, ..., $S_m\subseteq U$. Tým je zadaný tzv. *hypergraf*: prvky z $U$ sú jeho vrcholy a $S_1$, ..., $S_m$ sú jeho hrany. Úlohou je zafarbiť všetky vrcholy dvoma farbami tak, aby v každej hrane boli vrcholy oboch farieb (inými slovami: rozložiť $U$ na dve triedy tak, aby žiadna množina $S_j$ celá nepatrila do jednej triedy).

Nasledujúci výsledok možno pripísať Lovászovi (1973).

**Veta 4.** Úloha 2-zafarbenia hypergrafu je NP-ťažká, dokonca v prípade, že každá hrana obsahuje nanajvýš 3 vrcholy.

DÔKAZ. Uvedieme transformáciu z problému 3-splniteľnosti. Nech $(a_j\vee b_j\vee c_j)$, $j=1,\ldots,m$, sú klauzuly, kde $a_j,\,b_j,\,c_j\in\{x_1,\ldots,x_n,\bar{x}_1,\ldots,\bar{x}_n\}$. Vrcholová množina bude

$$U=\{x_1,\ldots,x_n,\bar{x}_1,\ldots,\bar{x}_n\}\cup\bigcup_{j=1}^{m}\{y_j,z_j\}\cup\{t,\bar{t}\}.$$

Za hrany volíme:

$\{t,\,\bar{t}\}$,
$\{x_i,\,\bar{x}_i\}$ pre $i=1,\ldots,n$,
$\{a_j,\,b_j,\,y_j\},\ \{c_j,\,z_j\},\ \{y_j,\,z_j,\,t\}$ pre $j=1,\ldots,m$.

Ak sú všetky klauzuly splnené, tak príslušné pravdivé symboly dostanú farbu 1 a ostatné 0. Vidíme, že ak symbol $t$ dostane farbu 1 a $\bar{t}$ farbu 0, tak symboly $y_j$ a $z_j$ možno zafarbiť tak, že zostrojený hypergraf bude zafarbený farbami 0 a 1.

Obrátene, nech tento hypergraf je zafarbený dvoma farbami. Môžeme predpokladať, že $t$ má farbu 1. Potom aspoň jeden zo symbolov $y_j$, $z_j$ má farbu 0, a teda aspoň jeden zo symbolov $a_j$, $b_j$, $c_j$ má farbu 1 ($1\leq j\leq m$). Preto bude každá klauzula splnená. ■

## 3-ZAFARBENIE GRAFU

Úlohou je priradiť každému vrcholu grafu jednu z farieb 1, 2, 3 tak, aby každé dva susedné vrcholy mali rôzne farby. (Vo všeobecnosti minimálny počet farieb, ktorými možno takto zafarbiť vrcholy grafu $G$, nazýva sa *chromatické číslo* alebo *farebnosť* a označuje sa symbolom $\chi(G)$. Vidíme že $\chi(G)=k$ práve vtedy, keď graf $G$ je $k$-partitný, ale nie je $(k-1)$-partitný.)

Zistiť, či je graf 2-zafarbiteľný, je ľahké. Nájsť farebnosť je NP-ťažká úloha [Karp (1972)]. Nasledujúci silnejší výsledok je z práce [Garey, Johnson a Stockmeyer (1976)].

**Veta 5.** Problém 3-zafarbenia grafu je NP-ťažký.

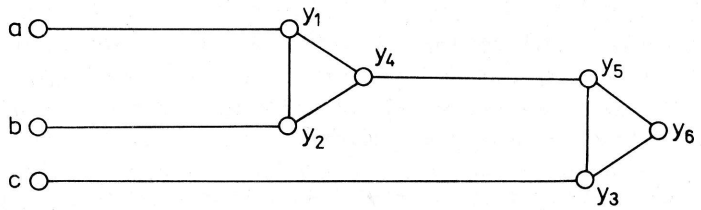
DÔKAZ. Vychádzame z problému 3-splniteľnosti s premennými $x_1, x_2, \ldots, x_p$ a klauzulami $C_1, \ldots, C_q$. Zostrojíme graf $G$, ktorý bude 3-zafarbiteľný práve vtedy, ak všetky $C_j$ možno splniť súčasne. Nech $C_j = a_j \vee b_j \vee c_j$ $(j=1,\ldots,q)$, kde $a_j, b_j, c_j \in \{x_1, \ldots, x_p, \bar{x}_1, \ldots, \bar{x}_p\}$. Kladieme:

$$V(G)=\{x_1,\ldots,x_p,\bar{x}_1,\ldots,\bar{x}_p\}\cup\{y_{jk}\,|\,1\leqslant j\leqslant q,\ 1\leqslant k\leqslant 6\}\cup\{v_1,v_2,v_3\},$$

$$E(G)=\{v_1v_2,\ v_2v_3,\ v_3v_1\}\cup\{x_i\bar{x}_i,\ x_iv_3,\ \bar{x}_iv_3\,|\,1\leqslant i\leqslant p\}\cup \bigcup_{j=1}^{q}(E_j\cup\{v_2y_{j6},\ v_3y_{j6}\}),$$

kde $E_j$ je množina hrán grafu $H_j$ z obr. 1 (index $j$ je tam všade vynechaný). Ku každej klauzule $C_j$ je priradený graf $H_j$. Ak každá klauzula $C_j$ je splnená, tak aspoň jeden zo symbolov $a_j, b_j, c_j$ je pravdivý. Každý pravdivý symbol nech má v $G$ farbu 1 a nepravdivý farbu 2. Z obr. 1 vidieť, že potom zvyšok grafu $H_j$ možno zafarbiť 3 farbami tak, aby $y_{j6}$ malo farbu 1. Nech $v_i$ dostane farbu $i$ pre $i=1, 2, 3$. Takže celý graf $G$ možno zafarbiť tromi farbami.

Obrátene, nech $G$ je zafarbený tromi farbami. Môžeme predpokladať, že $v_i$ má farbu $i$ $(i=1, 2, 3)$. Potom $y_{j6}$ má pre každé $j$ farbu 1. Keďže vrchol $v_3$ je spojený s každým $x_i$ a $\bar{x}_i$, tak $a_j, b_j, c_j$ môžu mať iba farbu 2 a 1. Z obr. 1 však vidieť, že ak by mali iba farbu 2, tak $y_{j6}$ by muselo mať farbu 2, čo je spor. Teda aspoň jedno




Obr. 1. Graf $H$ z dôkazu vety 5.





z $a_j, b_j, c_j$ má farbu 1, čo necháme zodpovedať pravdivej hodnote, takže každá klauzula $C_j$ bude splnená. (Poznamenajme, že $x_i\bar{x}_i \in E(G)$, takže každé $x_i$ má jedinú hodnotu.) ■

**Dôsledok 5(a).** Problém $\varrho$-aproximácie farebnosti grafu je NP-ťažký pre každé $\varrho<4/3$.

DÔKAZ. Každý $\varrho$-aproximačný algoritmus použije tri farby $(3=\lfloor 3\varrho\rfloor)$ na grafy farebnosti 3 a aspoň štyri farby na grafy farebnosti väčšej ako 3, takže je to algoritmus pre úlohu 3-zafarbenia grafu. ■

Poznamenajme, že dôsledok 5(a) možno rozšíriť až po $\varrho<2$ [Garey, Johnson (1976)], avšak pre $\varrho\geqslant 2$ je to otvorený problém.

Problém tzv. hranového zafarbenia je tiež NP-ťažký [Holyer (1981)].

## MAXIMOVÝ KOMPLETNÝ PODGRAF

V danom grafe treba nájsť kompletný podgraf s maximálnym počtom vrcholov. Nasledujúce tvrdenie dáva už Karp (1972).

**Veta 6.** Problém maximového kompletného podgrafu je NP-ťažký.

DÔKAZ. Urobíme transformáciu z problému splniteľnosti. Nech $C_1, \ldots, C_q$ sú klauzuly nad premennými $x_1, \ldots, x_p$. Zostrojme graf $G$ takto:

$$V(G)=\{(s,i)\,|\,s \text{ je symbol v } C_i,\ s\in\{x_1,\ldots,x_p,\bar{x}_1,\ldots,\bar{x}_p\},\ 1\leqslant i\leqslant q\},$$

$$E(G)=\{[(s,i),(t,j)]\,|\,s\neq\bar{t},\ i\neq j\}.$$

Ľahko vidieť, že forma $C_1\wedge\ldots\wedge C_q$ je splniteľná práve vtedy, ak $G$ má kompletný podgraf s $q$ vrcholmi. ■

**Dôsledok 6(a).** Problém nájsť nulový indukovaný podgraf (*nezávislú množinu*) s maximálnym počtom vrcholov je NP-ťažký.

DÔKAZ. Stačí urobiť komplement grafu (do kompletného) a použiť predchádzajúcu vetu. ■

## MINIMOVÉ VRCHOLOVÉ POKRYTIE HRÁN

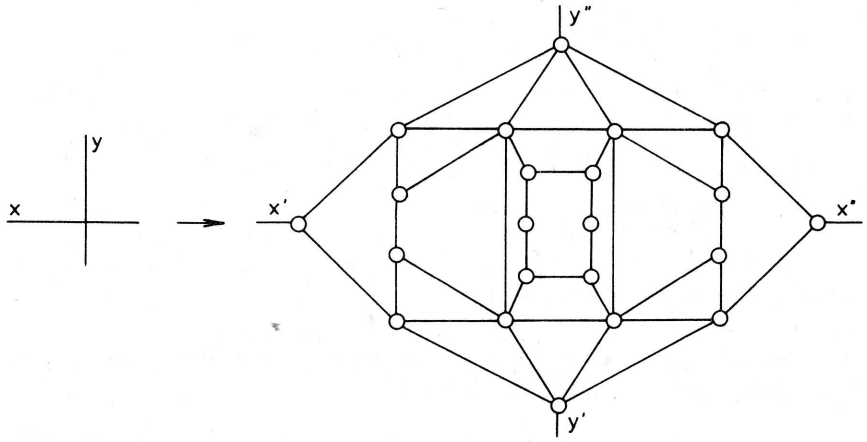
V danom grafe treba nájsť minimálny počet vrcholov pokrývajúcich všetky hrany. (Vrchol pokrýva hranu, ak je jej krajom.)

Základ nasledujúceho výsledku je už v práci [Karp (1972)]: predkladaná veta je z práce [Garey, Johnson(1977)].

**Veta 7.** Problém minimového vrcholového pokrytia hrán je NP-ťažký dokonca pre planárne grafy s maximálnym stupňom 3.

DÔKAZ. Nech je daný graf $G$. Ľahko vidieť [Gallai (1959)], že nejaká množina $M \subset V(G)$ pokrýva všetky hrany práve vtedy, ak je množina $V(G) - M$ nezávislá. Takto tvrdenie vyplýva ihneď z dôsledku 6(a), ak nekladieme obmedzenia na grafy.

Teraz dokážeme [Garey, Johnson, Stockmeyer (1976)], že sa možno obmedziť na planárne grafy. Pre daný graf $G$ uvažujme taký jeho diagram v rovine, v ktorom každý bod roviny je priesečníkom nanajvýš dvoch hrán. Každé križovanie hrán možno nahradiť tak, ako na obr. 2, a tým získať planárny graf $\hat{G}$ s $n(G)+22t$ vrcholmi a $m(G)+42t$ hranami, kde $t$ je počet priesečníkov v diagrame grafu $G$. Prenechávame čitateľovi, aby overil, že $p$ je minimálny počet vrcholov pokrývajúcich všetky hrany v $G$ práve vtedy, ak $p+13t$ je minimálny počet vrcholov pokrývajúcich všetky hrany grafu $\hat{G}$ (pokrytie v $G$ rozšírte na pokrytie v $\hat{G}$ pridaním vhodných 13 vrcholov v každom podgrafe z obr. 2). Tým je ukázané tvrdenie pre planárne grafy.

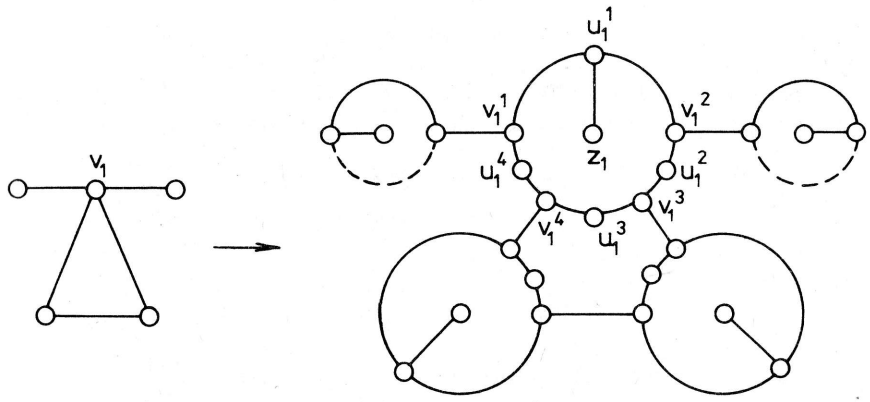



Obr. 2. Planárna realizácia priesečníka.

Konečne, dokážme tvrdenie aj pre planárne grafy s maximálnym stupňom 3. Na to stačí v $\hat{G}$ nahradiť každý vrchol jednoduchým podgrafom ako na obr. 3 (vrchol stupňa $s$ nahradíme cyklom dĺžky $2s$, z ktorého ešte vychádza jedna koncová hrana). Pôvodné hrany incidujú s každým druhým vrcholom cyklu. Získaný planárny graf (alebo multigraf) $G'$ má zrejme maximálny stupeň 3. Zasa prene-








Obr. 3. Zmenšenie stupňov na maximálny stupeň 3.

chávame čitateľovi, aby overil, že $q$ je minimálny počet vrcholov pokrývajúcich všetky hrany v $\hat{G}$ práve vtedy, keď $q + \Sigma \deg_{\hat{G}} v = q + 2\hat{m}$ je minimálny počet vrcholov pokrývajúcich všetky hrany v $G'(\hat{m} = m(\hat{G}))$.

Sumarizujúc vidíme, že z minimového pokrytia v $G'$ vieme určiť minimové pokrytie v $\hat{G}$ a potom aj v $G$. ■

**Dôsledok 7(a).** Problém nájsť nezávislú množinu s maximálnym počtom vrcholov je NP-ťažký dokonca pre planárne grafy s maximálnym stupňom 3.

## HAMILTONOVSKÉ ÚLOHY

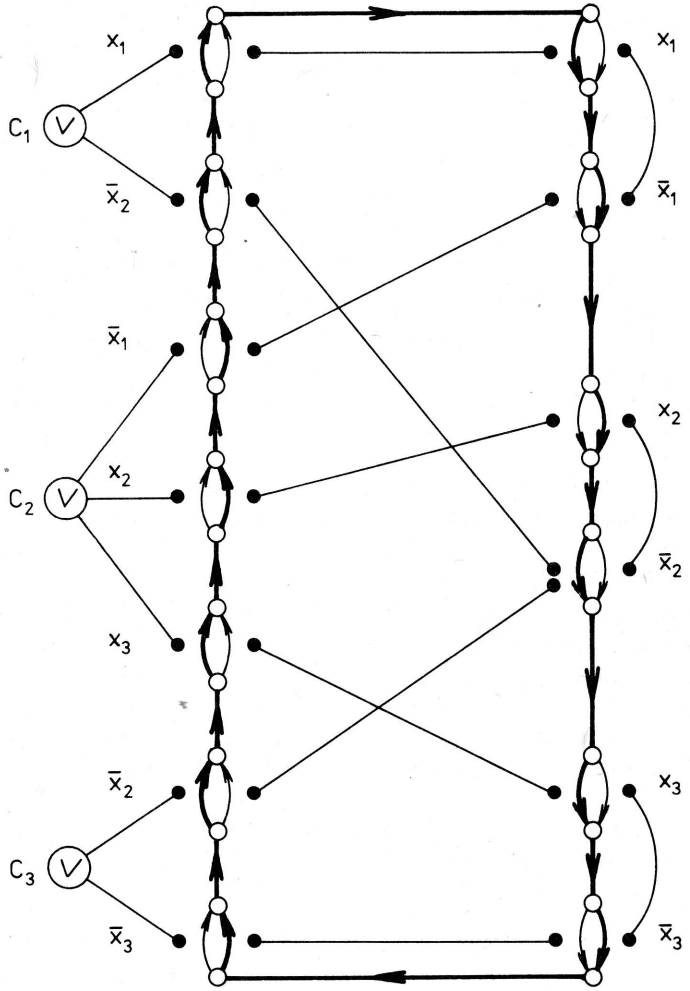
V danom digrafe treba nájsť tzv. hamiltonovský cyklus, t. j. cyklus obsahujúci všetky vrcholy. Príbuzné sú problémy nájdenia cesty, resp. polocyklu, resp. polocesty obsahujúcej všetky vrcholy. Tieto hamiltonovské úlohy sú NP-ťažké, ako uvádza už Karp (1972). Ďalšie príspevky boli zamerané na zosilnenia v rôznych smeroch. Nasledujúci výsledok a dôkaz je z práce [Plesník (1979)] a je založený na myšlienkach z [Garey, Johnson, Tarjan (1976)].

**Veta 8.** Problém nájsť hamiltonovský cyklus je NP-ťažký, dokonca v planárnych bipartitných digrafoch, kde každý vrchol $v$ má alebo $\deg^-(v)=1$ a $\deg^+(v)=2$ alebo $\deg^-(v)=2$ a $\deg^+(v)=1$.

DÔKAZ. Uvedieme transformáciu z problému splniteľnosti. Nech forma $F = C_1 \wedge C_2 \wedge \ldots \wedge C_q$, kde $C_j = s_{j1} \vee s_{j2} \vee \ldots \vee s_{jk_j}$ $(k_j \geq 2)$, pričom $s_{ij} \in \{x_1, \ldots, x_p, \bar{x}_1, \ldots, \bar{x}_p\}$. K forme $F$ skonštruujeme digraf $G$ s požadovanými vlastnosťami a taký, že $F$ je splniteľná práve vtedy, ak $G$ má hamiltonovský cyklus. Túto konštrukciu

ilustrujeme na obr. 4, kde je základná schéma digrafu $G$ pre $F = (x_1 \vee \bar{x}_2) \wedge (\bar{x}_1 \vee x_2 \vee x_3) \wedge (\bar{x}_2 \vee \bar{x}_3)$. (Tu je $p = q = 3$, $k_1 = 2$, $k_2 = 3$, $k_3 = 2$.)

Najprv priradíme k $F$ multidigraf $Z$ tvaru cyklu dĺžky $2(k_1 + \ldots + k_q + 2p)$, v ktorom sú striedavo umiestnené jednoduché a dvojnásobné šípy. Predpokladá




Obr. 4. Konštrukcia digrafu $G$ pre formu $F = (x_1 \vee \bar{x}_2) \wedge (\bar{x}_1 \vee x_2 \vee x_3) \wedge (\bar{x}_2 \vee \bar{x}_3)$: $F(1, 0, 1) = 1$.



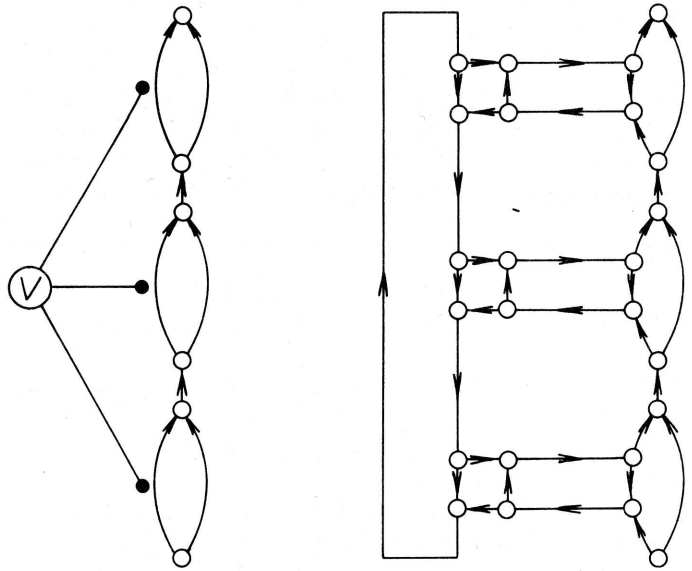


sa, že (a) ku každej klauzule $C_j$ je priradený úsek „cyklu" $Z$, ktorý obsahuje $k_j$ dvojnásobných šípov ($2k_j$ vrcholov), (b) okrem toho každej dvojici symbolov $x_i$ a $\bar{x}_i$ zodpovedá úsek obsahujúci dva dvojnásobné šípy, (c) všetky dvojnásobné šípy zodpovedajúce klauzulám sú v jednom úseku (budeme predpokladať, že v ľavom) a ostatné zodpovedajúce symbolom $x_i$ a $\bar{x}_i$ sú v druhom (v pravom). Tento multidigraf $Z$ je základom, ktorý budeme rozširovať na požadovaný digraf $G$ pomocou špeciálnych súčiastok (pozri obr. 4).

Budeme vyžadovať, aby hodnota pre $x_i$ (alebo $\bar{x}_i$) bola pravdivá, ak hamiltonovský cyklus obsahuje ľavý šíp z každej dvojice šípov priradených premennej $x_i$(alebo $\bar{x}_i$). Napríklad, hamiltonovský cyklus vyznačený na obr. 4 hrubou čiarou dáva $x_1 = 1$, $x_2 = 0$, $x_3 = 1$.

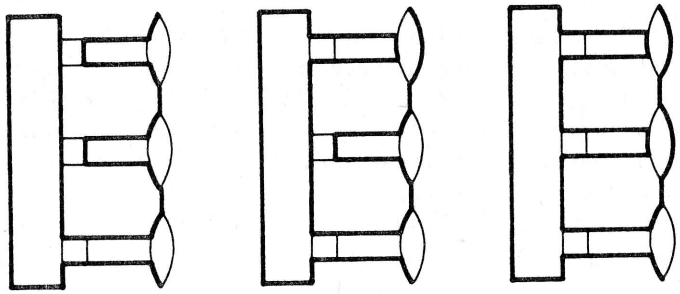
Ukážeme, že v digrafoch možno simulovať isté súčiastky (načrtnuté na obr. 4), ktoré nám zabezpečia, že každý hamiltonovský cyklus v $Z$ bude: (i) obsahovať aspoň jeden ľavý šíp v každom $C_j$, (ii) pre každý výskyt symbolu $x_i$(alebo $\bar{x}_i$) bude prechádzať alebo cez všetky ľavé šípy alebo cez všetky pravé šípy priradené k $x_i$ (alebo k $\bar{x}_i$), (iii) nikdy neprechádza súčasne cez nejaký ľavý šíp a cez nejaký pravý šíp zodpovedajúci $x_i$ (analogicky $\bar{x}_i$).

Hamiltonovský cyklus v $Z$ spĺňajúci (i), (ii) a (iii) budeme nazývať obmedzený hamiltonovský cyklus, stručne OH cyklus.



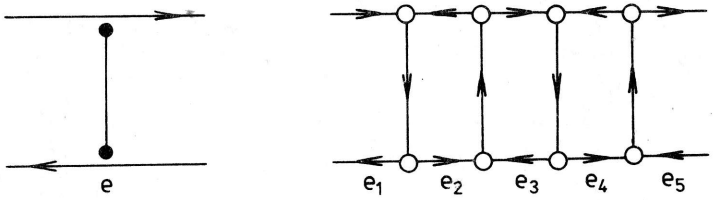
Obr. 5. 3-disjunktor a digrafová realizácia.

Prvá súčiastka nazvaná „$k$-disjunktor" spája $k$ ($\geq 2$) šípov zodpovedajúcich klauzule a zabezpečuje, že aspoň jeden z nich musí ležať v ľubovoľnom OH cykle v $Z$, čo značí, že každá klauzula $C_j$ bude splnená (podmienka (i)). Schematické znázornenie a digrafová realizácia 3-disjunktora je na obr. 5. (Čitateľ si to ľahko urobí pre ľubovoľné $k \geq 2$: ak každá klauzula v $F$ má tri symboly, netreba nič viac robiť.) Každý spájaný šíp $e$ je nahradený tromi šípmi $e_1$, $e_2$, $e_3$ tak, že $e$ sa vyskytuje v nejakom OH cykle v $Z$ práve vtedy, ak $e_1$ a $e_3$ sa súčasne vyskytujú v nejakom hamiltonovskom cykle takto rozšíreného multidigrafu $Z$. To možno ľahko vidieť pomocou obr. 6, kde sú načrtnuté základné možnosti pre taký cyklus (pre 3-disjunktor).



Obr. 6. Možné stavy pre 3-disjunktor (symetrické prípady sú vynechané).

Druhá súčiastka, ktorú nazveme „vylučujúci disjunktor", spája dva šípy a zabezpečuje, že práve jeden z nich sa bude vyskytovať v OH cykle. Istý počet takýchto vylučujúcich disjunktorov (pozri obr. 4) nám zabezpečuje podmienky (ii) a (iii). Vylučujúci disjunktor možno realizovať digrafom tak ako na obr. 7. Vidíme,



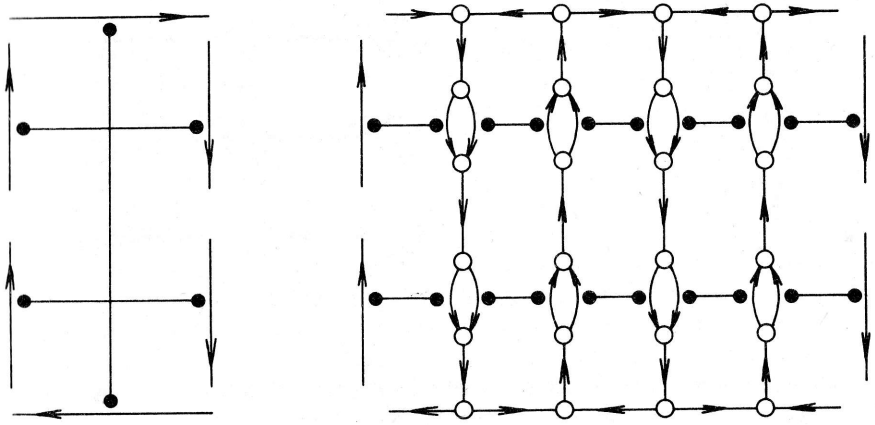

Obr. 7. Vylučujúci disjunktor a digrafová realizácia.

že šíp $e$ sa bude vyskytovať v nejakom OH cykle v $Z$ práve vtedy, ak šípy $e_1$, $e_3$ a $e_5$ sa budú súčasne vyskytovať v nejakom hamiltonovskom cykle takto rozšíreného multidigrafu $Z$.





Na zabezpečenie planarity treba ešte vyriešiť križovanie dvoch vylučujúcich disjunktorov. To však možno realizovať tak ako na obr. 8.



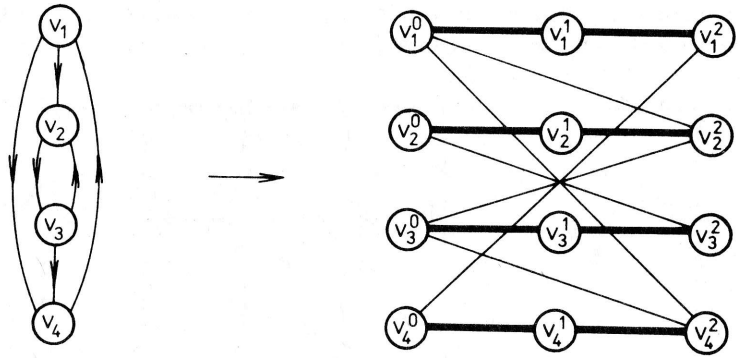
Obr. 8. Križovanie vylučujúcich disjunktorov a planárna realizácia.

Po realizácii všetkých súčiastok zrejme získame digraf $G$, ktorý je planárny a pre každý vrchol $v$ je alebo $\deg^+(v) = 1$ a $\deg^-(v) = 2$ alebo $\deg^+(v) = 2$ a $\deg^-(v) = 1$. Navyše, $G$ nemá polocyklus nepárnej dĺžky (overte!), a teda je bipartitný. Tiež prenechávame čitateľovi overiť, že $G$ sme zostrojili v polynomiálnom počte krokov (vzhľadom na $p + q$). Zrejme platí: forma $F$ je splniteľná práve vtedy, keď $G$ má hamiltonovský cyklus. Dôkaz je hotový. ■

Nasledujúci dôsledok je verziou istého výsledku z práce [Garey, Johnson, Tarjan (1976)].

**Dôsledok 8(a).** Problém nájsť hamiltonovský cyklus je NP-ťažký, dokonca v planárnych bipartitných grafoch s maximálnym stupňom 3.

DÔKAZ. Použijeme transformáciu z digrafov na grafy, ktorú navrhol R. E. Tarjan v [Karp (1972)]. Každý vrchol $v_i$ digrafu $G$ nahradíme tromi vrcholmi $v_i^0$, $v_i^1$ a $v_i^2$: pridáme rebrá $v_i^0 v_i^1$ a $v_i^1 v_i^2$; namiesto každého šípa $v_j v_k$ dáme rebro $v_j^0 v_k^2$. Konštrukciu takéhoto grafu $\hat{G}$ ilustruje obr. 9. Ľahko vidieť, že digraf $G$ má hamiltonovský cyklus práve vtedy, keď $\hat{G}$ má hamiltonovský cyklus. Ak vyjdeme z grafu $G$ spĺňajúceho obmedzenia vety 8, tak zrejme $\hat{G}$ bude bipartitný s maximálnym stupňom 3. ■




Obr. 9. Transformácia digrafu na graf.

**Veta 9.** (a) Problém nájsť hamiltonovskú cestu je NP-ťažký dokonca pre digrafy z vety 8.

(b) Problém nájsť hamiltonovskú cestu je NP-ťažký dokonca pre grafy z dôsledku 8(a).

DÔKAZ. (a) Ak na obr. 4 spojíme vrchný šíp so spodným šípom vylučovacím disjunktorom, tak zrejme získaný digraf $G'$ bude mať hamiltonovskú cestu práve vtedy, keď digraf $G$ (z dôkazu vety 8) má hamiltonovský cyklus (vrchný aj spodný šíp v ňom ležia). Keďže $G'$ má požadované vlastnosti, dôkaz je hotový.

(b) Stačí modifikovať konštrukciu z obr. 9 tak, že jeden zo stredných vrcholov, napr. $v_1^1$, nahradíme dvoma vrcholmi $u$ a $w$ a pridáme rebrá $v_1^0u$ a $wv_1^2$. Získaný graf je zrejme planárny, bipartitný, s maximálnym stupňom 3 a má hamiltonovskú cestu (s krajmi $u$ a $w$) práve vtedy, keď pôvodný graf $\hat{G}$ (z dôkazu dôsledku 8(a)) má hamiltonovský cyklus. ■

## POZNÁMKA

Uviedli sme niekoľko NP-ťažkých úloh. S niekoľkými sa ešte oboznámime v cvičeniach, resp. v ďalších kapitolách. Najväčší zoznam NP-ťažkých problémov je v knihe [Garey, Johnson (1979)], kde čitateľ nájde úlohy z rôznych oblastí: teória grafov, algebra a teória čísel, teória hier, logika, matematické programovanie a i. Je tam vyše 300 základných problémov, ktoré sa ešte špecializujú, čím tento počet mnohonásobne vzrastá.





# 2. E
# O navrhovaní a posudzovaní algoritmov

Nasledujúce kapitoly knihy sú venované grafovým úlohám. Väčšinou ide o kombinatorické problémy, t. j. také, kde z konečnej množiny treba vybrať prvok, resp. prvky požadovaných vlastností. Preto v tejto časti urobíme niekoľko poznámok k navrhovaniu a posudzovaniu algoritmov. Podrobnejšie informácie môže čitateľ nájsť napr. v [Aho et al. (1974)], [Gruska (1975)], [Reingold et al. (1977)] a [Garey a Johnson (1979)].

## NIEKTORÉ VŠEOBECNÉ METÓDY

Tu uvedieme niekoľko jednoduchých, ale užitočných metód používaných pri riešení kombinatorických úloh.

Významná a veľmi všeobecná je myšlienka *rekurzie,* ktorá spočíva v tom, že sa problém redukuje na jeden, resp. niekoľko menších problémov (často toho istého druhu), ktoré môžeme opäť redukovať atď., až získame problémy, ktoré vieme bezprostredne vyriešiť. Potom sa z riešení menších problémov utvoria riešenia väčších, až sa získa riešenie pôvodného problému. Príkladom použitia tejto myšlienky je v časti 2.A spomínaná metóda bisekcie ($O(\log n)$ algoritmus pre zistenie toho, či sa dané číslo nachádza v danej neklesajúcej postupnosti $n$ čísel).

Myšlienku rekurzie nachádzame aj v tzv. *metóde dynamického programovania*. Táto je použiteľná pre (optimalizačné) úlohy, ktorých riešenie možno získať cez postupnosť rozhodnutí. Pritom netreba generovať všetky možné postupnosti, ale len optimálne postupnosti. Ak podpostupnosti optimálnych postupností musia byť tiež optimálne pre príslušné podproblémy, tak hovoríme, že platí tzv. *princíp optimality*. (Vtedy také postupnosti, ktoré obsahujú neoptimálne podpostupnosti, netreba generovať.) Typické použitie tejto myšlienky uvidíme pri hľadaní najkratšieho sledu v kap. 4 ($i$—$j$ úsek najkratšieho sledu musí byť najkratším $i$—$j$ sledom).

Inou realizáciou myšlienky rekurzie je tzv. *metóda vetvenia a ohraničovania*. Tu množinu prípustných riešení optimalizačného problému rozkladáme (vetvíme) na niekoľko podmnožín a riešime, resp. vetvíme, príslušné podproblémy. Ak zistíme, že hodnota účelovej funkcie na nejakej podmnožine nemôže byť lepšia ako doteraz zistená hodnota (ohraničenie), tak túto podmnožinu vynecháme. Pri vyriešení niektorého podproblému často získame lepšie ohraničenie atď. Táto metóda je takmer vždy použiteľná, avšak zvyčajne dáva iba nepolynomiálne algoritmy.

Ďalšie realizácie myšlienky rekurzie sa objavujú v podobe rôznych operácií na grafoch. Takými sú napr. vynechávanie hrán, resp. vrcholov, kontrakcie niektorých podgrafov a pod.

Uvádzané metódy sú samozrejme len náčrty, ktoré treba pre daný problém prispôsobiť a doplniť. Hoci niekedy dávajú algoritmus, často získame len heuristiku. Tak je to aj s nasledujúcou metódou, ktorá sa nazýva *pažravá* (greedy) *metóda*. Úlohou je z danej konečnej množiny prvkov, z ktorých každý má priradenú nejakú hodnotu, vybrať podmnožinu maximálnej sumárnej hodnoty a spĺňajúcu isté ohraničenie. Pažravá metóda vyberá vždy najhodnotnejší prvok, potom druhý najhodnotnejší atď. Pritom sa vždy testuje, či utvorená podmnožina spĺňa požadované ohraničenie. Ak nie, tak posledný pridaný prvok vylúčime a v ďalších krokoch už viac neuvažujeme. Táto myšlienka našla uplatnenie napr. pri konštrukcii najlacnejšej kostry (kap. 5).

Veľký význam pri konštrukcii algoritmov majú rôzne *štruktúry dát*, lebo často zrýchľujú manipuláciu. Keďže nás budú zaujímať skôr základné myšlienky algoritmov, ako ich detailná implementácia (podrobne realizovaný algoritmus), tak sa obmedzujeme len na jednoduché štruktúry dát. Efektívnejšie algoritmy využívajúce zložitejšie a dômyselnejšie štruktúry nájde čitateľ v uvádzaných prameňoch. Používame len také jednoduché a dobre známe štruktúry ako *pole* (vektor, matica), *zásobník* (ukladáme za posledný prvok a vyberáme posledný prvok), *front* (ukladáme za posledný prvok a vyberáme prvý prvok), resp. ich rôzne modifikácie (napr. *prioritný front*) umožňujúce jednoducho vsúvať a vyberať aj vnútorné prvky (pozri najpr. [Aho et al. (1974)] alebo [Reingold et al. (1977)]).

## O POSUDZOVANÍ ALGORITMOV

V časti 2. A sme uviedli definíciu zložitosti. (Ide o *časovú zložitosť*. Tzv. *priestorová zložitosť*, udávajúca potrebnú veľkosť pamäti, sa definuje analogicky: túto však spomíname len sporadicky.) Táto definícia je zameraná na *najhorší prípad*, a preto sa môže stať, že tých príkladov, kde algoritmus vyžaduje veľa krokov, je veľmi málo, resp. sa nevyskytujú medzi príkladmi z praxe. Tak je to napr. so simplexovou metódou (pozri 2. B). Nabáda to k zmene pohľadu na algoritmy, a to neposudzovať ich podľa správania v najhoršom prípade, ale napr. podľa správania v *priemernom* prípade, čo je *pravdepodobnostné* kritérium.

*Pravdepodobnostné metódy* v teórii grafov sú diskutované napr. v [Erdös, Spencer (1974)] a objavujú sa aj aplikácie pri navrhovaní a analyzovaní algoritmov a heuristík (napr. [Pósa (1976)], [Kučera (1977)], [Karp (1977)], [Babai, Kučera (1979)] a i.). Pravda, tu vystupujú otázky týkajúce sa pravdepodobnostného rozloženia. Jednoducho to možno ilustrovať na úlohe nájsť priemer grafu. Postavíme takúto $O(1)$ heuristiku: Bez prezerania grafu prehlásime, že priemer je 2. Keďže je známe, že takmer každý graf má priemer 2, tak uvedená heuristika bude s pravdepodobnosťou 1 dávať správny výsledok. V mnohých príkladoch z praxe by sme zrejme s týmto výsledkom neboli spokojní.





V našej knihe pravdepodobnostné metódy neuvádzame. Tiež tu čitateľ nenájde zovšeobecnenia grafových algoritmov na abstraktnejšie štruktúry, tzv. *matroidy*. V tomto smere odporúčame knihu [Lawler (1976)]. V týchto špeciálnych množinových systémoch (hypergrafoch) sa však stráca prvok názornosti a algoritmy sú už „negrafové". Pre všeobecné hypergrafy neboli dosiahnuté výraznejšie algoritmické úspechy. O hypergrafoch pojednávajú napr. Berge (1970), Zykov (1974), Lovász (1979) a Nešetřil (1979).

Sumárne možno konštatovať, že pri posudzovaní algoritmov sú ešte stále veľké rozdiely medzi teóriou a praxou. Najviac sa to prejavuje pri úlohách malých rozmerov, kde napr. $O(n)$ algoritmus môže byť horší ako $O(n^3)$ algoritmus alebo dokonca ako $O(2^n)$ algoritmus, už len z toho dôvodu, že ide o asymptotickú zložitosť.

Na záver treba zdôrazniť, že mnohé významné problémy z praxe sú NP-ťažké, a preto je malá nádej na existenciu polynomiálnych algoritmov. Avšak existujúce polynomiálne algoritmy pre príbuzné úlohy tu môžu dobre poslúžiť ako základné prvky pri navrhovaní heuristík. Iným východiskom je štúdium špeciálnych podtried problémov, t. j. uvažovať algoritmy pre špeciálne triedy grafových štruktúr. Viaceré také algoritmy možno nájsť v knihe [Golumbic (1980)].

## 2. Z
## Cvičenia a poznámky

1. Nech je dané číslo $k$. Navrhnite $O(n)$ algoritmus pre usporiadanie celých čísel $a_1, a_2, \ldots, a_n$, ak vieme, že $-k \leq a_i \leq k$ pre každé $i$.
2. Navrhnite lineárne (vzhľadom na dĺžku vstupu plus výstupu) algoritmy prevádzajúce jedny grafové dáta na druhé (pozri 1.G).
3. Dokážte, že vybrať z daného systému množín maximálny počet navzájom disjunktných množín je NP-ťažký problém [Karp (1972)].
4. Nech je daný systém $\{S_1, \ldots, S_p\}$ podmnožín množiny $U$. Treba nájsť podmnožinu $U' \subseteq U$ takú, aby $S_i \cap U' \neq \emptyset$ pre každé $i$ a pritom množina $U'$ mala minimálnu mohutnosť. Dokážte, že úloha je NP-ťažká [Karp (1972)].
5. Dokážte zosilnenie vety 2.D.5: Problém 3-zafarbenia planárneho grafu je tiež NP-ťažký.
   Garey, Johnson a Stockmeyer (1976) pridávajú ešte hranicu 4 na maximálny stupeň.
6. Overte, že úlohu zafarbiť graf $G$ $k$ farbami možno sformulovať takto: Nech premenná $x_{ij} \in \{0, 1\}$ $(1 \leq i \leq n;\ 1 \leq j \leq k)$ má hodnotu 1, ak vrchol $i$ dostane farbu $j$ a 0 inak. Potom máme 0—1 úlohu LP (bez účelovej funkcie):

$$\sum_j x_{ij} = 1 \ (1 \leq i \leq n),\ x_{ij} + x_{sj} \leq 1 \text{ ak } is \in E_G \ (1 \leq j \leq k).$$

Pre hľadanie farebnosti grafov bolo navrhnutých mnoho heuristík, ale žiadna z nich nezaručuje dobrú aproximáciu (pozri napr. [Johnson (1974)] a [Plesník (1980)].)

7. Ukážte, že problém rozložiť vrcholy grafu do daného počtu $k$ tried tak, aby počet hrán spájajúcich vrcholy z rôznych tried bol maximálny, je NP-ťažký. [Návod: Uvážte problém farebnosti grafu [Sahni, Gonzales (1976)].]
Úloha je NP-ťažká dokonca pre $k=2$ (tzv. *maximový rez*) [Garey, Johnson, Stockmeyer (1976)]. Sahni a Gonzalez (1976) uvádzajú $(k+1)/k$-aproximačný algoritmus.
8. (a) Treba nájsť minimálny počet vrcholov, ktorých vynechanie z digrafu dá acyklický digraf. Ukážte, že je to NP-ťažké [Karp (1972)]. [Návod: Vrcholové pokrytie hrán grafu: každé rebro nahraďte dvojicou protišípov.]
(b) Dokážte analógiu úlohy (a) pre vynechávanie hrán [Karp (1972)]. [Návod: Použite (a): každý vrchol $w$ nahraďte šípom $w'w''$ a šíp $uv$ šípom $u''v'$.]
(c) Dokážte analógiu (b) pre prípad, že šípy nevynechávame, ale obraciame. [Návod: Dokážte, že minimá sú rovnaké [Grinberg, Dambit (1966)]: použite vetu 1.E.2.]
9. Množina vrcholov $S \subseteq V(G)$ grafu $G$ sa nazýva *dominujúca*, ak pre každé $u \in V(G) - S$ je $S \cap V(u) \neq \emptyset$. Ukážte, že problém nájsť dominujúcu množinu minimálnej mohutnosti je NP-ťažké. [Návod: V probléme vrcholového pokrytia hrán pridajte ku každej hrane $uv$ cestu $ux_{uv}v$, kde $x_{uv}$ je nový vrchol]. Problém zostáva NP-ťažký aj pre planárne grafy s maximálnym stupňom 3 [Garey, Johnson (1979) a Plesník (1980a)].
10. *Problém multicentra* ($p$-centra): Pre dané $p$ treba v grafe $G$ vybrať $p$-vrcholovú množinu $C$ tak, aby $\max_{u \in V(G)} \min_{v \in C} d(u, v)$ bolo minimálne. *Problém multimediánu* ($p$-mediánu): Tu $C$ treba vybrať tak, aby $\sum_{u \in V(G)} \min_{v \in C} d(u, v)$ bolo minimálne.
Dokážte, že obe úlohy sú NP-ťažké a že problém $\varrho$-aproximácie multicentra je pre $\varrho < 2$ tiež NP-ťažký. [Návod: Použite problém dominujúcej množiny.] (Tieto problémy sú NP-ťažké dokonca pre planárne grafy s maximálnym stupňom 3 [Plesník (1980a)].)
Uvedené problémy možno interpretovať ako úlohy otimálneho rozmiestnenia $p$ stredísk (požiarne zbory, nemocnice, obchody a pod.) v uzloch (obciach) komunikačnej (cestnej) siete. V aplikáciách treba uvažovať ohodnotené grafy, kde každá hrana má nejakú dĺžku a každý vrchol $u$ má váhu (dôležitosť) $w(u)$; potom v uvedených definíciách uvažujeme namiesto $d(u, v)$ „*váženú vzdialenosť*“ $w(u)\, d(u, v)$. Algoritmy pre takéto problémy možno nájsť napr. v [Christofides (1975)], [Minieka (1978)] a [Kariv, Hakimi (1979) a (1979a)].
11. Pre daný graf $G$ a číslo $k$ treba nájsť také vrcholové pokrytie $V_1$ všetkých hrán, ktoré tvorí súvislý podgraf (t. j. $G(V_1)$ je súvislý graf). Dokážte, že tento problém je NP-ťažký dokonca pre planárne grafy s maximálnym stupňom 4 [Garey, Johnson (1977)]. [Návod: Uvažujme rovinný diagram grafu $G$ v probléme minimového vrcholového pokrytia a doplňme ho takto: (a) V každej stene (oblasti) zakreslíme uzavretú krivku vedľa hranice steny. Potom do každej pôvodnej hrany $ij \in E(G)$ vsunieme dva nové vrcholy $x_{ij}$, $x_{ji}$ a do príslušného úseku každej priľahlej krivky zakreslíme 2 vrcholy $x'_{ij}$ a $x'_{ji}$; okrem toho pridáme dva vrcholy $x''_{ij}$, $x''_{ji}$ a hrany $x_{ij}x'_{ij}$, $x'_{ij}x''_{ij}$, $x_{ji}x'_{ji}$ a $x'_{ji}x''_{ji}$. Vedľa každej pôvodnej hrany sme prechádzali pri kreslení kriviek 2 razy, a preto z $x_{ij}$ vedú hrany na dve strany, takže bude stupňa 4. (b) Pre každý pôvodný vrchol $v \in V(G)$ pridáme dva vrcholy $v'$ a $v''$ tak, že $v'$ umiestnime na niektorej priľahlej krivke a $v''$ vnútri oblasti. Potom pridáme hrany $vv'$ a $v'v''$. Nech $G'$ je získaný graf a $r$ je počet všetkých vrcholov typu $x'_{ij}$, $x'_{ji}$ a $v'$. Potom $n(G') = n(G) + 2m(G) + 2r$. Dokážte, že $G$ má vrcholové pokrytie mohutnosti $k$ práve vtedy, keď $G'$ má súvislé vrcholové pokrytie mohutnosti $k' := k + m(G) + r$.]
12. Pre daný graf $G$ a číslo $d$ treba nájsť podgraf priemeru $d$ s maximálnym počtom vrcholov. Táto úloha je NP-ťažká, lebo je NP-ťažká pre $d=1$ (veta 2.D.6). Dokážte, že je NP-ťažká aj pre každé pevné $d \geq 2$ [Kyš, Wawruch (1981)]. [Návod: Možno urobiť transformáciu z problému pre $d=1$. Daný graf $G_1$ rozšírte na vhodný graf $G_d$ tak, aby maximový podgraf priemeru $d$ v $G_d$ určoval maximový kompletný podgraf v $G_1$.]
13. Ukážte, že problém rozdelenia multimnožiny prirodzených čísel $\{a_1, \ldots, a_n\}$ na dve časti s rovnakými súčtami, zostáva NP-ťažký aj pre množinu (t. j. $a_i \neq a_j$ pre $i \neq j$). [Návod: Uvážte rozdelenie množiny $\{1, 2, \ldots, n, Ka_1 + 1, Ka_2 + 2, \ldots, Ka_n + n\}$ pre dostatočne veľké $K$.]
14. Ukážte, že tzv. *úloha kvadratického programovania* je NP-ťažká dokonca pri celočíselných dátach (ide o úlohu minimalizovať kvadratickú funkciu $x^T Cx + p^T x$ pri lineárnych ohraničeniach $Ax \leq b$, kde prvky matíc C, A a vektorov $p$, $b$ sú dané celé čísla) [Sahni (1974)]. [Návod: Transformujte úlohu binárneho riešenia systému lineárnych rovníc; ukážte, že každé $x_i \in \{0, 1\}$ práve vtedy, keď $0 \leq x_i \leq 1$ a $\min \Sigma (x_i - x_i^2) = 0$.]
Poznamenajme, že pre úlohu konvexného kvadratického programovania (ak uvedená kvadratická funkcia je konvexná, alebo ekvivalentne, ak $C$ je pozitívne semidefinitná matica) existuje polynomiálny algoritmus [Kozlov, Tarasov, Chačijan (1979)].

# 3
# DOSIAHNUTEĽNOSŤ

Medzi základné, a pritom jednoduché úlohy na grafoch patrí úloha zistiť, či nejaké dva vrcholy v grafe súvisia, resp. nájsť všetky komponenty grafu. Tieto úlohy súvisia s klasickou úlohou prieskumu labyrintov a ňou aj začíname. Komponent grafu je jeho maximálny súvislý podgraf. Tu študujeme aj niektoré iné maximálne podgrafy. Konkrétne budeme hľadať bloky, listy a silne súvislé komponenty. Tiež uvažujeme otázku minimálneho doplnenia grafu (digrafu) na graf (digraf) s takýmito vlastnosťami. Záver kapitoly patrí hľadaniu minimálnych digrafov, resp. maximálnych digrafov s rovnakou dosiahnuteľnosťou (tranzitívna redukcia a tranzitívny uzáver).

## 3. A
## Prieskum grafov

Ak je daný graf $G$ napr. pomocou susedskej matice alebo zoznamov okolí, ťažko možno ihneď zistiť, či $G$ má predpísanú vlastnosť. Intuitívne je zrejmé, že každá „netriviálna" vlastnosť bude vyžadovať preskúmanie grafu. Inak to nie je ani vtedy, ak $G$ je zadaný diagramom a treba zistiť, či je graf $G$ súvislý. Totiž v diagrame nemusia byť jednotlivé komponenty zakreslené oddelene. Ak je pritom graf $G$ (resp. jeho diagram) rozsiahly, tak v jednom momente môžeme pozorovať iba jeho malú časť a graf sa nám javí ako labyrint.

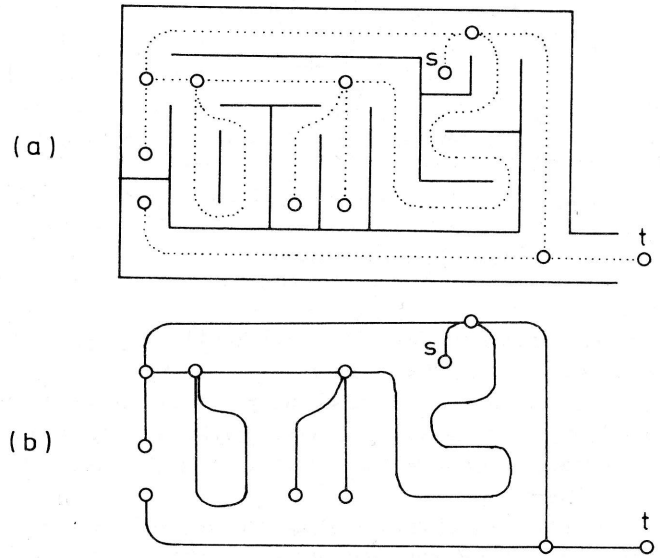
### LABYRINTY

Konštrukcia bludísk je staroveké umenie, ktoré čitateľ iste pozná z mytológie. Avšak matematické štúdium takýchto objektov sa objavuje až koncom 19. storočia a uvádza sa v knihách z rekreačnej matematiky [Lucas (1882)], [Čupr (1949)]. Uvažujme príklad labyrintu na obr. 1(a), kde rázcestia a konce chodieb sú vyznačené krúžkami a možnosti chodenia predstavujú bodkované čiary. Takto sme už priradili k danému labyrintu pseudograf, ktorý je prekreslený na obr. 1(b). Úlohou je dostať sa z miesta $s$ do miesta $t$ (teda von). V inej terminológii treba nájsť $s$—$t$ sled v príslušnom pseudografe. Keďže úvahy pre graf a pseudograf sa v ničom neodlišujú, budeme hovoriť len o grafe.





Jeden z najjednoduchších postupov uvádza Tarry (1895) a formuluje ho pre problém úniku z labyrintu. Tu uvedieme jeho metódu na grafe.




Obr. 1. Labyrint a príslušný pseudograf.

### TARRYHO PRIESKUM GRAFOV

Táto metóda spočíva v konštrukcii sledu začínajúceho v danom vrchole $s$ podľa nasledujúcich dvoch pravidiel.

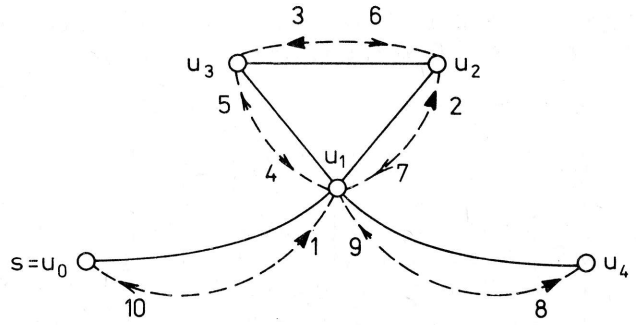
(T1) Každou hranou môžeme v jednom smere prechádzať nanajvýš raz.

(T2) Po tej hrane, po ktorej sme prišli do nejakého vrcholu prvýkrát, môžeme ísť späť iba vtedy, ak niet inej možnosti. ●

Na obr. 2 je príklad grafu a sledu vytvoreného týmto postupom (prerušované čiary so šípkami ukazujú, ktorým smerom hranou prechádzame, číslo určuje poradie hrany v slede, plná šípka označuje, že touto hranou sme do vrcholu prišli prvýkrát).

**Veta 1.** (a) Tarryho prieskum grafov je konečný.

(b) Ak pravidlá (T1) a (T2) nedovoľujú vytvoriť dlhší sled, tak posledný vrchol sledu je $s$ (teda sled je uzavretý) a každá hrana komponentu grafu obsahujúceho $s$ je prejdená oboma smermi.



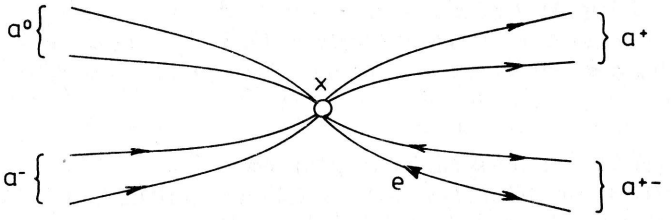

Obr. 2. Tarryho prieskum.

DÔKAZ. (a) Graf má konečne mnoho hrán a pravidlo (T1) zakazuje prechádzať hranou viac ako 2 razy (raz v každom smere).

(b) Nech je utvorený prípustný $s$—$x$ sled $Q$, ktorý už nemožno predĺžiť. Ak by $x \neq s$, tak existuje hrana $e$, ktorou sme prišli do $x$ prvýkrát. Z nepredĺžiteľnosti sledu vyplýva, že hrana $e$ je už prejdená aj v smere od $x$. Vráťme sa do okamihu tesne po tomto prejdení hrany $e$.

Všetky hrany incidentné s $x$ možno rozdeliť na triedy (obr. 3). Nech $a^0$, $a^-$, $a^+$, resp. $a^{+-}$ je počet tých hrán, ktoré sme vôbec neprešli, ktorými sme do $x$ len prišli, z $x$ len odišli, resp. prišli aj odišli. Z našich predpokladov máme: $a^+ + a^{+-} = a^- + a^{+-}$ (počet príchodov a odchodov musí byť rovnaký), $a^0 + a^- = 0$ (pravidlo (T2)). Teda aj $a^+ = 0$, čo značí, že všetky hrany incidujúce s $x$ sú už prejdené dvakrát. Takže po tomto odchode z $x$ nemožno nikdy do $x$ prísť (T1). Tento spor dokazuje, že $x = s$.

Je zrejmé, že celý sled $Q$ leží v komponente $C$ obsahujúcom $s$. Treba ukázať, že každá hrana v $C$ je prejdená oboma smermi (t. j. nachádza sa v $Q$ dvakrát). Nech




Obr. 3. Situácia po odchode hranou e.





$s = u_0, u_1, \ldots, u_q$ je poradie vrcholov, v akom sa v $Q$ vyskytujú prvýkrát. Indukciou dokážeme, že všetky hrany incidujúce s $u_r$ ($r = 0, 1, \ldots, q$) sú prejdené oboma smermi. Z $u_0$ sme vyštartovali a teraz v ňom stojíme, takže počet odchodov sa rovná počtu príchodov, a preto každá hrana incidujúca s $u_0$ je prejdená oboma smermi. Teda tvrdenie pre $u_0$ platí. Nech $r > 0$ a nech $u_k u_r$ je hrana, ktorou sme do $u_r$ prišli prvýkrát. Pretože $k < r$, tak táto hrana je podľa indukčného predpokladu prejdená oboma smermi. Avšak po jej prejdení v smere od $u_r$ už musia byť všetky hrany incidentné s $u_r$ prejdené oboma smermi (to sa ukáže ako vyššie pre $x$, obr. 3).

Z dokázaného už ľahko (indukciou) vidieť, že vrcholy každej $u_0$—$x$ cesty sme navštívili, a teda prezreli celý komponent $C$. ■

Poznamenajme, že ak chceme nájsť nejakú $s$—$x$ cestu, kde $s$ a $x$ sú predpísané vrcholy, tak nie je potrebné, aby algoritmus prebehol celý. Jednoducho ho zastavíme, ak prídeme do $x$ prvýkrát. Tým máme zaručený $s$—$x$ sled. Jeho podsled pozostávajúci z hrán prejdených práve raz je $s$—$x$ ťah (nie nevyhnutne cesta).

Pre skutočný labyrint Tarry navrhuje zaznamenať prechody chodbami a smery takto: Ak prechádzame chodbou prvý raz, zanecháme na jej začiatku dve značky a na jej konci jednu alebo tri značky podľa toho, či prichádzame do známeho (t. j. už predtým navštíveného) alebo do nového rázcestia. Ak sa vraciame chodbou, kde už na začiatku bola jedna značka, tak k nej pridáme ešte jednu (potom bude mať na každom konci dve). Potom obe pravidlá (T1) a (T2) možno zhrnúť do jedného: Na ľubovoľnom rázcestí nemožno vstúpiť do chodby s dvoma značkami a možno vstúpiť do chodby so žiadnou alebo jednou značkou, a ak niet inej možnosti, tak aj do chodby s tromi značkami.

## TRÉMAUXOV PRIESKUM

Analogický algoritmus ako Tarry dal už skôr Trémaux a uverejnil ho v knihe [Lucas (1882)] (avšak s chybným dôkazom). V skutočnosti ide o špecializáciu Tarryho algoritmu: Pravidlá (T1) a (T2) doplnené o pravidlo:

(T3) Ak prídeme hranou prechádzanou prvýkrát do známeho vrcholu, tak hneď v nasledujúcom kroku sa touto hranou vraciame. ●

Na rozdiel od všeobecného Tarryho algoritmu tu už máme zaručené, že z vytvoreného $s$—$x$ sledu stačí vynechať dvakrát prejdené hrany a získame $s$—$x$ cestu. Naviac takéto vynechávanie možno ľahko robiť už pri generovaní $s$—$x$ sledu, keďže stačí porovnávať poslednú hranu s predchádzajúcou a ak sú rovnaké, tak tento úsek dĺžky 2 zo sledu vynechať.

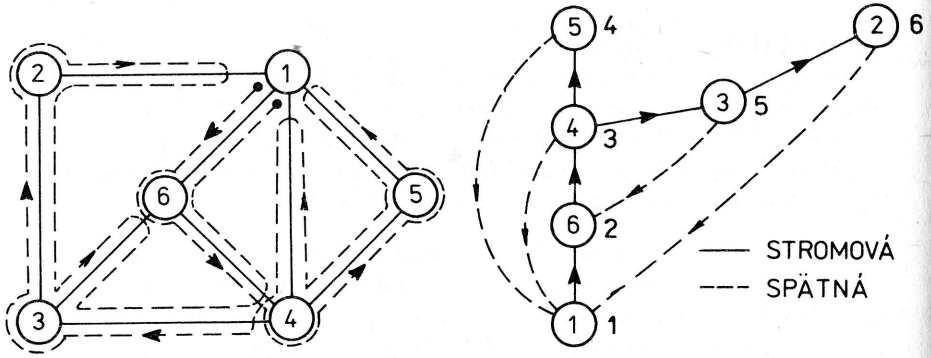
## POSTUPY DO HĹBKY A DO ŠÍRKY

Trémauxov algoritmus prieskumu grafu sa znova dostáva do pozornosti až okolo r. 1970 (porovnaj [Tarjan (1972)]), kedy sa začína využívať pri konštrukcii efektívnych algoritmov (v anglickej literatúre dostáva meno "depth first search"). Ide o zakódovanie grafu do vhodného tvaru, ktorý je užitočný v rôznych algoritmoch.

Trémauxov prieskum sa nazýva *postup do hĺbky*, ak je upravený takto: Vrcholy dostávajú čísla 1, 2, ..., $n$ v tom poradí, ako ich objavujeme (prvýkrát nachádzame). Ak sme zastali a ešte existujú neobjavené vrcholy (tie sú v inom komponente), tak v ľubovoľnom z nich začneme vytvárať nový sled atď., až pokiaľ nepreskúmame všetky vrcholy (číslovanie vrcholov pokračuje, takže posledný objavený vrchol dostane číslo $n$). Každá hrana dostane šípku označujúcu smer prvého prechodu hranou. Hrany rozdeľujeme do dvoch tried: do prvej triedy dávame tie, ktoré nás priviedli k novým vrcholom (tzv. *stromové* hrany) a do druhej ostatné, tzv. *spätné* hrany alebo *chordy*. Hrany prvej triedy tvoria zdrojové stromy (vetvenia). Súvislý graf sa takto zmení na digraf, ktorý sa nazýva *palma*. Ako obvykle pri zdrojových stromoch, ak $ij$ je stromový šíp, tak $i$ sa nazýva *otcom* vrcholu $j$ a $j$ *synom* vrcholu $i$. Obr. 4 ilustruje postup do hĺbky.

Teraz opíšeme algoritmus pre postup do hĺbky trochu podrobnejšie, z čoho bude vidieť, že uvedené myšlienky možno realizovať v lineárnom čase.

Nech $f(j)$ označuje otca vrcholu $j$, t. j. vrchol $j$ bol objavený hranou $ij$, kde $i=f(j)$. Ďalej, nech $p(j)$ označuje *poradie*, v ktorom bol vrchol $j$ objavený. Číslo $k$ udáva priebežný počet objavených vrcholov. Predpokladáme, že skúmame graf




Obr. 4. Postup do hĺbky (graf a získaná palma).





$G$ s $V(G)=\{1, 2, \ldots, n\}$. Číslo $q$ udáva najmenší nepreskúmaný vrchol a $s$ je tzv. *koreň* a označuje začiatok vytváraného sledu: $i$ a $j$ sú premenné vrcholy. Predpokladáme, že graf $G$ je zadaný okoliami vrcholov $V(i)$.

**Algoritmus**

KROK 0: Pre každé $i=1, 2, \ldots, n$ sú prvky vo $V(i)$ neoznačené a položíme $f(i):=p(i):=0$. Ďalej kladieme $f(n+1):=0$, $i:=s:=k:=p(1):=1$, $q:=2$.

KROK 1: Ak existuje vo $V(i)$ neoznačený prvok, tak zvolíme jeden z nich, nech je to $j$, a urobíme:

⌈Ak $p(j)=0$ ($j$ bolo doteraz neobjavené), tak urobíme:
⌈Ak $j=q$, tak urobíme:
A: ⌈Ak $q \leq n$, tak
⌊$q:=q+1$. Ak $f(q)>0$, tak ideme na A.
Prvok $j$ vo $V(i)$ dostane označenie „stromová“ (hrana $ij$ je stromová), $f(j):=i$, $k:=k+1$, $p(j):=k$, $i:=j$ a ideme na krok 1.
Ak $p(j)>0$, tak urobíme:
⌈Ak $j=f(i)$, tak $j$ vo $V(i)$ dostáva označenie „stromová“, inak „spätná“.
⌊Ideme na krok 1.

KROK 2: Ak sú všetky prvky vo $V(i)$ označené, tak urobíme:

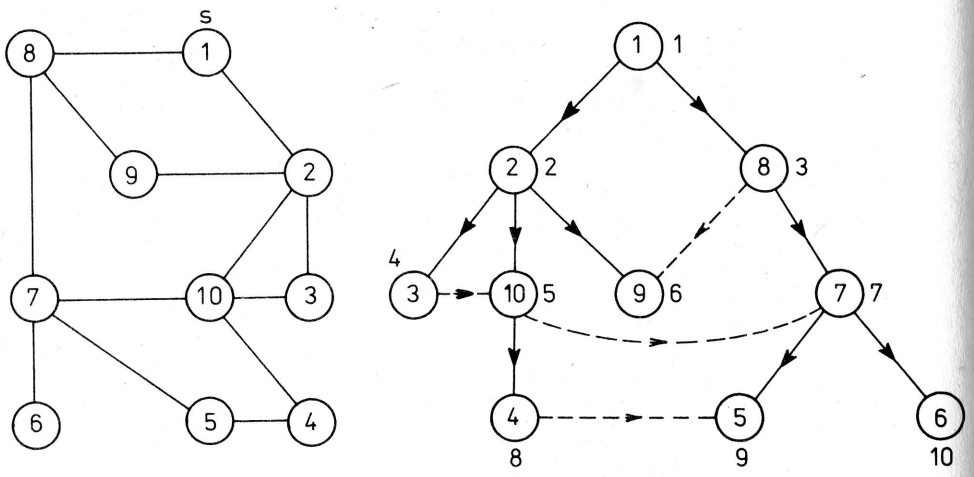
⌈Ak $i=s$, tak urobíme:
⌈Ak $q>n$, tak STOP. Inak $i:=s:=q$ a urobíme:
A′: ⌈Ak $q \leq n$, tak
⌊ $q:=q+1$. Ak $f(q)>0$, tak ideme na A′.
⌊Ideme na krok 1.
⌊Inak $i:=f(i)$ a ideme na krok 1. ●

Poznamenajme, že každá hrana je v okoliach $V(i)$ zapísaná 2-krát a algoritmus teda každú hranu označí 2-krát. K tomu, aby sme zistili smer prvého prechodu nejakou hranou $ij$, stačí zistiť, či $p(i)<p(j)$ alebo obrátene. V prvom prípade sme stromovú hranu prechádzali v smere od $i$ ku $j$ a spätnú od $j$ ku $i$: v druhom prípade je to obrátene.

Hneď vidieť, že ak vo výbere elementu $j$ z $V(i)$ máme dobrý systém (napr. berieme vždy posledný neoznačkovaný a jeho polohu si zaznamenávame), tak uvedený algoritmus je zložitosti $O(m+n)$.

V istom zmysle opačný je tzv. *postup do šírky* (breadth first search), v ktorom skúmame najprv vrchol $s$ (t. j. hrany z $E(s)$), potom jeho susedov, potom susedov

týchto susedov atď. Ide tu vlastne o rozklad vrcholovej množiny podľa vzdialeností (od $s$). (Podrobnejšie takéto úlohy skúmame v kap. 4.) Postup do šírky však negeneruje žiaden sled, keďže „preskakuje". Je to však systematický postup, ktorý možno tiež realizovať v lineárnom čase (3.Z.3). Na obr. 5 je príklad.




Obr. 5. Postup do šírky.

## HĽADANIE KOMPONENTOV GRAFU

Lineárny algoritmus pre nájdenie komponentov grafu sa používal už okolo r. 1970 (porovnaj [Hopcroft a Tarjan (1973)]). Takýto algoritmus dáva aj Tarryho prieskum. Tu si však ukážeme použitie postupu do hĺbky, keďže ho máme realizovaný pomerne detailne.

V skutočnosti stačí algoritmus z predchádzajúceho odseku trochu doplniť. Zavedieme zásobník $ZV$, do ktorého dáme $s$ a postupne ukladáme ďalšie objavované vrcholy. Ak v kroku 2 vznikne prípad, že $i = s$, tak obsah zásobníka $ZV$ dáme na výstup: zrejme všetky tieto vrcholy indukujú jeden komponent. Potom $ZV$ znovu napĺňame (začíname novou hodnotou $s$) atď. Každý výstup zo $ZV$ dáva jeden komponent. ●

Poznamenajme, že týmto máme zaručený lineárny algoritmus pre také špeciálne úlohy, ako je zistenie $s$—$t$ sledu pre dané $s$ a $t$, resp. zistenie, či graf je súvislý. Otázkami hľadania optimálnych sledov sa zaoberáme v kap. 4.

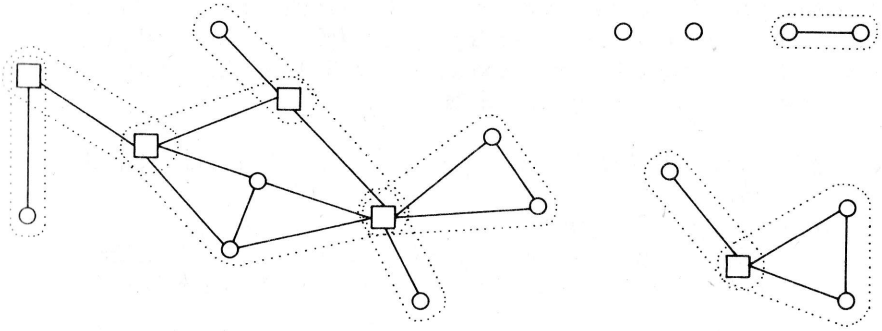




# 3. B
# Bloky a listy

## ARTIKULÁCIE, BLOKY, MOSTY A LISTY

Vrchol $u \in V(G)$ sa nazýva *artikulácia* netriviálneho grafu $G$, ak $G - u$ má väčší počet komponentov, ako má $G$. Podgraf $H$ grafu $G$ sa nazýva *blok*, ak je súvislý, netriviálny, bez artikulácie (vzhľadom na $H$) a je už maximálny s týmito vlastnosťami. Na obr. 1 sú vyznačené artikulácie štvorčekmi a v bodkovaných obálkach sú bloky.



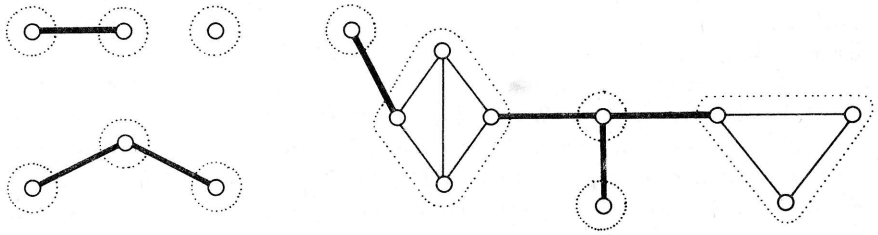
Obr. 1. Bloky a artikulácie.

Hrana $e$ grafu $G$ sa nazýva *most*, ak $G - e$ má väčší počet komponentov, ako má $G$. Podgraf $H \subseteq G$ sa nazýva *list*, ak je súvislý, bez mostov (vzhľadom na H) a už maximálny s týmito vlastnosťami. Na obr. 2 sú mosty vyznačené hrubými čiarami a v bodkovaných obálkach sú listy. Ak uvažujeme súvislý graf reprezentujúci nejakú komunikačnú sieť, tak artikulácie a mosty predstavujú „chúlostivé" miesta tejto siete (stačí, aby niektoré takéto miesto „vypadlo" a poruší sa súvislosť siete). Poznamenajme, že pojmy artikulácia, blok, most a list sa úplne rovnako definujú aj pre pseudografy.

Čitateľ ľahko overí, že graf $K_2$ má most, ale nemá artikuláciu, a že existujú grafy s artikuláciami, ktoré nemajú mosty. Hneď však vidieť, že ak súvislý graf má aspoň 3 vrcholy a má most, tak má aj artikuláciu (všeobecne: ak $uv$ je most a $\deg(u) > 1$, tak $u$ je artikulácia).

Existujú grafy, kde každá hrana je mostom, napr. stromy (všeobecne sú to práve lesy). Platí však:

**Veta 1.** Každý netriviálny súvislý graf má aspoň dva vrcholy, ktoré nie sú artikuláciami.



Obr. 2. Mosty a listy.

DÔKAZ. Nech $S$ je nejaká maximálna (t. j. nepredĺžiteľná) cesta v grafe $G$. Nech $S$ má kraje $u$ a $v$. Ak $G-v$ je nesúvislý, tak $S$ sa dá predĺžiť do nejakého vrcholu $w \in V_G(v)$, ktorý neleží v tom komponente grafu $G$ ako $u$. Tento spor ukazuje, že $v$ nie je artikulácia. Podobne ani $u$. ■

**Veta 2.** Nech $v$ je vrchol súvislého grafu $G$. Potom nasledujúce tvrdenia sú ekvivalentné.

(i) $v$ je artikuláciou grafu $G$.

(ii) Existuje rozklad množiny $V(G)-\{v\}$ na neprázdne podmnožiny $U$ a $W$ tak, že ľubovoľná cesta začínajúca v $U$ a končiaca vo $W$ obsahuje vrchol $v$.

DÔKAZ. (i) $\Rightarrow$ (ii) Graf $G-v$ má aspoň dva komponenty. Ak množinu vrcholov jedného komponentu označíme pomocou $U$ a položíme $W=(V-\{v\})-U$, tak máme požadované tvrdenie.

(ii) $\Rightarrow$ (i) Ihneď z definície. ■

Analogické tvrdenie platí pre mosty.

**Veta 3.** Nech $e$ je hrana súvislého grafu $G$. Potom nasledujúce tvrdenia sú ekvivalentné:

(i) $e$ je mostom v $G$.

(ii) $e$ neleží v žiadnom cykle grafu $G$.

(iii) Existuje rozklad množiny $V(G)$ na neprázdne podmnožiny $U$ a $W$ tak, že každá $U$—$W$ cesta obsahuje hranu e.

DÔKAZ je ľahký a prenechávame ho čitateľovi. ■

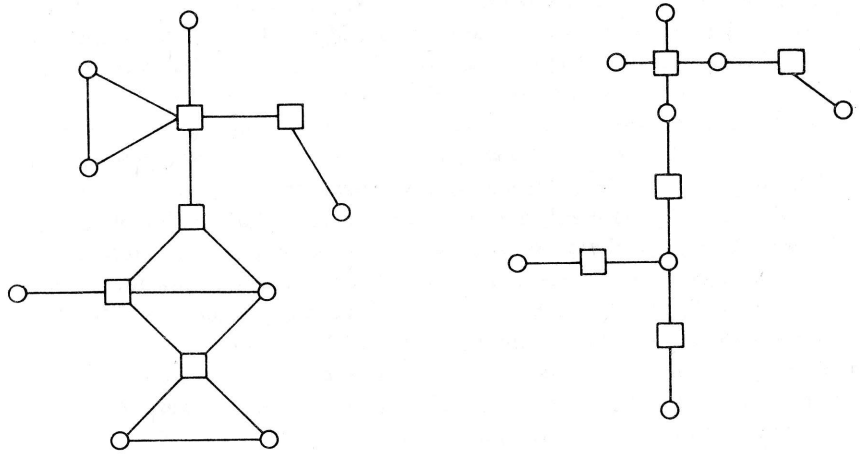
## BLOKOVO-ARTIKULAČNÝ GRAF

Všimneme si niektoré vzťahy blokov a artikulácií. Podobné vzťahy listov a mostov uvádzame v cvičeniach.





V teórii grafov sa definuje k netriviálnemu súvislému grafu $G$ tzv. *blokovo-artikulačný graf* $bc(G)$ ako graf, ktorého vrcholmi sú bloky a artikulácie grafu $G$ a dva vrcholy sú spojené hranou, ak jeden z nich zodpovedá bloku a druhý artikulácii, ktorá leží v tomto bloku. Obr. 3 túto konštrukciu ilustruje. Blokovo-artikulačný graf zaviedol a skúmal Sedláček (1957, 1959), neskoršie Gallai (1962) a i. (chybne už König (1936)).



Obr. 3. Graf a jeho blokovo-artikulačný graf.

Blok môže obsahovať niekoľko artikulácií grafu: ak obsahuje práve jednu artikuláciu, tak sa nazýva *koncový*, ak ani jednu, tak sa nazýva *izolovaný*.

Nasledujúca veta zhŕňa niektoré pozorovania.

**Veta 4.** Nech $G$ je netriviálny súvislý graf. Potom platí:

(1) Každé dva bloky v $G$ majú nanajvýš jeden spoločný vrchol.

(2) Každá hrana grafu $G$ leží práve v jednom bloku.

(3) Každý cyklus má všetky hrany práve v jednom bloku.

(4) Spoločný vrchol dvoch blokov je artikuláciou grafu $G$.

(5) Každá artikulácia grafu $G$ leží aspoň v dvoch blokoch.

(6) Blokovo-artikulačný graf grafu $G$ je strom.

(7) Ak má graf $G$ aspoň dva bloky, tak má aspoň dva koncové bloky.

DÔKAZ. (1) Nech $B_1 \neq B_2$ sú bloky v $G$. Predpokladajme, že $V(B_1) \cap V(B_2)$ obsahuje aspoň dva vrcholy. Potom $B_1 \cup B_2$ je súvislý, netriviálny graf bez artikulácie. Keďže $B_1 \subset B_1 \cup B_2$, tak máme spor s maximalitou bloku $B_1$.

(2) Hrana a jej kraje tvoria netriviálny súvislý podgraf bez artikulácií, a preto leží v nejakom bloku. Z (1) vyplýva, že leží v jedinom bloku.

(3) Samotný cyklus je súvislý, netriviálny podgraf bez artikulácie, a teda celý leží v nejakom bloku grafu $G$. Tvrdenie teraz vyplýva z (2).

(4) Nech vrchol $v$ leží aj v bloku $B_1$ aj v bloku $B_2$. Podľa (1) $B_1$ a $B_2$ viac spoločných vrcholov nemajú. Zvoľme $v_1 \in V(v) \cap V(B_1)$ a $v_2 \in V(v) \cap V(B_2)$. Tvrdíme, že v grafe $G - v$ neexistuje $v_1$—$v_2$ cesta. V opačnom prípade zreťazenie takejto cesty s cestou $v_2 v v_1$ by dávalo cyklus grafu $G$. Hrany tohto cyklu musia podľa (3) ležať v jedinom bloku, čo nie je možné, lebo $v_1 v \in E(B_1)$ a $v_2 v \in E(B_2)$. Teda $G - v$ je nesúvislý, a preto $v$ je artikulácia.

(5) Artikulácia $u$ je krajom nejakej hrany, a preto podľa (2) leží v nejakom bloku $B$. Ak $u$ leží len v jedinom bloku, tak $B - u$ je súvislý (lebo $B$ nemá artikulácie), a teda aj $G - u$ je súvislý, čo je nemožné.

(6) Zrejme $bc(G)$ je súvislý graf. Ukážeme, že neobsahuje cyklus. Nech $bc(G)$ má cyklus $Z$. Podľa definície blokovo-artikulačného grafu v $Z$ sa musia striedať vrcholy $B_i$ zodpovedajúce blokom a vrcholy $a_j$ zodpovedajúce artikuláciám. Preto má $Z$ tvar: $a_1 B_1 a_2 B_2 a_3 \ldots a_t B_t a_1$ ($t \geq 2$). Budeme predpokladať, že $Z$ je cyklus najmenšej dĺžky. Z toho potom vyplýva, že ak $i \neq j \neq i+1 \pmod t$, tak $B_i$ neobsahuje artikuláciu $a_j$ (inak by existoval kratší cyklus: $a_{i+1} B_{i+1} a_{i+2} \ldots a_j B_i a_{i+1}$). Potom môžeme ku $Z$ priradiť cyklus $Z_G$ v grafe $G$ takto: v každom bloku $B_i$ vyberieme nejakú $a_i - a_{i+1}$ cestu $P_i$ a položíme $Z_G = P_1 P_2 \ldots P_t$. Z vlastnosti (1), (2) a z toho, že $a_j \notin B_i$ ak $i \neq j \neq i+1$ vyplýva, že $P_i \neq P_j$ pre $i \neq j$. Takže $Z_G$ je naozaj cyklus a pritom obsahuje hrany z rôznych blokov. To je spor s (3).

(7) Zo (6) máme, že $bc(G)$ je strom a má aspoň tri vrcholy. Takýto strom má aspoň dva koncové (t. j. stupňa 1) vrcholy. Koncové vrcholy grafu $bc(G)$ však musia zodpovedať blokom grafu $G$ ako vyplýva z (5). ■

## VLASTNOSTI BLOKOV

Nasledujúca veta zhŕňa viaceré známe vlastnosti blokov (porovnaj [Harary (1969)]).

**Veta 5.** Nech $G$ je súvislý graf s aspoň tromi vrcholmi. Potom nasledujúce tvrdenia sú ekvivalentné.

(1) $G$ je blok.

(2) Ľubovoľné dva vrcholy grafu $G$ ležia na spoločnom cykle.

(3) Ľubovoľný vrchol a ľubovoľná hrana grafu $G$ ležia na spoločnom cykle.

(4) Ľubovoľné dve hrany grafu $G$ ležia na spoločnom cykle.

(5) Pre ľubovoľné dva vrcholy $u$, $v$ a ľubovoľnú hranu $e$ grafu $G$ existuje $u$—$v$ cesta obsahujúca hranu $e$.

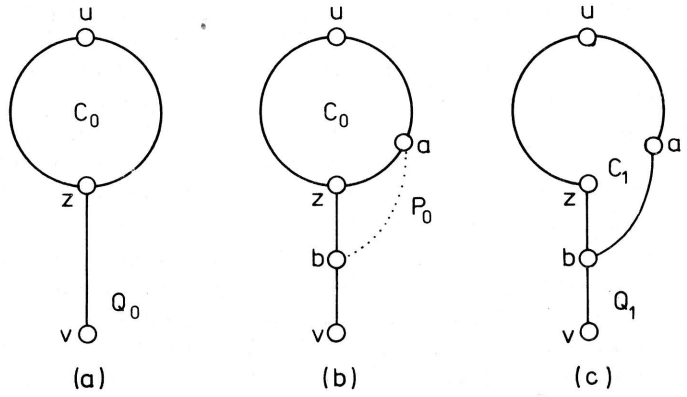




(6) Pre každé tri vrcholy $u$, $v$, $w$ grafu $G$ existuje $u$—$w$ cesta obsahujúca vrchol $v$.

(7) Pre každé tri vrcholy $u$, $v$, $w$ grafu $G$ existuje $u$—$w$ cesta neobsahujúca vrchol $v$.

DÔKAZ. (1) $\Rightarrow$ (2) Nech $u, v \in V(G)$. Keďže $G$ má aspoň tri vrcholy a nemá artikuláciu, tak nemá ani most. Preto každá hrana leží v cykle (veta 3). Takto vrchol $u$ leží v nejakom cykle $C_0$. Ak leží $v$ na $C_0$, niet čo dokazovať. Preto nech je mimo $C_0$. Graf $G$ je súvislý, a preto existuje cesta od cyklu $C_0$ k vrcholu $v$ : nech je to $z$—$v$ cesta $Q_0$. Situáciu znázorňuje obr. 4(a). Ukážeme, že existuje cyklus $C_1$ obsahujúci $u$ a cesta $Q_1$ od neho ku $v$, pričom $m(Q_1') < m(Q_0)$. Skutočne, vrchol $z$ nie je artikuláciou, a preto v grafe $G - z$ existuje cesta $P_0$ od $C_0$—$z$ ku $Q_0$—$z$ ; nech je to $a$—$b$ cesta (pozri obr. 4(b)). Z podgrafu tvoreného cyklom $C_0$ a cestami $Q_0$ a $P_0$ už ľahko vyberieme požadovaný cyklus $C_1$ a cestu $Q_1$ tak, ako na obr. 4(c). Ak $m(Q_1) = 0$, tak $v$ leží na cykle $C_1$ a dôkaz je hotový. V opačnom prípade tento postup zopakujeme a nájdeme $C_2$ a $Q_2$, kde $m(Q_2) < m(Q_1)$ atď., až získame cyklus $C_i$ obsahujúci $u$ aj $v$.




Obr. 4. Skrátenie cesty $Q_0$.

(2) $\Rightarrow$ (3) Nech je daný vrchol $u$ a hrana $vz$. Podľa predpokladu existuje cyklus $C_0$ obsahujúci $u$ a vrchol $z$. Ak vrchol $v$ leží na cykle $C_0$, tak cyklus obsahujúci vrchol $u$ a hranu $zv$ sa vyberie veľmi ľahko. Zostáva prípad, kedy vrchol $v$ je mimo cyklu $C_0$. To je situácia z obr. 4(a), pričom $m(Q_0) = 1$. Keďže z (2) vyplýva, že $G$ je blok, tak použijeme predchádzajúcu konštrukciu, ktorá je na obr. 4(b) a 4(c), pričom teraz $b = v$. Keďže $C_1$ obsahuje úsek $z - b$ cesty $Q_0$, tak teraz to značí, že $C_1$ obsahuje hranu $zb = zv$, a teda $C_1$ je požadovaný cyklus.

(3) $\Rightarrow$ (4) Nech sú dané hrany $ut$ a $zv$. Podľa predpokladu existuje cyklus $C_0$ obsahujúci hranu $ut$ a vrchol $z$. Ak $v$ leží na cykle $C_0$, hľadaný cyklus sa nájde ľahko. Ak vrchol $v$ leží mimo $C_0$, tak zase možno použiť konštrukciu z obr. 4 a hneď získame požadovaný cyklus $C_1$ (keďže zrejme (3) $\Rightarrow$ (1)).

(4) $\Rightarrow$ (5) Ak existuje hrana $uv$, tak existuje cyklus $C$ obsahujúci hranu $uv$ a hranu $e$. Zrejme $C-uv$ je požadovaná $u$—$v$ cesta obsahujúca $e$. Ak $uv \notin E(G)$, tak uvažujeme graf $G+uv$. Ak $G+uv$ spĺňa (4), tak úplne rovnako zaručíme požadovanú cestu. Avšak $G$ spĺňa (4), a teda aj (2). Potom aj $G+uv$ spĺňa (2), a teda podľa dokázaných implikácií aj (4).

(5) $\Rightarrow$ (6) Zvolíme hranu $e$ incidujúcu s vrcholom $v$. Podľa (5) existuje $u$—$w$ cesta obsahujúca $e$, a teda aj vrchol $v$.

(6) $\Rightarrow$ (7) Podľa (6) existuje $u$—$v$ cesta obsahujúca vrchol $w$. Potom $u$—$w$ úsek tejto cesty neobsahuje vrchol $v$.

(7) $\Rightarrow$ (1) Žiadny vrchol $v$ nemôže byť artikuláciou, lebo pre ľubovoľné dva vrcholy $u$, $w$ existuje $u$—$w$ cesta neobsahujúca $v$, a teda $G-v$ je súvislý graf. ■

## HĽADANIE BLOKOV A ARTIKULÁCIÍ

Keďže už máme lineárny algoritmus na zisťovanie komponentov grafu, ľahko možno vyrobiť polynomiálny algoritmus pre hľadanie artikulácií a blokov:

Pre každý netriviálny komponent $H$ a každý vrchol $u \in V(H)$ zistíme, či $H-u$ je súvislý graf. Ak nie je, tak $u$ je artikulácia: v opačnom prípade $u$ nie je artikuláciou. Blok zasa možno generovať takto: Zoberieme nejakú hranu $uv$ a hľadáme (napr. postupom do hĺbky z koreňa $u$) nejakú $u$—$v$ cestu v grafe $H-uv$. Ak takáto cesta $P$ existuje, tak spolu s hranou $uv$ vytvára cyklus $B$. Potom pre každý vrchol $u \in V(B)$, ktorý nie je artikuláciou v $H$, zoberieme susedný vrchol $v$ mimo $B$ a v grafe $H-u$ nájdeme nejakú cestu z $v$ do $V(B)-u$: ak existuje, tak jej vrcholy a hrany pridáme k $B$ a označíme symbolom $B$. Takto zrejme vyrobíme podgraf $B$, ktorého vrcholy indukujú blok v $H$.

Avšak takýto algoritmus je dosť zložitý. Tu načrtneme $O(m+n)$ algoritmus. Takýto algoritmus dáva Tarjan (1972) (pozri tiež [Hopcroft a Tarjan (1973)]) a tiež Dinic, Zajcev a Karzanov (1974).

Nech vrchol $v$ je artikuláciou súvislého grafu $G$. Potom maximálny súvislý podgraf $H$ grafu $G$ obsahujúci vrchol $v$, v ktorom však $v$ nie je artikuláciou, nazýva sa *vetva* grafu $G$ pri artikulácii $v$. Postup do hĺbky (Trémauxov prieskum) má takúto peknú vlastnosť: Ak vojdeme do nejakej vetvy pri $v$, tak z tejto vetvy vyjdeme až vtedy, ak ju celú pochodíme (každú hranu oboma smermi). Špeciálne, ak vojdeme do koncového bloku, tak ho opustíme iba vtedy, ak už je celý preskúmaný. Túto zrejmú vlastnosť využijeme na hľadanie blokov a artikulácií.

Základný postup do hĺbky doplníme o ďalšie záznamy. Používame zásobník $ZE$,





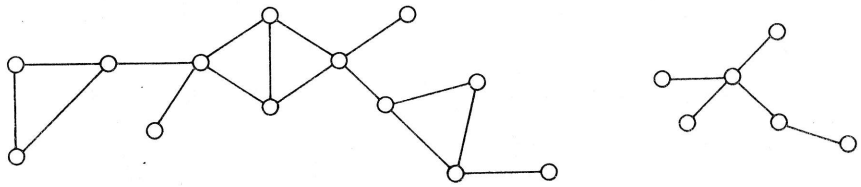
do ktorého postupne ukladáme objavované hrany. Vždy, keď preskúmame koncový blok, tak hrany tohto bloku dáme zo $ZE$ na výstup a pokračujeme v prieskume zvyšného grafu, ktorého koncový blok dáme po preskúmaní opäť na výstup atď. Na identifikáciu artikulácií a koncových blokov zaznamenávame pre každý objavený vrchol $u$ tzv. *dolné číslo* $r(u)$, čo je najmenšie *poradové číslo* $p(v)$ vrcholu $v$, pre ktorý existuje vo vytváranej palme $u$—$v$ cesta, pozostávajúca zo stromových hrán nasledovaných nanajvýš jednou spätnou hranou. Taká cesta môže byť aj triviálna, a preto pri objavení vrcholu $u$ kladieme $r(u) := p(u)$. Takto vždy $r(u) \leq p(u)$. Ak nájdeme spätnú hranu odchádzajúcu z $u$, napr. šíp $ux$, tak kladieme: $r(u) := \min\{r(u), p(x)\}$. Okrem toho, ak sa v prieskume vraciame, t. j. z vrcholu $w$ do $u$, kde $f(w) = u$, tak kladieme $r(u) := \min\{r(u), r(w)\}$. Tieto dve možnosti opravovania $r(u)$ nám zrejme vystačia na to, aby sme po úplnom preskúmaní vrcholu $u$ (t. j. ešte pred návratom do $f(u)$) vypočítali dolné číslo $r(u)$.

Uvažujme teraz situáciu, kedy sme ukončili prieskum nejakého vrcholu $y$ a budeme sa vracať do $f(y) = x$. Ak $r(y) \geq p(x)$, tak postupne vynecháme zo $ZE$ a prenesieme na výstup hrany od poslednej až po $xy$ vrátane: táto skupina hrán tvorí jeden blok (vždy koncový, resp. izolovaný v grafe bez tých blokov, ktoré sme už prv takto vynechali). Ak je navyše vrchol $x$ rôzny od koreňa $s$, tak $x$ je artikuláciou. Ak $x$ je koreňom a po skončení celého prieskumu z neho odchádzajú aspoň dve stromové hrany, tak je artikuláciou. Prenechávame čitateľovi nahliadnuť (3.Z.4(a)), že takto doplnený postup do hĺbky správne nájde bloky aj artikulácie v čase $O(m+n)$. ●

Teraz je už ľahké zostrojiť k danému grafu blokovo-artikulačný graf (3.Z.4(a)).

## HĽADANIE MOSTOV A LISTOV

Možno navrhnúť modifikáciu predchádzajúceho postupu pre zisťovanie mostov a artikulácií. Iná možnosť je transformovať túto úlohu na hľadanie blokov a artikulácií tak, že do každej hrany vsunieme jeden nový vrchol. Potom nový vrchol je artikuláciou práve vtedy, ak príslušná hrana je mostom v pôvodnom grafe. Listy sa tiež ľahko identifikujú (3.Z.4(b)).



Obr. 5. Graf a listová kondenzácia.

Istou analógiou blokovo-artikulačného grafu je tzv. *listová kondenzácia grafu*, ktorej vrcholmi sú listy a dva vrcholy sú spojené hranou, ak príslušné listy boli spojené hranou. Z vlastností listov a mostov hneď vidíme, že listová kondenzácia súvislého grafu je strom. Obr. 5 ilustruje tento pojem.

Opäť prenechávame čitateľovi navrhnúť $O(m+n)$ algoritmus pre nájdenie listovej kondenzácie (3.Z.4(b)).

## 3. C
## Najmenšie doplnenie grafu na list alebo blok

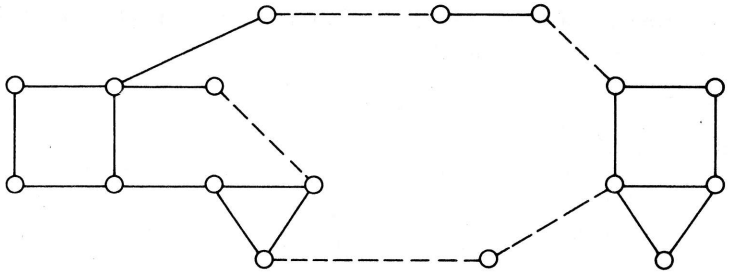
Nech je daný graf $G$ s $k$ komponentmi. Aký je minimálny počet hrán, ktoré treba pridať ku $G$ tak, aby sme získali súvislý graf? Odpoveď je zrejmá: treba pridať $k-1$ hrán (lebo strom na $k$ vrcholoch má $k-1$ hrán). Zovšeobecnenie tejto úlohy, kde dopĺňame ohodnotenými hranami a žiadame, aby súčet ohodnotení pridaných hrán bol minimálny, vieme tiež riešiť. Ide o problém najlacnejšej kostry a venujeme sa mu v kap. 5. Tu si všimneme dopĺňanie grafu na list, resp. blok.

### NAJMENŠIE DOPLNENIE GRAFU NA LIST

Nech je daný graf $G$, ktorý nie je list. Aký je najmenší počet hrán, ktoré treba pridať ku $G$ tak, aby vzniknutý multigraf $\hat{G}$ bol list? Budeme hovoriť, že $\hat{G}$ je *najmenší list obsahujúci* $G$. (Takáto úloha môže vzniknúť pri rozširovaní komunikačnej siete.) Obr. 1 ukazuje príklad najmenšieho doplnenia (čiarkované hrany).

Taký list grafu, ktorý neinciduje so žiadnym mostom, nazýva sa *izolovaný* a taký, ktorý inciduje práve s jedným mostom, nazýva sa *koncový*.

Nasledujúce tvrdenie možno nájsť v prácach [Eswaran a Tarjan (1976)] a [Plesník (1976)]. Príslušné dôkazy predstavujú zároveň polynomiálne algoritmy.



Obr. 1. Najmenšie doplnenie na list.





Tu uvádzame dôkaz z prvej práce, ktorý predstavuje dokonca lineárny algoritmus na doplnenie grafu na list.

**Veta 1.** Nech netriviálny graf $G$ nie je list a nech počet jeho koncových listov je $p$ a izolovaných $q$. Potom pre najmenší počet hrán $a(G)$, ktorými možno doplniť $G$ na list $\hat{G}$, platí: $a(G)=\lceil p/2\rceil+q$.

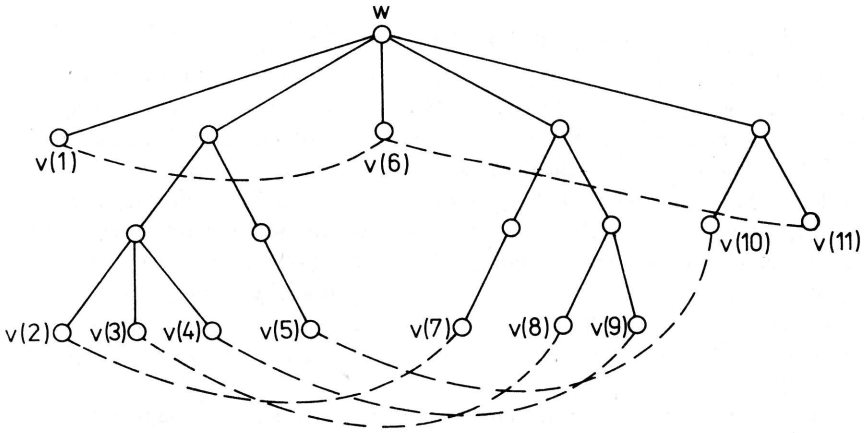
DÔKAZ. Najprv dokážeme, že $a(G)\geq\lceil p/2\rceil+q$. Každý izolovaný list grafu $G$ musí byť v $\hat{G}$ incidentný aspoň s dvoma novými hranami a každý koncový list aspoň s jednou novou hranou. Preto súčet stupňov v podgrafe multigrafu $\hat{G}$ pozostávajúcom z nových hrán je aspoň $2q+p$, takže nových hrán je aspoň $q+\lceil p/2\rceil$.

Teraz dokážeme obrátenú nerovnosť.

(1) K danému grafu $G$ s $k$ komponentmi, $p$ koncovými a $q$ izolovanými blokmi zostrojíme listovú kondenzáciu $G_0$. Zrejme $G_0$ má $k$ komponentov, $p$ koncových a $q$ izolovaných vrcholov.

(2) Les $G_0$ doplníme pomocou $k-1$ nových hrán na strom $T_0$ s $p_0=p+2q-2(k-1)$ koncovými vrcholmi. To možno urobiť vhodným zapojením stromov $T_i$ lesa $G_0$ „do série“: v každom netriviálnom strome $T_i$ zvolíme dva rôzne koncové vrcholy $u_i$, $v_i$ a v triviálnom strome $u_i=v_i$: potom pridáme hrany $u_1v_2$, $u_2v_3$, ..., $u_{k-1}v_k$.

(3) Strom $T_0$ doplníme pomocou $\lceil p_0/2\rceil$ nových hrán na list takto: Ak $T_0$ má iba dva vrcholy, úloha je triviálna. Inak nech $w$ je nejaký nekoncový vrchol stromu




Obr. 2. Doplnenie stromu na list.

$T_0$. Ďalej nech $v(1)$, $v(2)$, ..., $v(p_0)$ sú všetky koncové vrcholy stromu $T_0$ v tom poradí, v akom ich objavujeme postupom do hĺbky, začínajúcom vo $w$ (pozri príklad na obr. 2). Potom k $T_0$ pridávame nové hrany $v(i)v(i+\lfloor p_0/2 \rfloor)$ pre $i = 1, 2, \ldots, \lfloor p_0/2 \rfloor$ (porovnaj obr. 2). Vzniknutý graf $H$ bude bez mostov, a teda list. Overíme si to. Každá nová hrana spája dva vrcholy stromu, a teda leží v cykle. Uvažujme preto nejakú stromovú hranu $uv$. Nech $u$ je otcom vrcholu $v$. Predpokladajme, že $uv$ je mostom v $H$. Potom $H - uv$ má dva komponenty: $C_u$ obsahujúci $u$ a $C_v$ obsahujúci $v$. Z postupu do hĺbky vyplýva, že koncové vrcholy stromu $T_0$ ležiace v $C_v$ tvoria nejakú postupnosť $v(s)$, $v(s+1)$, ..., $v(t)$. Tieto vrcholy by mohli byť pospájané len takými novými hranami, ktorých oba kraje ležia v tejto postupnosti. To však zrejme vedie ku sporu s rozmiestnením nových hrán, ak $t \leq \lfloor p_0/2 \rfloor$ alebo ak $s > \lfloor p_0/2 \rfloor$. Ak $s \leq \lfloor p_0/2 \rfloor$ a súčasne $t > \lceil p_0/2 \rceil$, tak máme tiež spor, okrem prípadu $s = 1$ a $t = p_0$, čo však nie je možné, lebo vrchol $w$ má stupeň aspoň 2.

(4) Konečne, od listu $H$ prejdeme k listu $\hat{G}$ obsahujúcemu graf $G$. Každú novú hranu $xy$ v $H$ možno realizovať tak, že v listoch $L_x$ a $L_y$ zodpovedajúcich v listovej kondenzácii vrcholom $x$ a $y$ zvolíme po jednom vrchole (ľubovoľne) a tieto spojíme hranou. Je zrejmé, že vzniknutý graf $\hat{G}$ bude listom obsahujúcim $G$ a počet nových hrán je $k - 1 + \lfloor p_0/2 \rfloor$. Podľa (2) však $k - 1 = (2q + p - p_0)/2$, takže počet nových hrán je $q + p/2 - p_0/2 + \lfloor p_0/2 \rfloor = q + \lceil p/2 \rceil$. ■

Keďže jednotlivé kroky (1) až (4) vieme urobiť v čase $O(m+n)$, tak máme $O(m+n)$ algoritmus pre nájdenie najmenšieho listu obsahujúceho daný graf.

## NAJMENŠIE DOPLNENIE GRAFU NA BLOK

Nech je daný graf $G$. Aký je minimálny počet hrán, ktorými treba $G$ doplniť tak, aby získaný graf $\hat{G}$ bol blokom? $\hat{G}$ sa nazýva *najmenší blok obsahujúci G.* Týmto problémom sa zaoberali viacerí autori. Čiastočný úspech oznámil Slater (1975). Úplné riešenie dávajú Eswaran a Tarjan (1976), Plesník (1976) a Rosenthal a Goldner (1977), v ktorých sa nachádza aj doleuvedená veta.

Najprv zavedieme niekoľko pojmov a označení. Komponent $C$ grafu $G$ sa nazýva *izolovaný*, ak nemá koncové bloky. Nech $e_1(G)$ označuje počet koncových blokov grafu $G$, $e_2(G)$ počet izolovaných komponentov a $c(G)$ počet všetkých komponentov grafu $G$. Kladieme $e(G) := e_1(G) + 2e_2(G)$. Počet blokov grafu $G$ obsahujúcich daný vrchol $v$ označujeme symbolom $b(v)$ a $b(G) := \max \{b(v) \mid v \in V(G)\}$.

*Vetva* grafu $G$ pri artikulácii $u$ je maximálny súvislý podgraf grafu $G$ obsahujúci $u$, v ktorom $u$ nie je artikuláciou. *Prút* je vetva $T$, kde $e(T) = 2$.





**Veta 2.** Nech $G$ je graf s aspoň tromi vrcholmi a nech nie je blokom. Potom pre minimálny počet $f(G)$ nových hrán, ktorými možno $G$ doplniť na blok $\hat{G}$, platí:

$$f(G) = \max \{b(G) + c(G) - 2, \lceil e(G)/2 \rceil\}.$$

DÔKAZ. Nech $h(G)$ označuje pravú stranu predchádzajúceho výrazu. Najprv dokážeme, že $f(G) \geq h(G)$.

Graf $\hat{G} - G := \hat{G} - E_G$ má $f(G)$ hrán. Každý izolovaný komponent grafu $G$ inciduje v $\hat{G}$ aspoň s dvoma novými hranami a každý koncový blok aspoň s jednou novou hranou. Teda súčet stupňov vrcholov grafu $\hat{G} - G$ je aspoň $e(G)$, takže $f(G) \geq \lceil e(G)/2 \rceil$.

Ak $b(G) \leq 1$, tak $G$ nemá artikulácie, a preto má len izolované komponenty. Potom však $b(G) + c(G) - 2 < e_2(G) = \lceil e(G)/2 \rceil$. Nech teda $b(G) \geq 2$. Potom $G$ má nejakú artikuláciu $u$ obsiahnutú v $b(G)$ blokoch. Takto graf $G - u$ má $c(G) + b(G) - 1$ komponentov, a teda treba aspoň $c(G) + b(G) - 2$ nových hrán na ich spojenie do súvislého grafu. Graf $\hat{G} - u$ je súvislý, a preto $f(G) \geq c(G) + b(G) - 2$.

Teraz dokážeme, že $f(G) \leq h(G)$. Pre $h(G) = 1$ je to ľahké. Preto nech $h(G) \geq 2$. Dokazujeme indukciou podľa $h(G)$ a uvážime niekoľko prípadov a podprípadov.

(1) Ak je graf $G$ nesúvislý, zoberieme ľubovoľné dva komponenty a v každom z nich zvolíme jeden vrchol rôzny od artikulácie a ležiaci v koncovom bloku alebo v izolovanom komponente. Po spojení týchto dvoch vrcholov novou hranou pre získaný graf $G'$ zrejme platí: $h(G') = h(G) - 1$. Keďže podľa indukčného predpokladu $f(G') = h(G')$, tak aj $f(G) = h(G)$.

(2) Ak graf $G$ je súvislý, tak $e_2(G) = 0$ a $c(G) = 1$. Rozlíšime tri prípady.

(2.1) $b(G) - 1 > \lceil e_1(G)/2 \rceil$. Potom existuje jediný vrchol $u$ taký, že $b(u) = b(G)$ (inak by existovalo aspoň $2(b(G) - 1)$ koncových blokov). Zvoľme dve vetvy pri $u$ a v každej z nich jeden vrchol ležiaci v koncovom bloku grafu $G$ a rôzny od artikulácie grafu $G$. Spojením týchto dvoch vrcholov novou hranou získame graf $G'$ s $b(G') = b(G) - 1$ a $e_1(G') \leq e(G) - 1$. Takto $h(G') = h(G) - 1$ a zasa môžeme použiť indukciu.

(2.2) $b(G) - 1 = \lceil e_1(G)/2 \rceil$. Ak $b(G) = 2$, tak $e_1(G) = 2$ a $f(G) = 1$. Preto nech $b(G) \geq 3$. Potom $G$ má nanajvýš dva také vrcholy $u_1$, $u_2$, že $b(u_1) = b(u_2) = b(G)$ (inak by existovalo aspoň $3[b(G) - 1] - 1$ koncových blokov). Uvažujme nejakú vetvu $T$ pri $u_1$, ktorá má aspoň dva koncové bloky grafu $G$ (ak taká vetva neexistuje, tak $b(u_1) = 3$, $e_1(G) = 3$ a prípad je ľahký: stačí spojiť novou hranou dva vrcholy z koncových blokov rôzne od artikulácií). Ak existuje aj $u_2$ s $b(u_2) = b(G)$, tak vetva $T$ obsahuje $u_2$ (inak by graf $G$ mal viac koncových

blokov, ako predpokladáme) a potom už pre každé $u \in V(G) - \{u_1, u_2\}$ platí: $b(u) \leq 2$. Spojme ľubovoľný koncový blok grafu $G$ obsiahnutý v $T$ s ľubovoľným koncovým blokom grafu $G$ z inej vetvy pri $u_1$ novou hranou tak, že spojíme nejaké dva vrcholy týchto blokov, ktoré nie sú artikuláciami v $G$. Týmto získame graf $G'$ s $e_1(G') = e_1(G) - 2$ (nový blok obsahujúci priradenú hranu nebude v $G'$ koncový, lebo inciduje s aspoň dvoma artikuláciami). Ďalej, $b(G') = b(G) - 1$, lebo pridaním novej hrany sa počet vetiev pri $u_1$ (a ak $b(u_2) = b(G)$, tak aj pri $u_2$) zmenší o 1. Takto opäť $h(G') = h(G) - 1$.

(2.3) $b(G) - 1 < \lceil e_1(G)/2 \rceil$. Nech $u$ je taký vrchol, že $b(u) = b(G)$. Z nášho predpokladu vyplýva, že $G$ obsahuje nejaký vrchol $v \neq u$, kde $b(v) \geq 3$. Zvoľme nejaký koncový blok grafu $G$ z vetvy pri $u$ a nejaký koncový blok grafu $G$ z inej vetvy pri $u$. Spojením vybraných blokov novou hranou tak, že spojíme dva vrcholy rôzne od artikulácií grafu $G$, získame graf $G'$ s $e_1(G') = e_1(G) - 2$ a $b(G') \leq b(G)$. Preto opäť $f(G') = f(G) - 1$.

Tým sú prebraté všetky možnosti a veta je dokázaná. ■

Uvedený dôkaz zrejme poskytuje polynomiálny algoritmus pre nájdenie najmenšieho bloku obsahujúceho $G$. Získať však lineárny algoritmus nie je jednoduché a vyžaduje dômyselnú manipuláciu s dátami. Čitateľa odkazujeme na článok [Rosenthal a Goldner (1977)], kde je taký algoritmus uvedený.

## ZLOŽITOSŤ DOPĹŇANIA V OHODNOTENÝCH GRAFOCH

V reálnych úlohách zvyčajne nezáleží na počte nových liniek, ale je dôležitá ich sumárna cena. Preto v grafoch s ohodnotenými hranami zovšeobecníme uvedené problémy takto: Graf treba doplniť na list (resp. blok) tak, aby súčet ohodnotení pridaných hrán bol najmenší. Takéto doplnenie nazývame *najlacnejšie doplnenie.* Nasledujúci výsledok možno nájsť v [Eswaran a Tarjan (1976)].

**Veta 3.** Problém najlacnejšieho doplnenia grafu na list aj problém najlacnejšieho doplnenia grafu na blok sú NP-ťažké.

DÔKAZ. Problém nájdenia hamiltonovského cyklu v grafe $G$ polynomiálne transformujeme na uvedené problémy. Nech $H$ je nulový graf na množine vrcholov $V_G$, ktorý treba najlacnejšie doplniť na list, resp. blok, ak cena každej hrany $ij$ z komplementu grafu $H$ je 1, pre $ij \in E_G$ a 2, kde $ij \notin E_G$. Zrejme najlacnejšie doplnenie grafu $H$ na list, resp. blok, má cenu $n = |V_G|$, práve vtedy, keď $G$ má hamiltonovský cyklus. ■

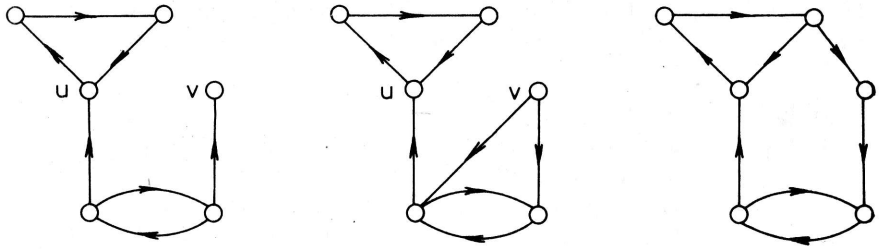




## 3. D
## Rôzne druhy súvislosti digrafov

Digraf sa nazýva *súvislý* alebo tiež *slabo súvislý,* ak pre každé jeho dva vrcholy $u$, $v$ existuje $u$—$v$ polosled. V tomto prípade sa môže stať, že neexistuje $u$—$v$ sled a dokonca ani $v$—$u$ sled. Digraf sa nazýva *jednostranne súvislý,* ak pre každé jeho dva vrcholy $u$, $v$ existuje $u$—$v$ sled alebo $v$—$u$ sled. Napokon digraf sa nazýva *silne súvislý*, ak pre každé jeho dva vrcholy $u$, $v$ existuje $u$—$v$ sled aj $v$—$u$ sled. (Úplne rovnako sa definujú tieto pojmy aj pre pseudomigraf.) Na obr. 1 sú príklady digrafov s rôznymi súvislosťami.

Uvádzame niekoľko základných pozorovaní známych z teórie grafov (pozri napr. [Harary (1969)]).




Obr. 1. Slabo, jednostranne a silne súvislé digrafy.

**Veta 1.** (a) Digraf $G$ je súvislý $\Leftrightarrow$ $G$ má polosled obsahujúci všetky vrcholy.

(b) Digraf $G$ je jednostranne súvislý $\Leftrightarrow$ $G$ má sled obsahujúci všetky vrcholy.

(c) Digraf $G$ je silne súvislý $\Leftrightarrow$ $G$ má uzavretý sled obsahujúci všetky vrcholy (tzv. *hamiltonovský sled*).

(d) Digraf $G$ je silne súvislý $\Leftrightarrow$ $G$ je súvislý a každá hrana leží v nejakom cykle.

DÔKAZ. V tvrdeniach (a), (b), (c) z existencie takého polosledu, resp. sledu, hneď vyplýva príslušná súvislosť, a preto dokážeme len obrátené implikácie. Nech je daný digraf $G$ s $V(G) = \{v_1, v_2, \ldots, v_n\}$.

(a) Pre každú dvojicu vrcholov $v_i$, $v_{i+1} \in V(G)$ existuje $v_i - v_{i+1}$ polosled a zreťazenie všetkých takýchto polosledov dá požadovaný polosled.

(b) Priraďme ku $G$ digraf $G'$ s $V(G') = V(G)$, pričom $v_iv_j \in E(G')$, ak v $G$ existuje $v_i - v_j$ sled. Keďže $G'$ obsahuje faktor, ktorý je turnajom, tak má cestu $D$ obsahujúcu všetky vrcholy (tzv. *hamiltonovskú cestu*), ako sa ľahko zistí indukciou (1.Z.1). Po nahradení každej hrany $v_iv_j$ v $D$ $v_i$—$v_j$ sledom v $G$ získame požadovaný sled.

(c) Podľa (b) existuje nejaký $v_p$—$v_q$ sled obsahujúci všetky vrcholy, ktorý s $v_q$—$v_p$ sledom dá požadovaný uzavretý sled.

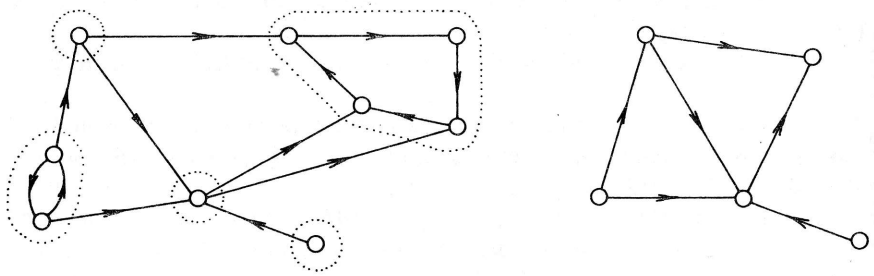
(d) Pre každú hranu $uv$ existuje $v$—$u$ sled, a teda aj $v$—$u$ cesta, ktorá s $uv$ dáva cyklus. Obrátene, podľa (a) existuje nejaký $u$—$w$ polosled $S$ obsahujúci všetky vrcholy. Ak namiesto každej hrany $v_iv_j$ v $S$ dáme ten z dvoch úsekov $v_iv_j$ a $Z_{ij} - v_iv_j$ cyklu $Z_{ij}$ obsahujúceho hranu $v_iv_j$, ktorý je orientovaný v smere polosledu $u$—$w$, tak získame $u$—$w$ sled obsahujúci všetky vrcholy. Berúc zvyšné úseky cyklov, získame $w$—$u$ sled, ktorý s predchádzajúcim dáva hamiltonovský sled. Tvrdenie teraz vyplýva z (c). ■

Rôzne súvislosti digrafov dávajú podklad pre definície rôznych komponentov. Maximálny slabo (resp. jednostranne, resp. silne) súvislý podgraf daného digrafu sa nazýva *slabý (jednostranný, silný) komponent.* Najväčší význam majú silné komponenty digrafu. Hneď je vidieť, že *každý vrchol digrafu leží práve v jednom silnom komponente* a že *rozklad na silné komponenty je jednoznačný.*

## KONDENZÁCIA DIGRAFU (DIGRAF SILNÝCH KOMPONENTOV)

Štúdium súvislostí digrafu $G$ uľahčuje tzv. digraf silných komponentov alebo tiež *kondenzácia* $G^*$, čo je digraf, ktorého vrcholmi sú silné komponenty $S_1$, $S_2$, ..., $S_k$ a hrana $S_iS_j$ existuje práve vtedy, keď v pôvodnom digrafe $G$ existovala aspoň jedna hrana $v_iv_j$, kde $v_i \in V(S_i)$ a $v_j \in V(S_j)$. Obr. 2 ilustruje zostrojenie kondenzácie digrafu.

**Veta 2.** Kondenzácia každého digrafu je acyklický digraf.



Obr. 2. Digraf a kondenzácia.

DÔKAZ. Ak kondenzácia obsahuje cyklus, tak pôvodný digraf obsahuje cyklus prechádzajúci aspoň dvoma silnými komponentmi. To však nie je možné, lebo





samotný cyklus je silne súvislý podgraf, a teda celý leží v jedinom silnom komponente. ■

Zaujímavú aplikáciu uvedených pojmov pri skúmaní matíc dáva Harary (1962). Predpokladajme, že danú štvorcovú maticu $Q$ s prvkami $q_{ij}$ sa nám podarí permutovaním riadkov a súčasným permutovaním zodpovedajúcich stĺpcov previesť na tzv. *blokovo-trojuholníkový tvar.* Potom vlastné čísla všetkých diagonálnych blokov sú vlastné čísla matice $Q$ a z inverzií jednotlivých diagonálnych blokov možno určiť inverznú maticu $Q^{-1}$. Ak matica $Q$ je pomerne riedka (má málo nenulových prvkov) možno očakávať, že získané diagonálne bloky budú malé, a preto tento postup môže byť výhodný. Požadovaný blokovo-trojuholníkový tvar možno nájsť takto: (1) Zostrojíme digraf $G$, ktorý má šíp $ij$ práve vtedy, keď $q_{ij} \neq 0$, $i \neq j$. (2) Určíme silné komponenty digrafu $G$. (3) Zostrojíme kondenzáciu $G^*$. (4) Monotónne očíslujeme vrcholy v $G^*$. (5) Potom podľa toho očíslujeme aj vrcholy digrafu $G$ tak, aby jeho susedská matica $A$ bola horná blokovo-trojuholníková. (6) Konečne v matici $A$ nahradíme jednotkové prvky a diagonálne prvky zodpovedajúcimi (podľa (1)) prvkami z matice $Q$. Kroky (1), (5) a (6) sú ľahké. Lineárne algoritmy pre kroky (2), (3) a (4) uvedieme v nasledujúcej časti.

## 3. E
## Prieskum digrafov

### TARRYHO PRIESKUM DIGRAFOV

Prieskum digrafov je analogický prieskumu grafov. Tu generujeme polosled začínajúci v nejakom vrchole $s$, pričom hranami môžeme prechádzať v oboch smeroch zachovávajúc nasledujúce pravidlo (TD) a pravidlá (T1), (T2) známe z prieskumu grafov (časť 3.A).

(TD) Prvýkrát môžeme hranou prechádzať iba v smere šípky.

(T1) Každou hranou môžeme v jednom smere prechádzať nanajvýš raz.

(T2) Po tej hrane, po ktorej sme prišli do nejakého vrcholu prvýkrát, môžeme ísť späť iba vtedy, ak niet inej možnosti. ●

Poznamenajme, že Tarry sa nezaoberal orientovaným bludiskom, ale rozšírenie jeho metódy na orientovaný prípad je jednoduché. Tiež máme analogickú vetu.

**Veta 1.** (a) Tarryho prieskum digrafov je konečný.

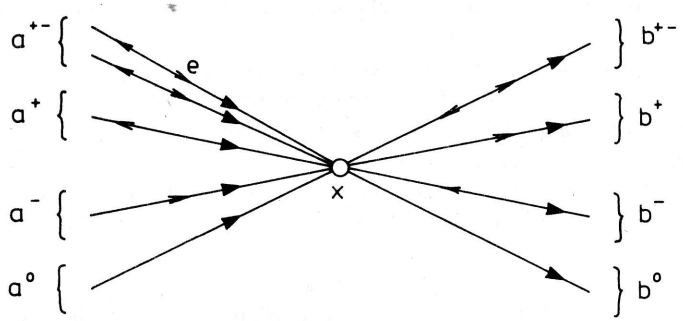
(b) Ak pravidlá (TD), (T1) a (T2) nedovoľujú vytvoriť dlhší polosled, tak posledný vrchol polosledu je začiatok $s$. Tento polosled obsahuje všetky vrcholy

dosiahnuteľné z $s$ a všetky hrany odchádzajúce z týchto vrcholov sú prejdené oboma smermi. Naviac, niet hrany, ktorá by bola prejdená iba raz.

DÔKAZ. (a) Hrán je konečne mnoho a každou možno prejsť nanajvýš 2 razy (pravidlo (T1)).

(b) Nech je utvorený $s$—$x$ polosled $Q$, ktorý už nemožno predĺžiť. Predpokladajme, že $x \neq s$ a nech $e$ je hrana, ktorou sme prišli do $x$ prvýkrát. Ak už niet iných možností, tak možno hranu $e$ prejsť aj v smere od $x$. Keďže stojíme v $x$, tak už bola táto možnosť použitá. Uvážme situáciu, ihneď po tomto spätnom prejdení hranou $e$ (obr. 1). (Príchody a odchody označujeme slabými šípkami, zatiaľčo orientácia hrán je vyznačená plnými šípkami.) Nech $a^0$, $a^-$, $a^+$, resp. $a^{+-}$ označuje počet tých šípov prichádzajúcich do $x$, ktorými sme ešte vôbec nešli, prišli do $x$, odišli z $x$, resp. prišli aj odišli. Analogický význam majú symboly $b^0$, $b^-$, $b^+$ a $b^{+-}$ pre šípy odchádzajúce z $x$. Z našich predpokladov máme: $a^- + b^- = a^+ + b^+$ (počet príchodov sa rovná počtu odchodov), $a^+ = 0$, $b^- = 0$ (pravidlo (TD)), $b^0 = 0$, $a^- = 0$ (pravidlo (T2)). Takže máme: $b^+ = 0$. Teda všetky hrany pri $x$ sú prejdené oboma smermi, alebo sú orientované do $x$ a neprejdené ani raz. Z toho vyplýva, že ak sme sa ešte niekedy vrátili do $x$, tak sme mohli z $x$ vyjsť ihneď tou a len tou istou hranou, a preto v $x$ nemôžeme zastať. Tým sme ukázali, že $x = s$. Teda stojíme v $x$.

Nech $s = u_0, u_1, \ldots, u_q$ je postupnosť vrcholov v tom poradí, v akom sa v našom poloslede vyskytli prvýkrát. Ukážeme, že všetky hrany odchádzajúce z $u_r$ sú prejdené oboma smermi a každá hrana prichádzajúca do $u_r$ je alebo vôbec neprejdená alebo prejdená oboma smermi ($r = 0, 1, \ldots, q$). Pre $r = 0$ sa to hneď ukáže (porovnaj obr. 1) a ďalej použijeme indukciu. Nech teda $r > 0$ a nech hrana $e = u_k u_r$ je tá, po ktorej sme prišli do $u_r$ prvýkrát. Podľa indukčného predpokladu niekedy prejdeme hranu $e$ aj v smere od $u_r$. Po tomto prechode budú (porovnaj obr. 1) všetky odchádzajúce hrany z $u_r$ prejdené dvakrát a prichádzajúce dvakrát




Obr. 1. Situácia po odchode hranou $e$.





alebo ani raz. Preto, ak ešte prídeme do $u_r$, tak len po prichádzajúcej hrane, avšak touto hneď aj odídeme (inej možnosti niet). Teda naše tvrdenie platí. Z tohto už ľahko vyplýva zvyšok tvrdenia vety. ■

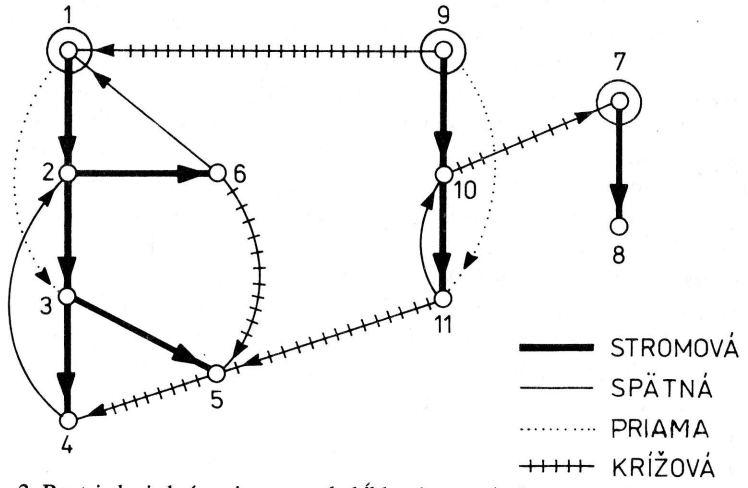
## POSTUP DO HĹBKY

Ak k predchádzajúcemu Tarryho algoritmu pridáme (ako v časti 3.A) nasledujúce pravidlo (T3), získame špeciálnu verziu, ktorej budeme hovoriť *Trémauxov prieskum digrafu* (hoci Trémaux orientované labyrinty neskúmal).

(T3) Ak prídeme hranou prechádzanou prvýkrát do známeho vrcholu, tak hneď v nasledujúcom kroku sa touto hranou vraciame. ●

To zodpovedá postupu do hĺbky a pod týmto názvom ho uvádza Tarjan (1972), ktorý ho explicitne študuje. Žiaľ, aj v tejto verzii je prieskum digrafov dosť zložitý. Oproti grafom, kde nám postup do hĺbky rozdelil hrany na dve triedy (stromové a spätné hrany), tu dostávame až štyri druhy hrán, ako ilustruje obr. 2. Číslo pri každom vrchole $u$ označuje poradie $p(u)$ jeho objavenia. Hrubé čiary predstavujú tzv. *stromové hrany* a vytvárajú zdrojové stromy (zdroje sú 1, 7 a 9). Nestromová hrana $ij$, kde $p(i) < p(j)$ je tzv. *priama hrana* (bodkovane vyznačená; pre otázky existencie sledov môže byť aj vynechaná). Nech $p(i) > p(j)$, ak existuje $j$—$i$ cesta v strome, tak $ij$ sa nazýva *spätná hrana* (tenká čiara na obr. 2); v opačnom prípade sa $ij$ nazýva *krížová hrana* (poprečiarkovaná čiara).

Na rozlíšenie spätnej a krížovej hrany $ij$ stačí vedieť, či je už vrchol $j$ úplne




Obr. 2. Roztriedenie hrán pri postupe do hĺbky: korene (začiatky polosledov) sú 1, 7 a 9.

preskúmaný v tom zmysle, že sme ho už opustili proti šípke stromovou hranou $kj$, kde $f(j) = k$. Vtedy $ij$ je krížová hrana. Inak je zrejme $j$ nejaký iterovaný otec vrcholu $i$, a preto $ij$ je spätná hrana. Po zastavení v koreni $s$ zvolíme za nový koreň ľubovoľný ešte neobjavený vrchol a proces opakujeme, pokiaľ nepreskúmame všetky vrcholy (a teda aj hrany).

Podobne ako pri grafoch aj tento postup možno uskutočniť v čase $O(m+n)$, ak je digraf zadaný odchádzajúcimi okoliami.

## HĽADANIE SILNÝCH KOMPONENTOV

Nájsť silné komponenty nie je ťažká úloha. Priamo z definície možno navrhnúť polynomiálny algoritmus. Keďže táto úloha má značný význam, tak sa hľadali čo najefektívnejšie postupy. (Napr. Liebl a Sedláček (1966) v súvislosti s úpravou matíc na blokovo-trojuholníkový tvar (pozri 3.D).) Avšak až v r. 1970 predložil Faradžev $O(m+n \log n)$ algoritmus založený na postupnom kontrahovaní cyklov a v r. 1972 Tarjan $O(m+n)$ algoritmus založený na postupe do hĺbky. Tu tiež uvádzame lineárny algoritmus využívajúci postup do hĺbky, keďže tento sa ľahšie realizuje ako všeobecný Tarryho prieskum. Algoritmus je založený na dvoch hlavných myšlienkach: (1) Tranzitivita silnej súvislosti a fakt, že cyklus je silne súvislý. (2) Ak prídeme do silného komponentu, ktorý v kondenzácii predstavuje ústie, tak z neho vyjdeme až po úplnom preskúmaní (toto platí aj pri Tarryho prieskume).

Budeme predpokladať, že digraf je zadaný zoznamami odchádzajúcich okolí, t. j. pre každý vrchol $i$ je daná množina $V^+(i)$ v zásobníku. Vrchol $f(i)$ je ten (otec), z ktorého sme prvýkrát prišli do $i$. Ďalej $p(i)$ značí poradie objavenia vrcholu $i$. V zásobníku $ZV$ zaznamenávame postupne objavované vrcholy a v zásobníku $ZE$ zase hrany. Pre každý vrchol $i \in ZV$ zaznamenávame číslo $r(i) = \min\{p(j) \mid j \in ZV$ a už sme zistili, že existuje $i$—$j$ cesta$\}$. Presnejšie, pri objavení vrcholu $i$ položíme $r(i) := p(i)$ a potom $r(i)$ zlepšujeme pri dvoch druhoch príležitostí: (1) Ak objavíme spätnú alebo krížovú hranu $ij$, kde $j \in ZV$, tak $r(i) := r(j)$. (2) Ak sa vraciame stromovou hranou $ij$ (proti šípke, t. j. $f(j) = i$, kde $r(j) < r(i)$, tak $r(i) := r(j)$.

Základný postup je podobný ako pri grafoch. Začíname z nejakého vrcholu $s$ (prvý koreň). Všeobecne, ak sme vo vrchole $i$, tak vo $V^+(i)$ nájdeme posledný neoznačený element $j$, ktorý označíme „stromová“, „priama“, „spätná“ alebo „krížová“ (myslí sa hrana $ij$) podľa uvedených definícií. Ak $j$ bol doteraz neobjavený, tak postúpime na tento vrchol atď. Pritom robíme všetky uvedené záznamy.

Ak prídeme do vrcholu $y$, kde už sú všetky prvky vo $V^+(y)$ označené, tak (ako vyplynie z ďalšieho) $y$ je posledný prvok v zásobníku $ZV$. Potom $y$ vynecháme zo





$ZV$ a prejdeme na vrchol $x = f(y)$ (ak $y$ nie je koreň). Ak pritom $r(y) = p(y)$, tak to znamená, že žiaden z vrcholov $i$, kde $p(i) < p(y)$, nie je dosiahnuteľný z $y$, a vtedy robíme výstup silne súvislého komponentu takto: Zo zásobníka $ZE$ postupne prenášame hrany na výstup, kým nezostane v $ZE$ hrana $xy$ ako posledná, resp. ak $y$ je koreň, tak $ZE$ celkom vyprázdnime. Potom vrchol $y$ spolu s množinou začiatkov prenesených hrán indukuje jeden silný komponent. Hrana $xy$ spája dva rôzne silné komponenty. Z vrcholu $x$ pokračujeme v prieskume atď.

Ak sme už prieskum z jedného koreňa $s$ ukončili, tak zvolíme ľubovoľne ďalší neobjavený vrchol za koreň a pokračujeme v postupe do hĺbky. ●

Správnosť takéhoto algoritmu vyplýva z uvedeného Tarryho prieskumu a z faktu, že spomínaný vrchol $y$ je vrcholom, ktorý predchádza v prieskume všetkým vrcholom silného komponentu prenášaného na výstup a ktorý zodpovedá v danej situácii ústiu kondenzácie. Detaily aj dôkaz lineárnosti prenechávame čitateľovi ako cvičenie. Tak isto nechávame na čitateľa navrhnúť lineárny algoritmus na zostrojenie kondenzácie digrafu.

## MONOTÓNNE OČÍSLOVANIE VRCHOLOV ACYKLICKÉHO DIGRAFU

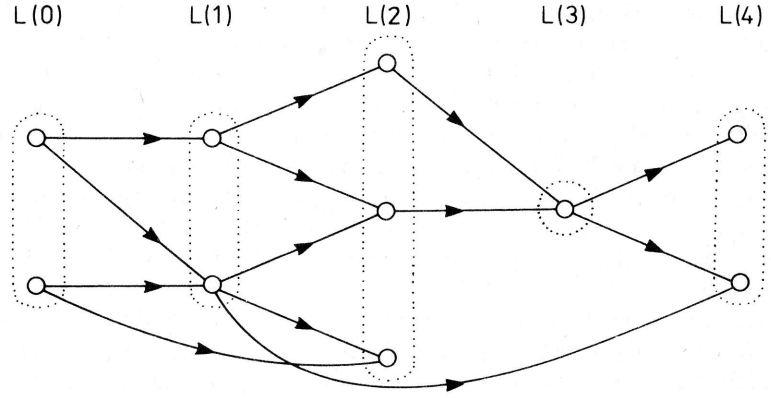
Ako už vieme (veta 1.E.2), vrcholy acyklického digrafu možno očíslovať číslami 1, 2, ..., $n$ tak, že každý šíp smeruje od menšieho čísla k väčšiemu. Takéto *očíslovanie* sa nazýva *monotónne*. (Niekedy sa uvádza pod názvom „topologické usporiadanie“). Tu ukážeme použitie postupu do hĺbky pre získanie takéhoto usporiadania (očíslovania).

Nech premenná $c$ je „číslovač“. Na začiatku položíme $c := n$. Zvolíme ľubovoľný vrchol za koreň a robíme postup do hĺbky. Ak sa dostaneme do nejakého vrcholu $y$, pre ktorý sú už všetky prvky vo $V^+(y)$ očíslované, tak vrcholu $y$ priradíme číslo $c$, potom položíme $c := c-1$ a pokračujeme v postupe. ●

Správnosť tohto algoritmu vyplýva zo skutočnosti, že v prieskume acyklického digrafu nebudú spätné hrany. Priame hrany zrejme vyhovujú získanému očíslovaniu. Pre stromové a krížové hrany to možno hneď ukázať indukciou. Keďže ide o prieskum do hĺbky doplnený o jednoduchú manipuláciu s číslom $c$, vidíme, že ide o $O(m+n)$ algoritmus.

## ROZTRIEDENIE VRCHOLOV ACYKLICKÉHO DIGRAFU DO ÚROVNÍ

Obmenou predchádzajúceho usporiadania je nasledujúci rozklad vrcholovej množiny acyklického digrafu $G$ do úrovní $L(0)$, $L(1)$, ..., $L(k)$ tak, že $L(0) = \{u \in V(G) \mid \deg_{\bar{G}}(u) = 0\}$ a pre $i \geq 1$ nech $G_i := G - (L(0) \cup \ldots \cup L(i-1))$




Obr. 3. Roztriedenie vrcholov acyklického digrafu do úrovní.

a $L(i) := \{u \in G_i \mid \deg^-_{G_i}(u) = 0\}$. Obr. 3 ukazuje príklad digrafu s takto roztriedenými vrcholmi (k = 4).

Nasledujúca metóda založená na definícii úrovní [Goralčíková a Koubek (1979)] je skôr postupom do šírky. Nech $h$ určuje tzv. *výšku vrcholov*, t. j. $h(v) = i$, ak $v \in L(i)$.

Nech je digraf $G$ zadaný zoznamami predchodcov $V^-(u)$ a nasledovníkov $V^+(u)$. Na začiatku určíme čísla $d^-(u) := \deg^-(u)$ (ktoré potom budeme redukovať), a teda určíme aj množinu $L(0)$.

**Algoritmus**

KROK 0: Určíme všetky $d^-(u)$, množinu $L(0)$, $L(i) := \emptyset$ $(i \geq 1)$, $i := 0$ a ideme na krok 1.

KROK 1: Pre každé $u \in L(i)$ urobíme:
[Pre každé $v \in V^+(u)$ urobíme:
[$d^-(v) := d^-(v) - 1$. Ak $d^-(v) = 0$, tak
[$L(i+1) := L(i+1) \cup \{v\}$ a $h(v) := i + 1$.

KROK 2: $i := i + 1$. Ak $L(i) \neq \emptyset$, tak ideme na krok 1: inak STOP ($k = i - 1$, jednotlivé triedy sú $L(0), \ldots, L(i-1)$ a výška každého vrcholu $v$ je $h(v)$). ●

Správnosť algoritmu je zrejmá. Prenechávame čitateľovi overiť, že má lineárnu zložitosť (3.Z.8). Nakoniec poznamenajme, že pomocou úrovní možno ľahko urobiť monotónne očíslovanie vrcholov.

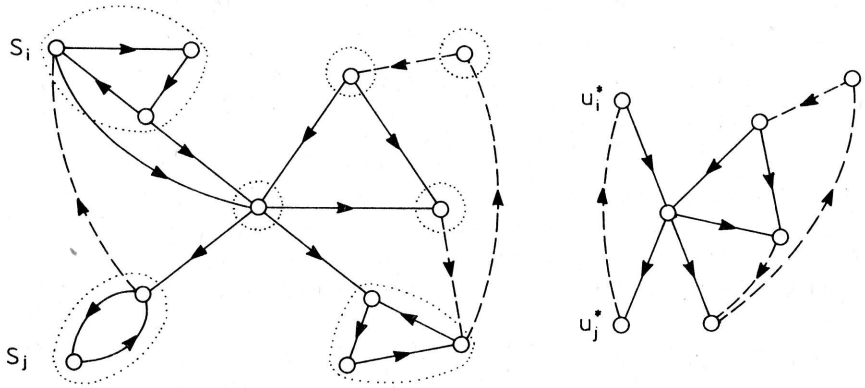




## 3.F
## Najmenšie doplnenie na silne súvislý digraf

Ak je daná komunikačná sieť reprezentovateľná digrafom $G$, ktorý nie je silne súvislý, tak má zmysel doplniť $G$ novými hranami tak, aby sme získali silne súvislý digraf $\hat{G}$. Prirodzená je požiadavka, aby sme pridali ku $G$ čo najmenej hrán (komunikačných liniek). V takom prípade $\hat{G}$ nazývame *najmenší silne súvislý digraf obsahujúci G*. Tento problém postavili a vyriešili Eswaran a Tarjan (1976). Nasledujúci výklad je založený na ich práci.

Najprv k danému digrafu $G$ zostrojíme kondenzáciu $G^*$. Potom utvoríme najmenší silne súvislý digraf $\hat{G}^*$ obsahujúci acyklický digraf $G^*$. Konečne, pomocou $\hat{G}^*$ zostrojíme $\hat{G}$ tak, že pre každú pridanú hranu $u_i^* u_j^*$ v $\hat{G}^*$, kde vrcholy $u_i^*$ a $u_j^*$ zodpovedajú silným komponentom $S_i$ a $S_j$ digrafu $G$, zvolíme ľubovoľne vrcholy $x \in S_i$, $y \in S_j$ a ku $G$ pridáme hranu $xy$ (pozri príklad na obr. 1). ●




Obr. 1. Doplnenie na silne súvislý digraf pomocou kondenzácie.

Je zrejmé, že takýto postup je správny. Okrem toho, kondenzáciu $G^*$ vieme zostrojiť v čase $O(m+n)$ a zostrojenie $\hat{G}$ pomocou $\hat{G}^*$ tiež nevyžaduje viac operácií. Takto zostáva vyriešiť problém pre acyklické digrafy.

**Veta 1.** Nech $G$ je acyklický digraf, ktorý má $s$ prameňov, $t$ ústí a $q$ nulových vrcholov. Potom minimálny počet hrán, ktoré treba pridať ku $G$, aby sme získali silne súvislý digraf, je rovný $\max\{s, t\} + q$.

DÔKAZ. Ak máme vyrobiť silne súvislý digraf, tak pre každý prameň a nulový vrchol musíme pridať aspoň jednu prichádzajúcu hranu, takže spolu aspoň $s+q$ hrán. Analogicky pre každé ústie a nulový vrchol treba pridať aspoň jednu odchádzajúcu hranu, teda spolu aspoň $t+q$ hrán. Porovnaním vidíme, že celkove treba pridať aspoň $\max\{s, t\}+q$ hrán.

Ešte ukážeme, že toľko hrán aj stačí pridať. Budeme predpokladať, že $s \leq t$ (inak to zmenou orientácie zaručíme a vo výsledku potom opäť zmeníme orientáciu). Nech existuje prirodzené číslo $p \leq s$, pramene $v(1), \ldots, v(p)$ a ústia $w(1), \ldots, w(p)$ spĺňajúce tri podmienky:

(i) Existuje $v(i)$—$w(i)$ cesta pre všetky $i=1, \ldots, p$.

(ii) Pre každý prameň $v$ existuje $v$—$w(i)$ cesta pre nejaké $i$.

(iii) Pre každé ústie $w$ existuje $v(i)$—$w$ cesta pre nejaké $i$.

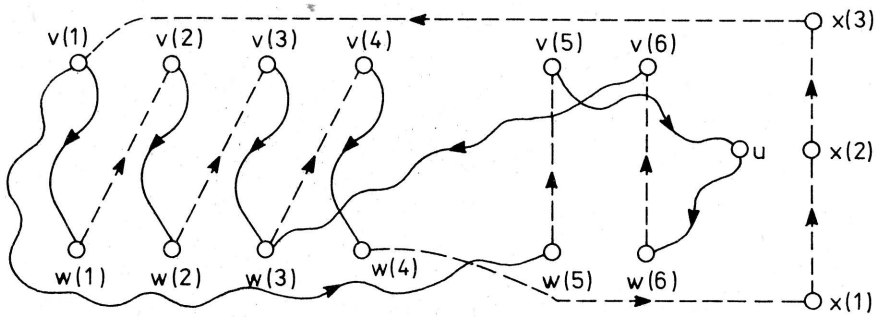
Ľahko vidieť, že také $p$ existuje; neskoršie dáme vhodný algoritmus. Nech ostatné pramene sú $v(p+1), \ldots, v(s)$ a $w(p+1), \ldots, w(s)$ nech sú ďalšie ústia. Nulové vrcholy a ostatné ústia označíme ako $x(1), x(2), \ldots, x(q+t-s)$.

K digrafu $G$ pridáme nasledujúcu množinu hrán:

$$E' = \{w(i)v(i+1) \mid 1 \leq i < p\} \cup \{w(i)v(i) \mid p+1 \leq i \leq s\} \cup$$

$$\cup \begin{cases} \{w(p)v(1)\}, & \text{ak } q+t-s=0, \\ \{w(p)x(1), x(q+t-s)v(1)\} \cup \{x(i)x(i+1) \mid 1 \leq i < q+t-s\}, \\ \text{ak } q+t-s>0. \end{cases}$$

Vidíme, že $|E'| = t+q$. Ľahko sa dokáže, že $G+E'$ je silne súvislý digraf. (Pozri obr. 2, kde máme $p=4$, $s=6$, $t=7$, $q=2$.) Totiž vrcholy $v(1), \ldots, v(p)$, $w(1), \ldots, w(p)$, $x(1), \ldots, x(q+t-s)$ budú podľa (i) ležať v nejakom uzavretom slede $Q$. Pre každý vrchol $u$ mimo $Q$ existuje $u$—$w$ cesta, kde $w$ je nejaké ústie a podľa (ii) potom aj $u$—$w(j)$ cesta, kde $1 \leq j \leq p$, teda existuje $u$—$V(Q)$ cesta. Analogicky




Obr. 2. Ilustrácia k dôkazu vety 1.





pomocou (iii) vidíme, že existuje $V(Q)$—$u$ cesta. Teda $G' = G+E'$ je silne súvislý digraf. ■

Ešte naznačíme, ako nájsť pramene a ústia spĺňajúce podmienky (i)—(iii) v lineárnom čase. To potom dá aj celkove lineárny algoritmus pre najmenšie doplnenie na silne súvislý digraf, lebo ostatné kroky v dôkaze možno uskutočniť tiež v lineárnom čase.

Najprv zvolíme prameň $v_1$ a vyrobíme maximálnu (nepredĺžiteľnú) $v_1$—$x_1$ cestu. Zrejme $x_1$ musí byť ústím, a teda $v(1)=v_1$ a $w(1)=x_1$. Pritom všetkým vrcholom tejto cesty dáme značku $+1$. Potom zvolíme neoznačkovaný prameň $v_2$ a nagenerujeme maximálnu cestu $v_2$—$x_2$ za predpokladu, že nanajvýš vrchol $x_2$ môže byť označkovaný značkou $+1$. Pritom každý neoznačkovaný vrchol tejto cesty dostane značku $+2$ atď. Potom analogicky generujeme „spätné“ cesty z neoznačkovaných ústí $w_1$, $w_2$, ... tak, že najprv hľadáme maximálnu $y_1$—$w_1$ cestu, kde nanajvýš vrchol $y_1$ môže byť označkovaný značkami predchádzajúcich ciest: pritom každý neoznačkovaný vrchol tejto cesty dostane značku $-1$ atď. Takto „pokryjeme“ aj všetky ústia. Ak niektorá z týchto spätných ciest $y_j$—$w_j$ bola taká, že $y_j$ malo značku $+i$ a žiadne ústie ju ešte nemá, tak zmeníme značku $-j$ pri $w_j$ na $+i$. Potom dvojica „(prameň, ústie)“ s tou istou značkou $+i$ tvorí jednu požadovanú dvojicu $(v(k), w(k))$. ●

Zrejme celý proces možno realizovať v čase $O(m+n)$.

Na záver poznamenajme, že problém s ohodnotenými hranami, kde hľadáme doplnenie hranami s minimálnym súčtom ohodnotení, je NP-ťažký (3.Z.13).

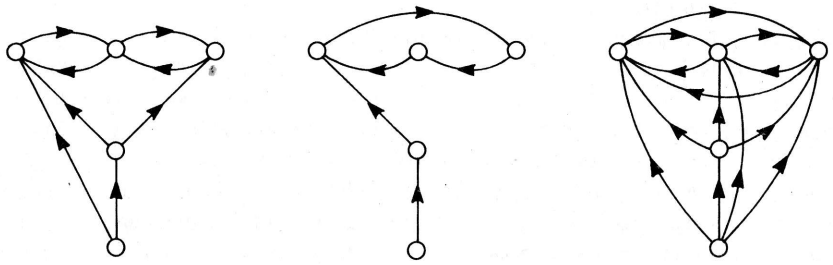
## 3.G
## Tranzitívny uzáver a redukcia

V modeloch rôznych systémov sa niekedy vyskytuje úloha zistiť pre daný digraf $G$ všetky také dvojice $(x, y)$, $x \neq y$, že v $G$ existuje $x$—$y$ sled. Digraf $G^T = (V_G, E^T)$, kde $E^T$ pozostáva zo všetkých takých dvojíc $(x, y)$ sa nazýva tranzitívny *uzáver* digrafu $G$. (Niekedy sa uvažuje aj prípad $x=y$ a skúmajú sa len netriviálne $x$—$y$ sledy. Toto možno zobraziť pseudodigrafom, ktorý sa nazýva reflexívno-tranzitívny uzáver a líši sa od $G^T$ iba tým, že pri niektorých vrcholoch má slučky. Slučka pri vrchole $x$ však zodpovedá tomu, že $x$ leží v netriviálnom silnom komponente, čo spolu pre všetky vrcholy overíme v lineárnom čase. Preto sa stačí zaoberať len s $G^T$.)

Tranzitívny uzáver možno definovať aj ako digraf s maximálnym počtom hrán a s rovnakou dosiahnuteľnosťou, ako má $G$. Digraf s minimálnym počtom hrán

a s rovnakou dosiahnuteľnosťou ako má $G$, nazývame *tranzitívna redukcia* $G^R$. Obr. 1 ilustruje tieto pojmy.

Tranzitívny uzáver digrafu je zrejme jednoznačne určený. Tak isto hneď vidieť, že stačí hľadať tranzitívny uzáver acyklických digrafov, lebo kondenzácia digrafu je acyklická a tranzitívny uzáver každého silného komponentu je kompletný digraf.



Obr. 1. Digraf, tranzitívna redukcia a uzáver.

Pre tranzitívnu redukciu zasa vidíme, že tranzitívna redukcia každého silného komponentu je cyklus. Takto potom stačí vedieť hľadať tranzitívnu redukciu kondenzácie, teda acyklického digrafu. Ak z digrafu $G$ vynecháme postupne tranzitívne hrany (hrana $xy$ sa nazýva *tranzitívna*, ak existuje $x$—$y$ sled neobsahujúci hranu $xy$), tak získame faktor bez tranzitívnych hrán a s rovnakou dosiahnuteľnosťou, ako má $G$. Preto stačí vedieť hľadať tranzitívnu redukciu pre acyklické digrafy bez tranzitívnych hrán. Ľahko možno dokázať, že každý acyklický digraf $G$ bez tranzitívnych hrán má tranzitívnu redukciu rovnú $G$ [Aho, Garey a Ullman (1972)]. Z toho hneď vidíme, že tranzitívna redukcia acyklického digrafu je jediná.

Takto zrejme existujú polynomiálne algoritmy pre zostrojenie tranzitívneho uzáveru aj redukcie.

## WARSHALLOV ALGORITMUS PRE TRANZITÍVNY UZÁVER

Nasledujúci $O(n^3)$ algoritmus je od Warshalla (1962). Algoritmus tejto zložitosti dáva aj Martynjuk (1962).

Nech je digraf $G$ s $V_G = \{1, 2, \ldots, n\}$ zadaný susedskou maticou $A$ (boolovská matica): $a_{ij} = 1$ ak $ij \in E_G$ a $a_{ij} = 0$ inak. Postupne počítame boolovské matice $A^{(0)} := A$, $A^{(1)}$, $A^{(2)}$, ..., $A^{(n)}$. Pritom pre každé $k \in \{0, 1, 2, \ldots, n\}$ žiadame, aby prvok $a_{ij}^{(k)}$ matice $A^{(k)}$ bol rovný 1 práve vtedy, keď v $G$ existuje $i$—$j$ sled, ktorého každý vnútorný vrchol patrí do množiny $\{1, 2, \ldots, k\}$. Pre $k = 0$ je to zrejme





splnené. Pre $k \geq 1$ vidíme, že každý takýto sled alebo neobsahuje vrchol $k$, alebo je zreťazením nejakého $i$—$k$ sledu a $k$—$j$ sledu, pričom oba majú všetky vnútorné vrcholy v množine $\{1, 2, \ldots, k-1\}$. Takto máme:

$$\text{(W1)} \qquad a_{ij}^{(k)} = a_{ij}^{(k-1)} \vee (a_{ik}^{(k-1)} \wedge a_{kj}^{(k-1)}).$$

Z uvedeného je zrejmé, že $A^{(n)}$ bude susedskou maticou reflexívno-tranzitívneho uzáveru a vynechaním slučiek získame tranzitívny uzáver. Keďže prechod od $A^{(k-1)}$ k $A^{(k)}$ vyžaduje $O(n^2)$ operácií, tak na celý algoritmus stačí $O(n^3)$ operácií.

Ešte si všimnime, že vystačíme s jedným poľom, v ktorom sa na začiatku nachádza matica $A^{(0)}$. Skutočne, z (W1) máme:

Ak $a_{ik}^{(k-1)} = 0$, tak $a_{ij}^{(k)} = a_{ij}^{(k-1)} = a_{ij}^{(k-1)}$ a ak $a_{ik}^{(k-1)} = 1$, tak $a_{ij}^{(k)} = a_{ij}^{(k-1)} \vee a_{kj}^{(k-1)}$. Keďže vždy je $a_{kj}^{(k)} = a_{kj}^{(k-1)}$, tak pre $i$-tý riadok $r_i^{(k)}$ matice $A^{(k)}$ platí: $r_k^{(k)} = r_k^{(k-1)}$ a všeobecne:

$$\text{(W2)} \qquad r_i^{(k)} = \begin{cases} r_i^{(k-1)}, & \text{ak } a_{ik}^{(k-1)} = 0, \\ r_i^{(k-1)} \vee r_k^{(k-1)}, & \text{ak } a_{ik}^{(k-1)} = 1. \end{cases}$$

Tento vzťah umožňuje získať z matice $A^{(k-1)}$ maticu $A^{(k)}$ na tom istom poli. ●

Konečne poznamenajme, že efektívnosť tohto algoritmu je v tom, že boolovská disjunkcia sa veľmi ľahko na počítači realizuje, keďže pri malom $n$ celý riadok $r_i$ predstavuje jedno, resp. niekoľko „slov" (takže namiesto $n$ operácií možno hovoriť o jednej).

## TRANZITÍVNY UZÁVER CEZ KONDENZÁCIU

Keďže existuje $O(m+n)$ algoritmus na zostrojenie kondenzácie digrafu, ponúka sa myšlienka urobiť najprv tranzitívny uzáver kondenzácie a z neho potom tranzitívny uzáver pôvodného digrafu. Pretože, ako vieme, kondenzácia je vždy acyklický digraf, tak je tu nádej, že tranzitívny uzáver sa bude robiť jednoduchšie než vo všeobecnosti. Takýto prístup volia napr. Faradžev (1970) a Munro (1971).

Prenechávame čitateľovi dokázať, že ak poznáme tranzitívny uzáver $G^{*T}$ kondenzácie $G^*$ digrafu $G$, tak v čase $O(n^2)$ ľahko zostrojíme tranzitívny uzáver $G^T$ digrafu $G$ (3.Z.15). Potom celý algoritmus závisí od toho, aký dobrý máme algoritmus pre zostrojenie tranzitívneho uzáveru acyklického digrafu. Faradžev (1970) dáva $O(mn + n^2)$ algoritmus. Tu uvádzame takýto algoritmus založený na postupe do hĺbky.

Nech acyklický digraf $G$ je zadaný zoznamami nasledovníkov. Ďalej budeme používať maticu typu $(n, n)$, ktorá sa na začiatku rovná matici susednosti $A(G)$ digrafu $G$ a na konci algoritmu matici susednosti $A(G^T)$ jeho tranzitívneho

uzáveru. Vždy keď sa v prieskume vraciame proti šípke hrany (každého druhu) $ij$, tak k nasledovníkom vrcholu $i$ pridáme všetkých nasledovníkov vrcholu $j$. To možno realizovať tak, že boolovský súčet riadkov $i$ a $j$ našej matice dáme do riadku $i$. Takto pre každú hranu vystačíme s $O(n)$ operáciami, a teda celkove s $O(mn)$ operáciami. Na prvú maticu $A(G)$ vystačíme s $O(n^2)$ operáciami, a preto na celý algoritmus s $O(mn+n^2)$ operáciami. To je nanajvýš $O(n^3)$, ale pri riedkych digrafoch to môže byť podstatne menej. ●

## HĽADANIE TRANZITÍVNEJ REDUKCIE A UZÁVERU ACYKLICKÉHO DIGRAFU SÚČASNE

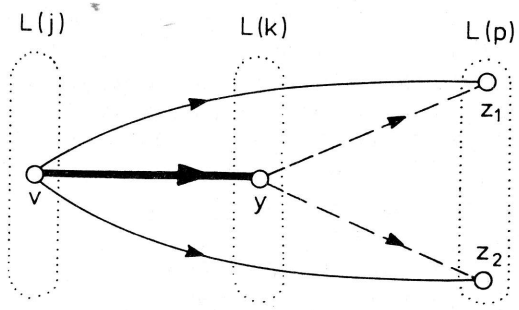
Podobne ako pre tranzitívny uzáver aj pre redukciu platí, že z tranzitívnej redukcie kondenzácie vieme v čase $O(n^2)$ vyrobiť tranzitívnu redukciu pôvodného digrafu (3.Z.15). Preto aj tu sa pozornosť sústreďuje na acyklické digrafy. Algoritmus z práce [Goralčíková a Koubek (1979)], ktorý teraz uvedieme, súčasne počíta pre acyklický digraf $G$ tranzitívnu redukciu $G^R$ aj tranzitívny uzáver $G^T$, a to v čase $O(nm_R+m)$, kde $m_R$ je počet hrán v $G^R$. V ďalšom kladieme $T:=G^T$ a $R:=G^R$.

Predpokladajme, že digraf $G$ má vrcholy rozložené do úrovní $L(0)$, $L(1)$, ..., $L(s)$ a pre každý vrchol $v$ je daná jeho výška $h(v)$ (číslo úrovne). To možno urobiť v lineárnom čase (pozri 3.E). Pripomeňme si, že $R$ vznikne z $G$ vynechaním tranzitívnych hrán. Ihneď z definície úrovňových množín máme:

$$(GK1)\ L(h(v)+1)\cap V_R^+(v)=L(h(v)+1)\cap V_T^+(v)=L(h(v)+1)\cap V_G^+(v).$$

Ak označíme:

$$A(v,p):=\bigcup_{k=h(v)+1}^{p-1}\ \bigcup_{y\in L(h)\cap V_R^+(v)} L(p)\cap V_T^+(y),$$




Obr. 2. Ilustrácia vzťahov (GK2).





tak pre každé $p>h(v)+1$ môžeme písať:

$$\text{(GK2)}\qquad \begin{aligned} L(p)\cap V_R^+(v)&=[L(p)\cap V_G^+(v)]-A(v,p),\\ L(p)\cap V_T^+(v)&=[L(p)\cap V_R^+(v)]\cup A(v,p).\end{aligned}$$

K dôkazu týchto rekurentných vzťahov si stačí uvedomiť definíciu $T$ a $R$ (pozri aj obr. 2, kde $vy\in R$, $vz_1\in G$ a ostatné hrany sú z $T-E_G$: tu $z_1\notin L(p)\cap V_R^+(v)$ a $z_2\in L(p)\cap V_T^+(v)$).

Zo vzťahov (GK1) a (GK2) môžeme postupne určiť množiny $V_R^+(v)$ a $V_T^+(v)$ pre každý vrchol $v$ tým, že určíme jednotlivé prieniky s úrovňovými množinami.

### Algoritmus

Pre $i=1, 2, \ldots, s$ urobíme:
  Pre $j=0, 1, 2, \ldots, s-i$ urobíme:
    $p:=j+i$ a pre každé $v\in L(j)$ urobíme:

$$A:=\bigcup_{k=j+1}^{p-1}\ \bigcup_{y\in L(k)\cap V_R^+(v)} L(p)\cap V_T^+(y),$$

$$L(p)\cap V_R^+(v):=[L(p)\cap V_G^+(v)]-A,$$

$$L(p)\cap V_T^+(v):=[L(p)\cap V_R^+(v)]\cup A.\ \bullet$$

Poznamenajme, že $i$ tu určuje rozdiel $p-j$, ktorý je najprv 1, potom 2 atď., čo umožňuje stavať na predchádzajúcich výpočtoch.

Odhadneme zložitosť uvedeného algoritmu za predpokladu, že množiny $L(i)\cap V_G^+(v)$ a $L(i)\cap V_R^+(v)$ sú ukladané ako zásobníky, zatiaľ čo každá z množín $L(i)\cap V_T^+(v)$, $A(v,i)$ je vyjadrená charakteristickou funkciou na množine $L(i)$ (táto funkcia má hodnotu 1, ak $v\in L(i)$ patrí do množiny a 0 inak).

Pre najvnútornejší cyklus algoritmu stačia nasledujúce počty operácií (časy). Na zostrojenie množiny A:

$$\sum_{k=j+1}^{p-1} |L(k)\cap V_R^+(v)|\,|L(p)|.$$

Množinu $L(p)\cap V_G^+(v)$ zostrojíme pomocou výškovej funkcie $h$ v čase $O(|V_G^+(v)|)$. Potom množinu $L(p)\cap V_R^+(v)$ zostrojíme v čase $O(|L(p)\cap V_G^+(v)|)$ a množinu $L(p)\cap V_T^+(v)$ v čase $O(|L(p)|\cap V_R^+(v)|)$.

Množina $A$ je vlastne $A(v,p)$ a počas celého algoritmu sa každé $A(v,p)$ počíta len raz. Preto celkový čas na všetky množiny A nepresiahne:

$$\sum_{v\in VG}\sum_{p=0}^{s} O\Big(|L(p)|\sum_{k=j+1}^{p-1}|L(k)\cap V_R^+(v)|\Big)\leqslant$$

$$\leqslant O\Big[\sum_{v\in V_G}\Big(\sum_{p=0}^{s}|L(p)|\Big)|V_R^+(v)|\Big]\leqslant O\Big[\sum_{v\in V_G}|V_G|\,|V_R^+(v)|\Big]=O(nm_R).$$

Analogicky vidieť, že celkový čas na konštrukciu množín $L(p)\cap V_G^+(v)$ nepresiahne

$$\sum_{v\in V_G}O(|V_G^+(v)|)=O(m).$$

Na konštrukciu množín $L(p)\cap V_R^+(v)$ stačí čas

$$\sum_{v\in V_G}\sum_{p=0}^{s}O(|L(p)\cap V_G^+(v)|)\leqslant\sum_{v\in V_G}O(|V_G^+(v)|)=O(m).$$

Úplne podobne vidieť, že množiny $L(p)\cap V_T^+(v)$ možno skonštruovať v čase $O(m_R)$.

Sumarizujúc vidíme, že na celý algoritmus vystačíme s $O(mn_R+m)$ operáciami. Keďže $m_R$ môže byť dosť nízke, napr. $O(n)$, tak niekedy to dáva $O(n^2)$ algoritmus. Na druhej strane, môžeme dostať len $O(n^3)$ algoritmus. Aho, Garey a Ullman (1972) dokázali, že ak $2\leqslant\alpha\leqslant 3$, tak existuje $O(n^\alpha)$ algoritmus pre tranzitívny uzáver práve vtedy, ak existuje $O(n^\alpha)$ algoritmus pre tranzitívnu redukciu acyklického digrafu.

## TRANZITÍVNY UZÁVER A NÁSOBENIE MATÍC

Teraz uvedieme algoritmus pre tranzitívny uzáver, ktorý má menšiu zložitosť ako $O(n^3)$ a využíva rýchly algoritmus pre násobenie matíc. Vychádzame z práce [Munro (1971)].

Ak $G$ je digraf, tak $G^2$ označuje tzv. druhú mocninu digrafu $G$, t. j. $V(G^2)=V_G$ a $E(G^2)=\{xy\,|\,x,y\in V_G, x\neq y$ a vzdialenosť $d_G(x,y)\leqslant 2\}$. (Vo všeobecnosti je $G^{n-1}=G^T$.) Monotónne očíslovaný acyklický digraf $G$ s $2q$ vrcholmi $1, 2, \ldots, 2q$, ktoré sú zároveň ich číslami, nazývame kvázitranzitívny, ak oba indukované podgrafy $G(\{1, 2, \ldots, q\})$ a $G(\{q+1, \ldots, 2q\})$ sú tranzitívne uzavreté.

**Veta 1.** Ak existuje $O(n^\alpha)$ algoritmus pre 2. mocninu kvázitranzitívnych digrafov, kde $2\leqslant\alpha\leqslant 3$, tak existuje aj $O(n^\alpha)$ algoritmus pre tranzitívny uzáver digrafov.

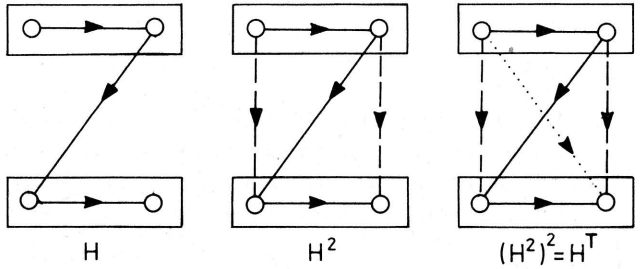
DÔKAZ. Podľa uvedeného môžeme sa ohraničiť na acyklické digrafy. Teda nech $G$ je acyklický digraf a nech má $n=2^t$ vrcholov (to možno dosiahnuť pridaním nulových vrcholov), ktoré sú monotónne očíslované.

Najprv urobíme tranzitívne uzávery $2^{t-1}$ indukovaných podgrafov $G(\{1, 2\})$, $G(\{3, 4\})$, ..., $G(\{n-1, n\})$: pomocou nich potom tranzitívne uzávery digrafov $G(\{1, 2, 3, 4\})$, $G(\{5, 6, 7, 8\})$, ..., ktorých je $2^{t-2}$ atď. až získame tranzitívny





uzáver digrafu $G$. Vidíme, že pri tomto stačí robiť len tranzitívne uzávery kvázitranzitívnych digrafov $H$. Poslednú úlohu však zvládneme, ak urobíme 2. mocninu digrafu $H$, t. j. $H^2$ a potom 2. mocninu výsledku, t. j. $(H^2)^2$ (pozri obr. 3). ●




Obr. 3. Zostrojenie uzáveru kvázitranzitívneho digrafu.

Nech existuje $O((2^i)^\alpha)$ algoritmus na utvorenie 2. mocniny kvázitranzitívneho digrafu $H$ s $2^i$ vrcholmi. Na utvorenie 4. mocniny máme teda tiež $O((2^i)^\alpha)$ algoritmus. Nech $c>0$ je také číslo, že vždy vystačíme s $c(2^i)^\alpha$ operáciami. Keďže máme urobiť 4. mocninu pre $n/2^i$ digrafov, z ktorých každý má $2^i$ vrcholy, tak v tejto iterácii vystačíme s $c(2^i)^\alpha n/2^i$ operáciami. Pretože $i$ prebieha od 2 po $t$ $(n=2^t)$, tak na celý algoritmus vystačíme s počtom operácií:

$$\sum_{i=2}^{t}c(2^i)^\alpha n/2^i=cn\sum_{i=2}^{t}(2^{\alpha-1})^i<cn(2^{\alpha-1})^{t+1}=$$

$$=cn2^{\alpha-1}.(2^t)^{\alpha-1}=cn.2^{\alpha-1}.n^{\alpha-1}=c.2^{\alpha-1}.n^\alpha.$$

Teda $O(n^\alpha)$ operácií nám stačí. ■

Ako vieme z kap. 1, prvky $a_{ij}$ 2. mocniny matice susednosti $A$ digrafu $G$ vyjadrujú počty rôznych $i$—$j$ sledov dĺžky 2 v $G$. Takže nahradiac v $A^2$ všetky nenulové prvky jednotkami a prvky v hlavnej uhlopriečke nulami, získame maticu susednosti digrafu $G^2$. Avšak na vyrobenie $A^2$ nám stačí $O(n^{\log_2 7})$ operácií [Strassen (1969)], takže tým máme zaručený $O(n^{2,81})$ algoritmus pre tranzitívny uzáver acyklického, a tým aj ľubovoľného digrafu. Okrem Munra takýto algoritmus ohlásili aj Fischer a Meyer (1971). Výsledok Strassena bol viackrát zlepšený. Ako uvádza Tarjan (1980), V. Y. Pan našiel dokonca $O(n^{2,49})$ algoritmus pre súčin dvoch matíc typu $(n, n)$, a tým aj $O(n^{2,49})$ algoritmus pre tranzitívny uzáver, resp. redukciu. Tento výsledok má skôr teoretický než praktický význam.

Na záver ešte poznamenajme, že Schnorr (1977) oznamuje pravdepodobnostný algoritmus, pre ktorý dokazuje, že ak $m_T$ je očakávaný počet hrán tranzitívneho uzáveru, tak $O(n+m_T)$ je očakávaný počet operácií pre nájdenie tranzitívneho uzáveru.

## 3.Z
## Cvičenia a poznámky

1. Dokážte, alebo vyvráťte: Pre nájdenie cesty z rovinného labyrintu sa stačí pridržať stále ľavej steny (t. j. na rázcestí vždy volíme prvú chodbu vľavo).
2. Zostrojte lineárny algoritmus pre vybranie $u$—$v$ cesty z daného $u$—$v$ sledu.
3. Navrhnite podrobnejšie lineárne algoritmy pre postupy do hĺbky, resp. šírky v grafoch.
4. (a) Dokážte, že postup načrtnutý v 3.B správne hľadá bloky aj artikulácie. Navrhnite $O(m+n)$ algoritmus pre identifikáciu blokov, artikulácií a tiež pre zostrojenie blokovo-artikulačného grafu.
   (b) Analogicky pre listy, mosty a listovú kondenzáciu.
5. Navrhnite podrobnejší $O(m+n)$ algoritmus pre postup do hĺbky v digrafoch načrtnutý v 3.E.
6. Zostrojte podrobnejší $O(m+n)$ algoritmus pre hľadanie silných komponentov digrafu a tiež pre zostrojenie kondenzácie digrafu.
7. Navrhnite lineárny algoritmus pre zisťovanie jednostrannej súvislosti digrafu.
8. Ukážte, že algoritmus pre roztriedenie vrcholov acyklického digrafu do úrovní (3.E) je lineárny.
9. Dokážte, že ak je digraf slabo súvislý a rovnovážne orientovaný, tak je silne súvislý.
10. (a) Nech $G$ je silne súvislý digraf s $n \geq 2$. Potom pre každý vrchol $v \in V_G$ existuje vrchol $u(v) \neq v$ tak, že $G-u(v)$ je jednostranne súvislý a vrchol $v$ je stokom digrafu $G-u(v)$ [Horák (1983)]. [Návod: Nech $Z$ je cyklus obsahujúci $v$. Ak $V_Z \neq V_G$, tak stiahnite $Z$ na jediný vrchol a použite indukciu.]
    (b) Navrhnite polynomiálny algoritmus pre nájdenie takého vrcholu $u(v)$.
11. (a) Dokážte, že graf má silne súvislú orientáciu práve vtedy, keď je list [Robbins (1939)].
    (b) Dokážte, že silne súvislý migraf $G$ má silne súvislú orientáciu (zorientujeme rebrá) $\Leftrightarrow$ pre každú hranu $e$ je $G-e$ slabo súvislý migraf $\Leftrightarrow$ dezorientácia migrafu $G$ (šípy zmeníme na rebrá) je listom.
12. Dokážte, že pre ľubovoľné rebro $e$ bloku $G$ s $n \geq 3$ vrcholmi existuje taká silne súvislá orientácia $\hat{G}$ grafu $G$, ktorej každý cyklus obsahuje hranu $e$ (t. j. $\hat{G}-e$ je acyklický digraf) [Sedláček (1959)].





13. Dokážte, že problém najlacnejšieho doplnenia digrafu $G$ na silne súvislý digraf je NP-ťažký (hrany mimo $G$ majú ceny).
14. Dokážte, že problém najlacnejšieho doplnenia grafu $G$ na blok je NP-ťažký, dokonca aj vtedy, ak $G$ je strom.
15. Ukážte, ako možno z tranzitívneho uzáveru $G^{*T}$ kondenzácie $G^*$ zostrojiť v čase $O(n_G^2)$ tranzitívny uzáver digrafu $G$. Tiež ukážte analógiu pre redukcie.
16. (a) Dokážte, že nájsť podgraf $H$ grafu $G$ s minimálnym počtom hrán a s rovnakou dosiahnuteľnosťou možno v čase $O(m+n)$.
    (b) Ukážte, že analogický problém v digrafoch je NP-ťažký, dokonca v planárnych digrafoch s maximálnym stupňom 3. [Návod: Uvážte problém hamiltonovského cyklu.]
17. Dokážte, že pravidelný bipartitný graf nemá artikuláciu [Lovász (1979)].
18. Množina vrcholov $S \subset V(G)$ sa nazýva *jadro digrafu* $G$, ak je *nezávislá* (t. j. ak $u, v \in S$, tak $uv \notin E(G)$) a *absorbujúca* (t. j. ak $x \notin S$, tak $S \cap V^+(x) \neq \emptyset$). Jadrá sa študovali v súvislosti s hrami dvoch hráčov. Každej pozícii $P_i$ je priradený vrchol digrafu $v_i$ a šíp $v_iv_j$ označuje, že z pozície $P_i$ možno na jeden ťah prejsť do pozície $P_j$. Ak je hráč v jadre, tak (pri jeho správnej hre) už ho z jadra nemožno vytlačiť.
    (a) Dokážte, že každý digraf bez cyklov nepárnej dĺžky má jadro [Richardson (1946)]. [Návod: Nech $H_0$ je silný komponent zodpovedajúci ústiu v kondenzácii (ak $G$ je silne súvislý, tak $H_0 = G$). Nech $v_0 \in V(H_0)$ a nech $S_0 = \{u \in V(H_0) |$ v $H_0$ existuje $v_0$—$u$ sled párnej dĺžky$\}$. Dokážte, že $S_0$ je jadrom v $H_0$. Ak $H_0 = G$, tak $S_0$ je jadrom v $G$. Inak nech $G_1 := G - S_0 - V_G^-(S_0)$. V $G_1$ nájdeme silný komponent $H_1$ zodpovedajúci ústiu a v ňom (ako hore) jadro $S_1$. Potom nech $G_2 := G_1 - S_1 - V_G^-(S_1)$ atď. Po vyčerpaní všetkých vrcholov v $G$, nech $S = S_0 \cup S_1 \cup S_2 \cup \ldots$. Ukážte, že $S$ je jadrom digrafu $G$. [Berge (1970)].
    (b) Dokážte, že nájsť jadro digrafu (vo všeobecnosti) je NP-ťažký problém [V. Chvátal (pozri [Garey, Johnson (1979)])]. [Návod: Pre problém splniteľnosti zostrojte digraf $G$ s vrcholmi $x_1, \bar{x}_1, \ldots, x_p, \bar{x}_p, C_1, \ldots, C_q$ a šípmi $x_i\bar{x}_i$, $\bar{x}_ix_i$ pre všetky $i$: $C_jx_k$, ak klauzula $C_j$ obsahuje symbol $x_k$ a podobne $C_j\bar{x}_k$. Potom ešte pridáme vrcholy $C_j'$, $C_j''$ a šípy $C_jC_j'$, $C_j'C_j''$ a $C_j''C_j$. Dokážte, že forma je splniteľná práve vtedy, keď $G$ má jadro.]
19. Nech $G$ je graf s $V(G) = \{v_1, \ldots, v_n\}$. Ku každej bijekcii $f$: $V(G) \to \{1, 2, \ldots, n\}$ priraďme číslo $\varepsilon_f(G) := \Sigma\, |f(v_i) - f(v_j)|$ (suma prebieha cez všetky hrany $v_iv_j$). Číslo $\varepsilon(G) = \min_f \varepsilon_f(G)$ sa nazýva *elevácia grafu* $G$ a číslo $\bar{\varepsilon}(G) = \max_f \varepsilon_f(G)$ *koelevácia* grafu $G$.

(a) Dokážte, že elevácia kompletného grafu $K_n$ je $\binom{n+1}{3}$.

(b) Ak $\bar{G}$ je komplement grafu $G$ s $n$ vrcholmi, tak $\varepsilon(G)+\bar{\varepsilon}(\bar{G}) = \varepsilon(K_n)$. Dokážte.

Tieto aj ďalšie pozorovania uvádzajú Niepel a Tomasta (1981) a úloha súvisí s optimálnym rozmiestňovaním objektov na priamke. Avšak Garey, Johnson a Stockmeyer (1976) dokázali, že nájsť eleváciu grafu je NP-ťažké.

20. Nech $G$ je graf s $n$ vrcholmi. Ku každej bijekcii $f$: $V(G)\to\{1, 2, \ldots, n\}$ priraďme číslo $w_f(G) = \max|f(u)-f(v)|$, kde maximalizujeme cez všetky $uv \in E(G)$: $w_f(G) := 0$ ak $m=0$. Tzv. *pásová šírka grafu* $G$ je číslo $w(G)$ $= \min_f w_f(G)$. (Ak vrcholy grafu $G$ usporiadame podľa čísel $f(u)$ a zostrojíme príslušnú susedskú maticu $A$, tak všetky nenulové prvky budú ležať v prvých $w_f(G)$ naddiagonálach a v prvých $w_f(G)$ poddiagonálach matice $A$: číslo $w(G)$ zodpovedá susedskej matici s minimálnou šírkou pásu; bližšie pozri napr. [Fiedler (1981)].)

(a) Ukážte, že $m \leq (n-1)+(n-2)+\ldots+(n-w(G))$.

(b) Navrhnite lineárny algoritmus na zistenie toho, či $w(G) \leq 1$.

(c) Navrhnite nejakú heuristiku pre hľadanie $w(G)$.

V [Garey, Graham et al. (1978)] je lineárny algoritmus na zistenie toho, či $w(G) \leq 2$: tiež je tam ukázané, že nájsť $w(G)$ je NP-ťažký problém dokonca pre stromy s maximálnym stupňom 3: literatúra citovaná v tomto článku obsahuje viaceré algoritmy a heuristiky.



# 4
# OPTIMÁLNE SLEDY

Mnoho reálnych optimalizačných úloh možno previesť na úlohu nájsť optimálny sled (obyčajne najkratšiu cestu). Tieto úlohy treba upresniť, a takto vznikajú viaceré problémy podľa toho, či je to úloha v grafe alebo digrafe, či sú hrany ohodnotené, či treba nájsť sled z jedného vrcholu do druhého alebo medzi každými dvoma vrcholmi, alebo či stačí jeden sled a pod.

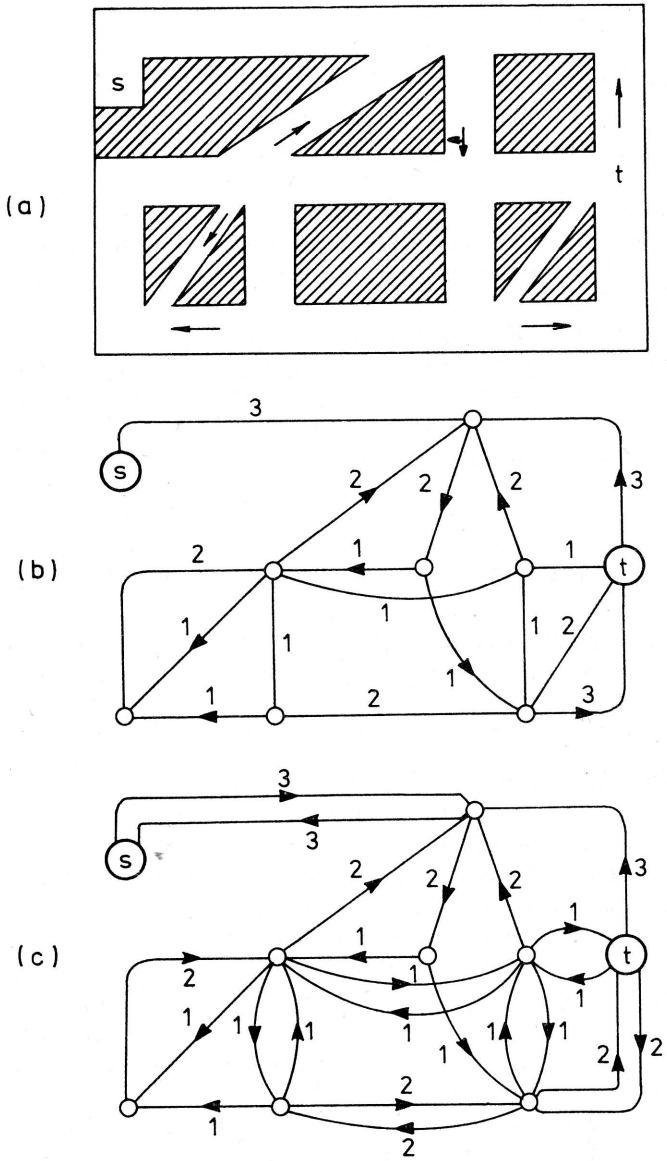
Uvedieme viaceré polynomiálne algoritmy pre rôzne úlohy. Najčastejšie však uvažujeme najkratšie cesty. Pravda, za istých predpokladov na štruktúru, resp. ohodnotenie. Takmer všetky algoritmy sú na digrafoch, keďže tieto problémy z grafov možno často previesť na digrafy.

## 4.A
## Príklady na problém optimálnej cesty

Uvedieme niekoľko úloh, ktoré možno previesť na problém optimálnej cesty. S ďalšími úlohami, kde sa tento problém vyskytuje (ako podúloha), oboznámime sa neskôr.

### NAJKRATŠIA TRASA V CESTNEJ SIETI

Uvažujme cestnú sieť nejakého mesta. Úlohou je nájsť najkratšiu trasu z daného bodu $s$ do daného bodu $t$. Pritom každý úsek $ij$ ulice (medzi dvoma križovatkami) má kladnú dĺžku $c_{ij}$: niektoré úseky sú obojsmerné, niektoré jednosmerné a na niektorých križovatkách sú zakázané niektoré smery jazdy (pozri príklad na obr. 1(a)). Toto možno zobraziť pseudomigrafom (obr. 1(b)) a tento prípadne ešte redukovať na digraf (obr. 1(c)). Úlohou potom je nájsť najkratší $s$—$t$ sled v tomto digrafe. (Čitateľ by si mal uvedomiť, že pri hľadaní najkratšieho sledu možno každé rebro nahradiť dvojicou protišípov tej istej dĺžky a že namiesto množiny násobných šípov stačí uvažovať najkratší z nich a ostatné vynechať). Čitateľ by si mal premyslieť alternatívne riešenie križovatiek s prikázanými smermi. Po transformácii úlohy do digrafu hľadáme *najkratší $s$—$t$ sled*, t. j. taký $s$—$t$ sled, ktorého súčet dĺžok hrán je minimálny. Z predpokladu kladných dĺžok hrán vyplýva, že taký $s$—$t$ sled bude cestou.




Obr. 1. Úloha: (a) v cestnej sieti, (b) v migrafe, (c) v digrafe.





## NAJSPOĽAHLIVEJŠÍ SLED

V danej komunikačnej sieti, kde každá linka $e$ má danú spoľahlivosť $p_e$ (pravdepodobnosť správnej činnosti), treba nájsť *najspoľahlivejší s—t sled*. Spoľahlivosť sledu je rovná súčinu spoľahlivostí jednotlivých liniek.

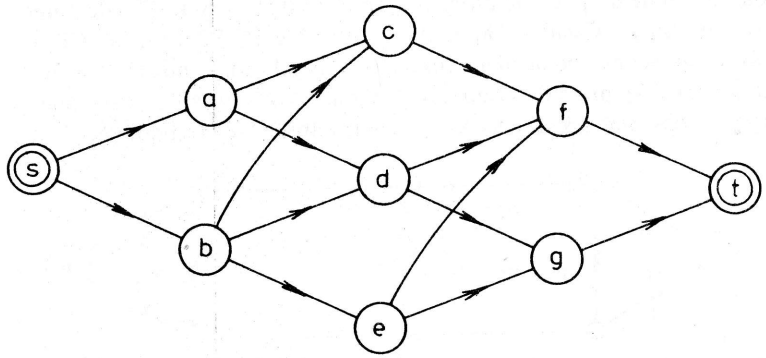
Ak v príslušnom digrafe (alebo grafe) položíme dĺžku každej hrany $e$ rovnú $-\log p_e \geq 0$, dostávame problém najkratšej $s$—$t$ cesty.

## NAJŠIRŠÍ SLED

Ak má každá linka komunikačnej siete predpísanú šírku, tak sa môžeme pýtať na *najširší s—t sled*. Pritom pod šírkou sledu rozumieme šírku najužšej linky v ňom. Takéto úlohy vznikajú napr. pri preprave nadmerných nákladov v cestnej sieti.

## ANALÝZA PROJEKTU

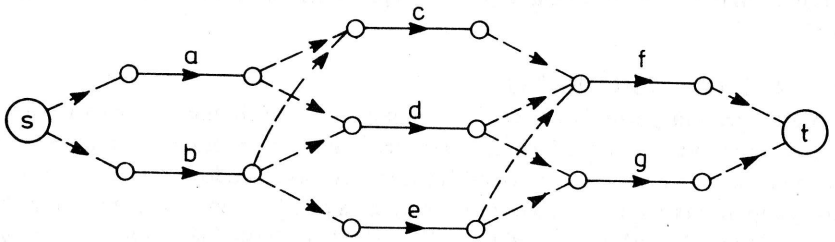
Veľký projekt pozostáva obvykle z množstva drobných činností (úloh). Každá činnosť si vyžaduje istý čas (*trvanie činnosti*). Naviac, niektorá činnosť môže začať až vtedy, keď sa nejaké iné už skončili (napr. pri stavbe budovy staviame múry až po vykopaní základov, vybudovaní kanalizácie a pod.). Teda sú tu „technologické väzby", ktoré čiastočne usporadúvajú činnosti. Toto čiastočné usporiadanie možno zobraziť digrafom: každej činnosti zodpovedá jeden vrchol a šípy udávajú technologické väzby (šíp $uv$ znamená, že činnosť $u$ sa má vykonať prv ako činnosť $v$): okrem toho pridáme dve fiktívne činnosti — vrcholy $s$ a $t$, kde $s$ zodpovedá *začatiu projektu* (začiatok) a $t$ *ukončeniu projektu* (koniec). Každá činnosť nasleduje až po $s$ a predchádza $t$. Získaný digraf musí byť acyklický (inak projekt




Obr. 2. Digraf projektu, kde činnosti sú reprezentované vrcholmi.

nemožno realizovať) a každý jeho vrchol leží na nejakej $s$—$t$ ceste. Avšak takýto digraf môže mať zbytočne mnoho hrán a pre ďalšie účely stačí uvažovať jeho tranzitívnu redukciu. Príklad výsledného digrafu je na obr. 2.

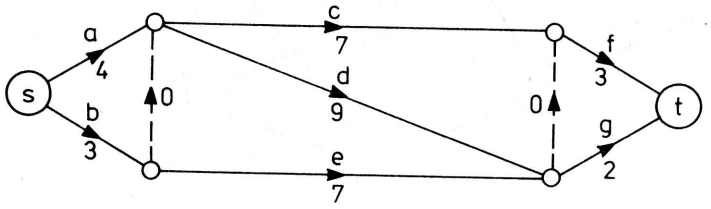
Digrafy tohto druhu sú však málo používané. Častejšie k projektu priraďujeme taký digraf, kde *činnosti* sú *reprezentované hranami*. Takýto digraf možno získať z predchádzajúceho jednoducho tým, že každý vrchol $u$ (okrem $s$ a $t$) nahradíme šípom $u_1u_2$. Potom tie šípy, ktoré prichádzali do $u$, budú prichádzať do $u_1$ a tie, čo z $u$ odchádzali, budú odchádzať z $u_2$. Takto každá činnosť je reprezentovaná šípom. Šípy zabezpečujúce technologické väzby nazývame *fiktívnymi činnosťami* (a zvyčajne ich zakresľujeme čiarkovane). Digraf z obr. 2 sa takto zmení na digraf z obr. 3.




Obr. 3. Reprezentácia činností šípmi.

V takomto digrafe môže byť mnoho fiktívnych hrán, a preto ho často zjednodušujeme kontrahovaním niektorých fiktívnych činností tak, aby sme zachovali pôvodné čiastočné usporiadanie činností. Obr. 4 ukazuje digraf z obr. 3 po takomto zjednodušení.

V ďalšom budeme pod *digrafom projektu* myslieť vždy taký, kde činnostiam zodpovedajú šípy. (Časté sú aj názvy *sieťový graf*, PERT *sieť*, CPM *sieť*). Vrcholy zodpovedajú udalostiam (ukončenie nejakých činností). Takýto digraf možno zostaviť aj priamo. Uspokojivý digraf však obyčajne zostavíme až po viacerých vylepšeniach, keďže nie vždy sú nám dané všetky technologické väzby.




Obr. 4. Digraf projektu po zjednodušení (trvania činností sú vyznačené).

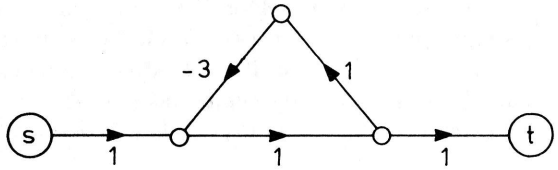




Nech každý šíp, ktorý predstavuje skutočnú činnosť, má dĺžku rovnú trvaniu tejto činnosti a fiktívne šípy nech majú nulové dĺžky. Potom vidieť, že ak najdlhšia $s$—$t$ cesta má dĺžku $T$, tak celý projekt má trvanie $T$. (Po zmene trvaní na záporné, budeme mať problém najkratšej cesty.) Vo všeobecnosti, činnosť $ij$ môže začať až v čase rovnom dĺžke najdlhšej $s$—$i$ cesty. Zistenie možných časov pre začatie, resp. ukončenie jednotlivých činností bez predĺženia trvania celého projektu, tvorí tzv. *časovú analýzu projektu*. Tu sa oboznámime len so základmi používaných metód. Pre praktické detaily je čitateľ odkázaný na špeciálnu literatúru ako napr. [Trávnik, Vlach (1975)].

## OBŤAŽNOSŤ ÚLOHY O NAJKRATŠEJ CESTE

V prvých dvoch úlohách, ktoré sme uviedli, všetky hrany mali nezáporné dĺžky. Vo štvrtej úlohe to boli tiež len nezáporné dĺžky, ale tam sme hľadali najdlhšiu cestu. Táto úloha je rovnocenná hľadaniu najkratšej cesty po zmene každej dĺžky $c_{ij}$ na $-c_{ij}$. Pre spomínané úlohy poznáme polynomiálne algoritmy (uvedieme ich v ďalších častiach), avšak pre štvrtú úlohu je to len vďaka acykličnosti digrafu projektu. Takéto algoritmy v skutočnosti hľadajú najkratší sled, ktorý vždy nemusí existovať, ako ukazuje obr. 5. Preto sa v algoritmoch vyskytne predpoklad neexistencie cyklu zápornej dĺžky.




Obr. 5. Najkratší $s$—$t$ sled neexistuje.

Ako sme už naznačili, úlohu v grafoch prevádzame do digrafov nahradením každého rebra dvojicou protišípov. Tým však z rebra zápornej dĺžky vzniká záporný cyklus, s čím si potom nevieme poradiť. Ak však graf neobsahuje cyklus zápornej dĺžky, možno úlohu riešiť ako istý problém faktoru. Preto tento prípad budeme vedieť riešiť až v kap. 8, hoci je zaradený v časti 4.D.

**Veta 1.** Problém najkratšej cesty je NP-ťažký dokonca pre kubické planárne grafy, resp. digrafy.

DÔKAZ. Pre takúto triedu grafov (resp. digrafov) je problém nájdenia hamiltonovskej cesty NP-ťažký, dokonca ak je daný jej začiatok $s$ a koniec $t$. To je

rovnocenné nájdeniu najdlhšej $s$—$t$ cesty, a teda nájdeniu najkratšej $s$—$t$ cesty, ak všetky hrany budú mať ohodnotenie $-1$. ■

Jedna myšlienka riešenia úlohy, nájsť najkratšiu $s$—$t$ cestu, by mohla byť takáto: Najprv nájdeme všetky $s$—$t$ cesty a z nich vyberieme najkratšiu. Prenechávame čitateľovi (4.Z.2) ukázať, že všetkých ciest môže byť exponenciálne mnoho (v závislosti od $m+n$), a teda taký postup nemožno pokladať za dobrý.

## 4.B
## Vzdialenosti z jedného vrcholu do ostatných

Základnou ideou pri hľadaní najkratších ciest je rekurencia (princíp optimality z dynamického programovania). Preto úloha nájsť najkratšiu $s$—$t$ cestu v digrafe $G$ pre dané $s$ a $t$, vyžaduje taký základný postup ako úloha nájsť najkratšiu $s$—$v$ cestu pre každé $v \in V_G$.

### MOOROV ALGORITMUS

Tento algoritmus hľadá vzdialenosti (a potom aj príslušné najkratšie cesty) z daného vrcholu $s$ do ostatných vrcholov v neohodnotenom digrafe (t. j., ak každá hrana má dĺžku 1). Algoritmus navrhol Moore (1957) a využíva sa tu postup do šírky. Vrcholom postupne priraďujeme značky. Vo všeobecnosti, vrchol $j$ dostane značku $(p_j, d_j)$, ak sme našli $s$—$j$ cestu dĺžky $d_j$, ktorej posledná hrana je $p_j j$. V skutočnosti bude $d_j = d(s, j)$ a algoritmus takto urobí *distančný rozklad* vrcholovej množiny z vrcholu $s$: $V_G = V_0 \cup V_1 \cup \ldots$, kde $V_k = \{i \in V \mid d(s, i) = k\}$.

**Algoritmus**

KROK 0: Vrchol $s$ dostáva značku $(0, 0)$, ostatné vrcholy sú neoznačkované, $k := 0$, $V_0 := \{s\}$.

KROK 1: Pre každé $i \in V_k$ urobíme:
[$V_{k+1} := \emptyset$. Každého neoznačkovaného nasledovníka $j$ vrcholu $i$ označkujeme: $(p_j, d_j) := (i, k+1)$, $V_{k+1} := V_{k+1} \cup \{j\}$.

KROK 2: Ak $V_{k+1} = \emptyset$, STOP. Inak $k := k+1$ a ideme na krok 1. ●

**Veta 1.** Po skončení Moorovho algoritmu platí: Ak vrchol $j$ je označkovaný, tak $d_j = d(s, j)$ a ak je neoznačkovaný, tak neexistuje žiadna $s$—$j$ cesta.

DÔKAZ. Z algoritmu vidíme, že každý vrchol dostáva značku nanajvýš raz. Ak vrchol $j$ má značku $(i, k+1)$, tak existuje $s$—$j$ cesta dĺžky $k+1$. Skutočne, takú





cestu získame zreťazením $s$—$i$ cesty dĺžky $k$ (ktorá existuje podľa indukčného predpokladu) a šípa $ij$. (Pre $k=0$ je tvrdenie zrejmé.)

Nech $d(s, j) = b$. Indukciou podľa $b$ ukážeme, že potom $j$ má značku, a pritom $d_j = b$. Nech $Q$ je $s$—$j$ cesta dĺžky $b$ a nech $i$ je jej predposledný vrchol. Potom $d(s, i) = b-1$ a podľa indukčného predpokladu $i$ je označkovaný, pričom $d_i = b-1$. Algoritmus však pri $k = b-1$ priradí vrcholu $j$ značku, pričom $d_j = k+1 = b$ (lebo podľa prvej časti dôkazu vrchol $j$ zostáva pri $k < b-1$ neoznačkovaný). ■

Hneď vidieť, že ak nás zaujíma len vzdialenosť $d(s, t)$ pre jeden vrchol $t$, tak algoritmus možno zastaviť ihneď po označkovaní vrcholu $t$.

Uvedený algoritmus ešte nedáva požadovanú $s$—$t$ cestu, avšak najkratšiu $s$—$t$ cestu možno určiť zo značiek veľmi ľahko. Nech $t = i_q$ a má značku $(i_{q-1}, d_{iq})$ a nech $i_{q-1}$ má značku $(i_{q-2}, d_{i_{q-1}}, \ldots$, až nech $i_1$ má značku $(i_0, d_{i_1})$, kde $i_0 = s$. Potom cesta $i_0 i_1 \ldots i_q$ je najkratšia $s$—$t$ cesta.

Zložitosť algoritmu. Predpokladajme, že $G$ je zadaný pomocou nasledovníkov vrcholov. Potom algoritmus pre rozklad prezerá každú hranu nanajvýš raz, a teda nevyžaduje viac ako $O(m)$ operácií. Nájdenie $s$—$t$ cesty nevyžaduje viac ako $O(n)$ operácií, a teda celkove Moorov algoritmus vystačí na nájdenie najkratšej $s$—$t$ cesty s $O(m+n)$ operáciami. Na nájdenie najkratších $s$—$v$ ciest pre všetky $v \in V_G$ potom stačí $O(mn+n^2)$ operácií. „Rozumný" digraf má $m \geq n$, a preto máme algoritmus zložitosti $O(m)$ pre jednu cestu a zložitosti $O(mn)$ v druhom prípade.

### DIJKSTROV ALGORITMUS

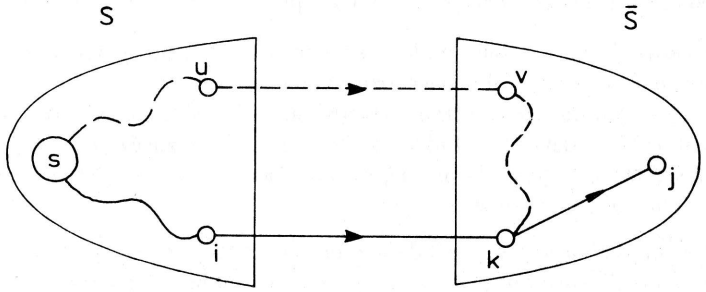
V nasledujúcom algoritme, ktorého autorom je Dijkstra (1959), predpokladáme, že dĺžka každej hrany $ij$ je $c_{ij} \geq 0$. Tu podobne ako v Moorovom algoritme vrchol $j$ dostáva značku $(p_j, d_j)$, ak sme našli $s$—$j$ cestu dĺžky $d_j$, ktorej predposledný vrchol je $p_j$. Tam boli značky trvalé, avšak tu sa môžu meniť (ak nájdeme kratšiu $s$—$j$ cestu). Teda značky budú dočasné a postupne ich budeme zlepšovať až sa zmenia na trvalé značky, t. j. potom už $d_j = d(s, j)$.

Na začiatku dostane vrchol $s$ trvalú značku $(0, 0)$ a každý iný vrchol $j$ dočasnú značku $(s, c_{sj})$, kde $c_{sj} = \infty$ pre $sj \notin E_G$. Nech v algoritme $S$ označuje množinu všetkých vrcholov, ktoré majú už permanentnú značku a $\bar{S} = V_G - S$. Na začiatku je $S = \{s\}$. Všeobecný krok algoritmu vyzerá takto (pozri obr. 1): V množine $\bar{S}$ nájdeme taký vrchol $k$, že $d_k = \min\{d_j \mid j \in \bar{S}\}$ a presunieme ho do $S$. Potom zostávajúce dočasné značky pozmeníme: pre každé $j$ z (novej) množiny $\bar{S}$ značku ponecháme, ak $d_j \leq d_k + c_{kj}$ a zmeníme na značku $(k, d_k + c_{kj})$ v opačnom prípade.

Správnosť tejto idey overíme indukciou. Predpokladajme, že pre každé $i \in S$ platí $d_i = d(s, i)$. Dokážeme, že ak $d_k$ je najmenšia dočasná značka, tak $d_k =$

$d(s, k)$. Nech existuje $s$—$k$ cesta $P$ dĺžky $c(P) < d_k$ a nech $v$ je jej prvý vrchol ležiaci v $\bar{S}$ (pozri obr. 1) a $u$ jeho predchodca v $P$. Potom z algoritmu máme: $d_v \leq d_u + c_{uv} \leq d(s, u) + c_{uv} \leq c(P) < d_k$, čo je spor s voľbou $k$.

Úplne rovnako ako v predošlom algoritme možno z trvalých značiek v čase $O(n)$ zistiť najkratšiu $s$—$t$ cestu, ak $d_t < \infty$. (Ak $d_t = \infty$, tak žiadna $s$—$t$ cesta neexistuje.) Preto zhrnieme iba algoritmus pre vzdialenosti.




Obr. 1. K Dijkstrovmu algoritmu ($s$—$k$ cesta $P$ je čiarkovaná).

**Algoritmus**

KROK 0: $S := \{s\}$, $\bar{S} := V_G - S$, $(p_s, d_s) := (0, 0)$ a $(p_j, d_j) := (s, c_{sj})$ pre $j \in \bar{S}$, kde $c_{sj} = \infty$ pre $sj \notin E_G$.

KROK 1: Nájdeme vrchol $k \in \bar{S}$ taký, že $d_k = \min \{d_j \mid j \in \bar{S}\}$. Ak $d_k = \infty$, STOP (žiadna $s$—$\bar{S}$ cesta neexistuje). $S := S \cup \{k\}$, $\bar{S} := \bar{S} - \{k\}$. Ak $\bar{S} = \emptyset$, STOP (dĺžka najkratšej $s$—$t$ cesty je $d_t$). Inak ideme na krok 2.

KROK 2: Pre každé $j \in \bar{S} \cap V^+(k)$ urobíme:
Ak $d_j > d_k + c_{kj}$, tak $(p_j, d_j) := (k, d_k + c_{kj})$ a inak značku nemeníme.
Ideme na krok 1. ●

Poznamenajme, že ak nás zaujíma iba vzdialenosť $d(s, t)$, tak výpočet možno ukončiť hneď potom, keď sa $t$ dostane do $S$. Preto je vhodné, keď pri minimalizácii v kroku 1 vrchol $t$ kandiduje na $k$ ako prvý.

Zložitosť algoritmu. Krok 0 je zanedbateľný. Krok 1 vyžaduje $O(n)$ operácií a robí sa $n-1$ raz, teda celkove $O(n^2)$ operácií. Podobne je to v kroku 2, a teda celkove má algoritmus zložitosť $O(n^2)$.

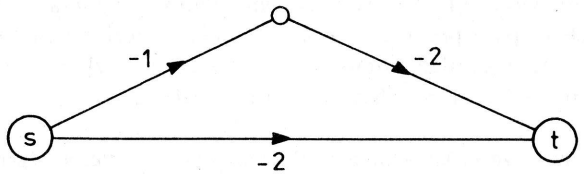
Poznámky o zložitosti. Hoci možno jemnejšie implementovať a odhadnúť





operácie v Dijkstrovom algoritme (4.Z.3), predsa zostáva zložitosť $O(n^2)$. Významnejšie príspevky pre prípad nezáporných dĺžok hrán však dávajú Wagner (1976) a D. B. Johnson(1977). Wagner dáva $O(m + n + d_{\max})$ algoritmus, kde $d_{\max} = \max \{d(s, v) \mid v \in V_G\}$. Keďže $d_{\max} \leq nc_{\max}$, kde $c_{\max} = \max \{c_{ij} \mid ij \in E_G\}$, tak máme $O(m + n + nc_{\max})$ algoritmus, čo pre riedke digrafy (t. j. $m \ll n^2$) a malé dĺžky hrán je lepšie ako Dijkstrov algoritmus. Aj Johnsonov príspevok je zameraný na riedke digrafy. Jeho algoritmus má zložitosť $O(\min \{n^{1+1/k} + m, (m+n) \log n\})$, kde $k$ je pevné prirodzené číslo.

Dijkstrov algoritmus je silne viazaný na predpoklad nezáporných dĺžok, ako ukazuje digraf z obr. 2, ktorý je dokonca acyklický.




Obr. 2. Dijkstrov algoritmus tu zlyháva.

Existujú pokusy rozšíriť Dijkstrov algoritmus pre digrafy bez cyklov zápornej dĺžky, ale pripúšťajúce záporné dĺžky hrán. Jeden taký návrh je v [Edmonds a Karp (1972)]. Avšak D. B. Johnson (1973) uvádza príklad, kde tento algoritmus vyžaduje exponenciálne mnoho operácií.

## FORDOV ALGORITMUS

Tento algoritmus nájde všetky vzdialenosti $d(s, v)$, pre $v \in V_G$, za predpokladu, že digraf $G$ neobsahuje cyklus zápornej dĺžky, avšak záporné dĺžky hrán sa pripúšťajú. Každému vrcholu $j \in V_G$ priradíme trojicu $(a_j, p_j, d_j)$, kde $a_j$ udáva počet hrán nejakej $s$—$j$ cesty $Q$, ktorá je z preskúmaných $s$—$j$ ciest najkratšia a spomedzi takýchto najkratších má najmenší počet hrán. Tak isto ako prv, $d_j = c(Q)$ a $p_j$ je predposledný vrchol cesty $Q$.

V nasledujúcom algoritme, za autora ktorého sa pokladá Ford (1956) a tiež Bellman (1958), sa najprv prezerajú $s$—$j$ cesty obsahujúce 1 hranu, potom 2 hrany atď. (premenná $k$ vyjadruje tento počet). Predpokladá sa, že digraf $G$ s dĺžkami hrán $c_{ij}$ je zadaný zoznamami nasledovníkov $V^+(i)$, $i \in V_G$.

**Algoritmus**

KROK 0: Vrcholu $s$ priradíme trojicu $(0, 0, 0)$ a každému vrcholu $j \neq s$ trojicu $(0, 0, \infty)$: $k := 0$.

KROK 1: ($k$-tá iterácia): Ak $k = n-1$, alebo ak $a_i \neq k$, pre každé $i \in V_G$, STOP. Inak, pre každé $i \in V_G$ také, že $a_i = k$, urobíme:

Pre každé $j \in V^+(i)$ urobíme:
Ak $d_j > d_i + c_{ij}$, tak urobíme:
$d_j := d_i + c_{ij}$,
$p_j := i$,
$a_j := a_i + 1$,
$k := k+1$ a ideme na krok 1. ●

Správnosť algoritmu vyplýva z významu značiek $(a_j, p_j, d_j)$, $j \in V_G$. Tvrdíme, že po skončení $k$-tej iterácie je $d_j$ rovné dĺžke najkratšej z takých $s$—$j$ ciest, ktoré majú nanajvýš $k+1$ hrán. To hneď vidieť indukciou podľa $k$. Pre $k = 0$ je to zrejmé a nech to platí pre $k-1$. V $k$-tej iterácii algoritmu sa každá dĺžka $d_j$ najkratšej $s$—$j$ cesty s počtom hrán $a_j \leq k$ porovnáva s dĺžkou každej $s$—$j$ cesty s počtom hrán $k+1$ a ak je dlhšia, tak sa ňou nahradí.

Zložitosť algoritmu. Algoritmus urobí nanajvýš $n-1$ iterácií (pre $k = n-1$ sa už zastaví). Na každú iteráciu stačí $O(m)$ operácií. Teda celkove má Fordov algoritmus zložitosť $O(mn) \leq O(n^3)$.

## ACYKLICKÉ DIGRAFY

Acyklický digraf nemá cykly vôbec, a preto tu možno úlohu riešiť najjednoduchšie pri ľubovoľných reálnych ohodnoteniach hrán. Budeme zisťovať vzdialenosti $d(s, j)$, kde $s$ je prameň digrafu $G$. Nasledujúci algoritmus má menšiu zložitosť ako Fordov a jeho princíp je všeobecne známy. Predkladaná implementácia vychádza z práce [Noltemeier (1976)]. Každému vrcholu $j$ priradíme trojicu $(r_j, p_j, d_j)$, kde $r_j$ bude označovať počet nepreskúmaných šípov prichádzajúcich do $j$: $p_j$ a $d_j$ majú význam ako prv ($d_j$ je dĺžka najkratšej nájdenej $s$—$j$ cesty a $p_j$ jej predposledný vrchol). Predpokladáme, že digraf $G$ je zadaný zoznamami nasledovníkov vrcholov. Okrem toho používame jeden pomocný zoznam $Z$ v tvare zásobníka.

### Algoritmus

KROK 0: Vrcholu $s$ priradíme trojicu (0, 0, 0) a každému vrcholu $j \neq s$ trojicu $(\deg_G^-(j), 0, \infty)$. Do zásobníka $Z$ dáme všetky pramene digrafu $G$ a vrchol $s$ ako posledný element.

KROK 1: Pre posledný element $i \in Z$ urobíme:





Odoberieme $i$ zo $Z$.
Pre každé $j \in V_G^+(i)$ urobíme:
Ak $d_j > d_i + c_{ij}$, tak urobíme:
$d_j := d_i + c_{ij}$,
$p_j := i$.
$r_j := r_j - 1$.
Ak $r_j = 0$ tak do $Z$ pridáme $j$ (ako posledný element).

KROK 2: Ak $Z = \emptyset$, STOP. Inak ideme na krok 1. ●

K dôkazu správnosti algoritmu stačí nahliadnuť, že po jeho skončení (ktoré je zaručené tým, že každý vrchol sa dostane do zásobníka práve raz) už bude $d_j = d(s, j)$ pre každé $j$. To je však pravda, lebo ak $Q$ je $s$—$j$ cesta dĺžky $d(s, j)$ a $i$ je jej predposledný vrchol, tak z toho, že $d_i = d(s, i)$ (indukčný predpoklad), vyplynie požadovaná rovnosť. Používame indukciu podľa poradia, v ktorom sme vrcholy vyberali zo $Z$.

Zložitosť algoritmu. Na zistenie prameňov, resp. prichádzajúcich stupňov v kroku 0 stačí $O(m+n)$ operácií. Každý krok 1 je riadený vrcholom $i$ a urobí sa v ňom $O(|E^+(i)|)$ operácií. Keďže krok 1 sa realizuje pre každý vrchol práve raz, tak celkove vystačíme s $O(m)$ operáciami. Preto celý algoritmus má zložitosť $O(m+n)$.

Poznamenajme, že pre zistenie *najdlhších* $s$—$j$ ciest stačí zmeniť znamienka všetkých dĺžok hrán (vynásobiť číslom $-1$). Pretože najdlhšie dráhy sa hľadajú napr. pri analýze činností v digrafoch projektov, je vhodné prerobiť algoritmus. Na to stačí všetky $d_j$ a $c_{ij}$ v ňom vynásobiť číslom $-1$. (Potom už algoritmus bude hľadať najdlhšie cesty na základe pôvodných údajov $c_{ij}$.)

Algoritmus pre najdlhšiu cestu možno použiť aj na *monotónne očíslovanie*, t. j. na usporiadanie vrcholov acyklického digrafu do postupnosti $v_1, v_2, \ldots, v_n$ tak, že pre každú hranu $v_i v_j$ platí: $i < j$ (pozri vetu 1.E.2 a algoritmus v 3.E). Zrejme postupnosť, v ktorej vrcholy opúšťajú zásobník $Z$, už má takú vlastnosť.

## 4.C
## Vzdialenosti z každého vrcholu do každého

V tejto časti skúmame problém nájdenia najkratšej $i$—$j$ cesty pre každú dvojicu vrcholov $i$, $j$ digrafu $G$. Táto úloha obsahuje predchádzajúcu, a preto predpoklad o neexistencii cyklov zápornej dĺžky je tu dôležitý. Uvádzame dva postupy. V prvom sa problém redukuje na úlohu z predchádzajúcej časti a v druhom je špeciálna metóda pre tento problém.

## TRANSFORMÁCIA DĹŽOK HRÁN

Úlohu vyriešime, ak vyriešime $n$ „jednokoreňových" úloh z predchádzajúcej časti tak, že zvolíme za $s$ postupne všetky vrcholy. Použitím Fordovho algoritmu by sme takto mali $O(n^4)$ algoritmus. Ak však $G$ má všetky hrany nezáporných dĺžok, tak môžeme použiť Dijkstrov algoritmus a celková zložitosť bude $O(n^3)$. Ukážeme, že transformáciou dĺžok hrán možno vždy zaručiť takúto zložitosť.

### Algoritmus

KROK 0: Zvolíme nejaký vrchol $s \in V_G$ a zistíme množinu vrcholov $D_s$ dosiahnuteľných z $s$ (pozri 3.E). Ak $D_s = V_G$, položíme $H := G$ a ideme na krok 1. Inak zvolíme číslo $b > (n-1)c_{\max}$, kde $c_{\max} := \max\{c_{ij} \mid ij \in E_G\}$, a pre každé $j \in V_G - D_s$ pridáme ku $G$ hranu $sj$ dĺžky $b$, čím získame digraf $H$.

KROK 1: Pre každé $j \in V_H$ zistíme Fordovým algoritmom vzdialenosť $d_j := d_H(s, j)$.

KROK 2: Pre každú hranu $ij \in E_H$ vypočítame $\hat{c}_{ij} := d_i + c_{ij} - d_j$ (čo je podľa Fordovho algoritmu vždy nezáporné).

KROK 3: Použitím Dijkstrovho algoritmu pri dĺžkach $\hat{c}_{ij}$ v $H$ urobíme:
Pre každé $i \in V_H$ zistíme:
$d_H(i, j)$ a príslušné najkratšie cesty $Q_{ij}$ pre všetky $j \in V_H$.

KROK 4: Pre každé $ij \in E_H$ urobíme:
$c(Q_{ij}) := \hat{c}(Q_{ij}) - d_i + d_j$. Ak $c(Q_{ij}) < b$, tak kladieme $d_{ij} := c(Q_{ij})$,
inak $d_{ij} := \infty$.
STOP ($d_{ij}$ sú hľadané vzdialenosti v $G$ a $Q_{ij}$ sú príslušné cesty). ●

Správnosť algoritmu je zrejmá a hneď aj vidieť, že má zložitosť $O(n^3)$. Myšlienka transformácie sa nachádza v práci [Edmonds a Karp (1972)] a na jej základe D. B. Johnson využíva svoj výsledok (spomínaný v predchádzajúcej časti) a dáva algoritmus zložitosti $O(\min\{n^{2+1/k} + nm,\ n^2 \log n + nm \log n\})$, kde $k$ je pevné prirodzené číslo.

## FLOYDOV ALGORITMUS

Nasledujúci algoritmus navrhol Floyd (1962) na základe práce [Warshall (1962)], s ktorou sme sa už stretli pri hľadaní tranzitívneho uzáveru.

Predpokladá sa, že je daná matica $D^{(0)}$ typu $(n, n)$ „priamych" vzdialeností $d_{ij}^{(0)}$, t. j.





$$d_{ij}^{(0)} = \begin{cases} 0, & \text{ak } i = j, \\ c_{ij}, & \text{ak } ij \in E_G, \\ \infty, & \text{inak.} \end{cases}$$

Algoritmus konštruuje z $D^{(0)}$ postupnosť matíc $D^{(1)}, D^{(2)}, \ldots, D^{(n)}$, kde pre každé $k \in \{1, \ldots, n\}$ prvok $d_{ij}^{(k)}$ matice $D^{(k)}$ je rovný dĺžke najkratšieho $i$—$j$ sledu, ktorého každý vnútorný (t. j. nekrajný) vrchol patrí do množiny $\{1, 2, \ldots, k\}$. Je zrejmé, že každý takýto sled alebo neobsahuje vrchol $k$, alebo je zreťazením najkratšieho $i$—$k$ sledu a najkratšieho $k$—$j$ sledu, pričom oba majú vnútorné vrcholy v množine $\{1, 2, \ldots, k-1\}$. Preto môžeme písať:

$$\text{(F)} \qquad d_{ij}^{(k)} = \min\{d_{ij}^{(k-1)},\ d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\}.$$

Špeciálne máme $d_{ik}^{(k)} = d_{ik}^{(k-1)}$ a $d_{kj}^{(k)} = d_{kj}^{(k-1)}$, lebo $d_{kk}^{(k-1)} = 0$ (neexistujú cykly zápornej dĺžky).

To znamená, že maticu $D^{(k-1)}$ môžeme transformovať na $D^{(k)}$ postupne po riadkoch, pritom $k$-tý riadok sa nezmení.

Samotné cesty možno potom určiť z výslednej matice $D^{(n)}$ rekurentne zo vzťahu: $d_{ij}^{(n)} = d_{iq}^{(n)} + d_{qj}^{(0)}$. Ak nás však zaujíma viac najkratších ciest, tak je dobré paralelne s $D^{(k)}$ budovať maticu $P^{(k)}$, ktorej element $p_{ij}^{(k)}$ označuje predposledný vrchol nejakého $i$—$j$ sledu dĺžky $d_{ij}^{(k)}$ a určíme ho takto: Ak sa vo vzťahu (F) realizuje minimum na $d_{ij}^{(k-1)}$, tak $p_{ij}^{(k)} = p_{ij}^{(k-1)}$; inak $p_{ij}^{(k)} = p_{kj}^{(k-1)}$. ●

Tým je Floydov algoritmus opísaný aj zdôvodnený. Na zostrojenie každej matice vystačíme s $O(n^2)$ operáciami, a teda celkove má algoritmus zložitosť $O(n^3)$.

## ZISŤOVANIE ZÁPORNÝCH CYKLOV

Doteraz sme neuviedli, ako zistiť, či digraf má *záporný cyklus* (t. j. cyklus zápornej dĺžky). Avšak Floydov algoritmus umožňuje riešiť aj takúto úlohu. Definujme maticu $D^{(0)}$ takto: $d_{ii}^{(0)} := c_{ii}$ ak $ij \in E_G$ a $d_{ij}^{(0)} := \infty$ inak. Ak podľa vzťahu (F) vypočítame postupne matice $D^{(1)}, D^{(2)}, \ldots, D^{(n)}$, tak zrejme prvok $d_{ij}^{(k)}$ je rovný dĺžke najkratšieho netriviálneho $i$—$j$ sledu, ktorého každý vnútorný vrchol patrí do množiny $\{1, 2, \ldots, k\}$. Preto platí:

**Veta 1.** (a) Digraf obsahuje záporný cyklus $\Leftrightarrow$ $d_{ii}^{(k)} < 0$ pre nejaké $i, k \in \{1, 2, \ldots, n\}$.

(b) Ak digraf neobsahuje záporný cyklus, tak $d_{ij}^{(n)} = d(i, j)$ pre $i \neq j$ a $d_{ii}^{(n)}$ je dĺžka najkratšieho $i$—$i$ cyklu (pre všetky $i, j \in \{1, 2, \ldots, n\}$).

Takto modifikovaný Floydov algoritmus má stále zložitosť $O(n^3)$. Poznamenaj-

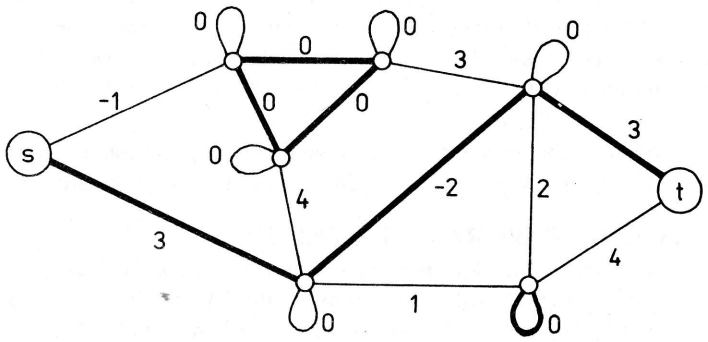
me, že záporný cyklus možno odhaliť aj v čase $O(\min\{n^{2+1/k}+nm,\ n^2 \log n + nm \log n\})$, kde $k$ je pevné prirodzené číslo [D. B. Johnson (1977)].

## 4. D
## Najkratší sled s ohraničeniami

Niekedy je potrebné nájsť najkratší zo sledov spĺňajúcich isté ohraničenia. Napr. sledy majú prechádzať, resp. neprechádzať predpísanými prvkami, alebo hrany majú ešte ďalšie ohodnotenie a vzhľadom na toto má sled byť tiež minimálny a pod.

### NAJKRATŠIA $s$—$t$ CESTA V GRAFE

Ak je daný graf $G$ s dĺžkami hrán $c_{ij} \geq 0$ a dva vrcholy $s$, $t \in V_G$, potom ľahko vieme nájsť najkratšiu $s$—$t$ cestu: Každé rebro $ij$ nahradíme dvojicou protišípov $ij$ a $ji$ s dĺžkami $c_{ij}$ a riešime problém v digrafe. Ak však $G$ má aj záporné dĺžky, táto transformácia nie je vhodná (vzniknú záporné 2-cykly).




Obr. 1. Transformácia problému najkratšej $s$—$t$ cesty na problém faktora.

Úloha najkratšej $s$—$t$ cesty v grafe pri záporných dĺžkach niektorých hrán má svoje opodstatnenie. Napr. pri výbere najlepšej trasy v cestnej sieti zodpovedajúcej grafu môžeme každému úseku $ij$ pripísať číslo $c_{ij}$ zodpovedajúce stratám (námahe a pod.) potrebným na prekonanie úseku $ij$, pritom niektoré $c_{ij}$ môžu byť aj záporné (zaujímavé úseky prírodných krás spôsobujú osvieženie, ak nimi prechádzame prvý raz). Ak z takýchto dôvodov tiež nechceme tým istým mestom prechádzať viac razy, tak máme problém najkratšej $s$—$t$ cesty. Pravda, takýto





problém nevieme vždy riešiť, ale ak graf nemá cykly zápornej dĺžky, možno postupovať takto [Lawler (1976)]:

Ku každému vrcholu $i \neq s$, $t$ pridáme slučku dĺžky $c_{ii} = 0$ a vo vzniknutom pseudografe $\hat{G}$ hľadáme najlacnejší faktor $F$ so stupňami: $\deg_F(s) = \deg_F(t) = 1$ a $\deg_F(i) = 2$, pre každé $i \neq s$, $t$ (dĺžky hrán považujeme za ceny). Ak z $F$ vynecháme všetky cykly (všetky sú nulovej dĺžky), tak získame hľadanú $s$—$t$ cestu. ● Obr. 1 ilustruje uvedenú transformáciu.

Vzniknutý problém faktora tu však nebudeme riešiť. Polynomiálny algoritmus pre tento problém uvedieme v časti 8.D.

### NAJKRATŠÍ SLED S MINIMÁLNYM POČTOM ŠPECIÁLNYCH HRÁN

Klee a Larman (1979) rozširujú Floydov algoritmus pre najkratší sled v digrafoch bez záporných cyklov pri rôznych ohraničeniach. Tu uvedieme len jednu z úloh, ktoré riešili.

V digrafe $G$ bez cyklov zápornej dĺžky je daná množina $M$ špeciálnych hrán a číslo $r$. Pre každé $q \leq r$ a každé dva vrcholy $i, j \in V_G$ treba nájsť najkratší $i$—$j$ sled obsahujúci nanajvýš $q$ špeciálnych hrán.

Nech pre $k = 0, 1, \ldots, n$ je $D^{(k)}(q)$ matica typu $(n, n)$, pričom jej prvok $d_{ij}^{(k)}(q)$ znamená dĺžku takého najkratšieho $i$—$j$ sledu, ktorého každý vnútorný vrchol leží v množine $\{1, 2, \ldots, k\}$ a ktorý obsahuje nanajvýš $q$ špeciálnych hrán. Zrejme platí:

$$\text{(KL)} \qquad d_{ij}^{(k)}(q) = \min\{d_{ij}^{(k-1)}(q),\ d_{ik}^{(k-1)}(q_1) + d_{kj}^{(k-1)}(q_2)\},$$

kde minimalizácia prebieha cez všetky usporiadané rozklady čísla $q$ na dve čísla $q_1, q_2 \geq 0$. (Napr. pre $q = 2$ sú tri možnosti: $0+2$, $1+1$ a $2+0$.) Vo všeobecnosti je práve $q+1$ takých rozkladov a toľko treba robiť výpočtov pre (KL). Keďže $q$ prebieha hodnoty $0, 1, \ldots, r$, tak ide o $O(r^2)$ operácií pre pevné $i, j, k \in \{1, 2, \ldots, n\}$. Teda na celý algoritmus vystačíme s $O(n^3 r^2)$ operáciami. ●

### NAJKRATŠÍ SLED PRI ČASOVOM OHRANIČENÍ

Tu sa predpokladá, že je daný digraf $G$, kde každá hrana $ij$ má dve ohodnotenia, dĺžku $c_{ij}$ a prechodový čas $t_{ij}$. Úloha je takáto:

Pre daný vrchol $s \in V_G$ a číslo $T$ treba pre každé $j \in V_G$ nájsť najkratší $s$—$j$ sled s celkovým prechodovým časom nepresahujúcim $T$.

Túto úlohu študoval Lawler (1967, 1976) a za predpokladu, že každé $t_{ij}$ je celé kladné číslo, dáva nasledujúci algoritmus (pre ľubovoľné reálne čísla $c_{ij}$).

Nech $d_j(t)$ je dĺžka najkratšieho $s$—$j$ sledu s prechodovým časom nanajvýš $t$. Potom zrejme môžeme písať rekurentné vzťahy:

$$\text{(L)} \qquad \begin{cases} d_j(t) = \infty, \text{ ak } t<0, \\ d_s(0) = 0, \\ d_j(t) = \min\{d_j(t-1), \min\limits_k \{d_k(t-t_{kj}) + c_{kj}\}\}. \end{cases}$$

Tieto vzťahy umožňujú postupne pre $t=0, 1, \ldots, T$ vypočítať všetky $d_j(t)$. Pre každé $t$ vystačíme s $O(n^2)$ operáciami, a teda máme $O(n^2T)$ algoritmus na riešenie celej úlohy. (Poznamenajme, že ak $T=n-1$ a každé $t_{ij}=1$, tak máme klasickú situáciu najkratšieho sledu a algoritmus bude v podstate Fordov algoritmus.) Nájdenie príslušných sledov vyžaduje len malú úpravu, ktorú už poznáme. ●

Ak chceme riešiť úlohu pre všetky možné dvojice vrcholov, tak stačí vyriešiť $n$ jednokoreňových úloh, a máme teda $O(n^3T)$ algoritmus. Ak sú však všetky $t_{ij} \leqslant t_0$, kde $t_0 \ll T/\log T$, tak metóda predložená v [Lawler (1976)] dáva $O(n^3 t_0^2 \log T)$ algoritmus. Lawler tiež uvádza nasledujúci zaujímavý príklad na uvedenú úlohu.

Treba nájsť najkratšiu trasu z $u$ do $v$ v cestnej sieti pre automobil, ak v sieti je len $q$ čerpacích benzínových staníc ($u$ a $v$ sú medzi nimi). S plnou nádržou automobil prejde len $T$ kilometrov, a preto jeho trasa musí prechádzať stanicami tak, aby medzi dvoma po sebe idúcimi stanicami nebola väčšia vzdialenosť ako $T$.

Riešenie možno nájsť v dvoch etapách. Najprv uvedeným spôsobom vypočítame všetky hodnoty $d_{ij}(T)$ (t. j. $s=i$). Nech potom $H$ je kompletný digraf na $q$ vrcholoch, kde každá hrana $ij$ má dĺžku $d_{ij}(T)$. Potom dĺžka najkratšieho $u$—$v$ sledu v $H$ (bez ohraničení) je zrejme rovná dĺžke hľadanej trasy v pôvodnej cestnej sieti.

Poznámka o zložitosti. Čitateľ si iste všimol, že čas $O(n^2T)$ nie je polynomiálny (lebo $T$ možno vyjadriť pomocou $O(\log T)$ bitov), ale len „pseudopolynomiálny". Avšak polynomiálny algoritmus sotva možno očakávať, keďže Megiddo (pozri [Garey a Johnson(1979)]) ukázal, že takýto problém je NP-ťažký.

**Veta 1.** Problém nájdenia $u$—$v$ sledu $Q$ dĺžky $c(Q) \leqslant L$ a s prechodovým časom $t(Q) \leqslant T$, kde $u$, $v$ sú dané vrcholy a $L$, $T$ dané čísla, je NP-ťažký, dokonca ak všetky $c_{ij}$ a $t_{ij}$ sú prirodzené čísla a problém je v grafoch alebo acyklických digrafoch, ktoré sú planárne, bipartitné a s maximálnym stupňom 3.

DÔKAZ. Dáme polynomiálnu transformáciu problému rozdelenia multimnožiny čísel $a_1, a_2, \ldots, a_p$ na dve triedy, každú so súčtom $b = \Sigma\, a_i/2$, na našu úlohu.



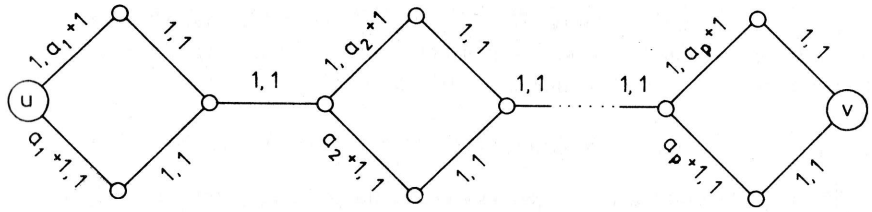


Uvažujme graf z obr. 2, kde na každej hrane $ij$ máme dvojicu $c_{ij}$, $t_{ij}$. Žiadajme $L := T := b + 3p - 1$. Zrejme sa môžeme ohraničiť na $u$—$v$ cesty $Q$. Každá takáto cesta $Q$ prechádza alebo hornou alebo dolnou časťou nejakého štvoruholníka. Takto ak $I \subseteq \{1, 2, \ldots, p\}$ je množina indexov tých štvoruholníkov, ktorými $Q$ prechádza v dolnej časti, tak

$$c(Q) = \sum_{i \in I} a_i + 3p - 1,$$

$$t(Q) = \sum_{i \in I} a_i + 3p - 1.$$

Zrejme $c(Q) \leqslant b + 3p - 1$ a súčasne $t(Q) \leqslant b + 3p - 1$ práve vtedy, ak $\sum_{i \in I} a_i = b$. Orientujúc všetky rebrá smerom od $u$ ku $v$, máme dôkaz aj pre digrafy. ■




Obr. 2. K dôkazu vety 1.

## 4.E
## Generovanie zoznamu najkratších sledov

Niekedy môžu byť obmedzenia na sledy veľmi zložité, alebo dokonca aj nie celkom jasné. Vtedy sa uchyľujeme k zostaveniu nejakých $M$ najkratších sledov medzi danými dvoma vrcholmi $u$ a $v$. Z tohto zoznamu sa potom vyberá vyhovujúci sled. Pod $M$ najkratšími $u$—$v$ sledmi rozumieme $M$ rozličných $u$—$v$ sledov vybraných tak, že každý iný $u$—$v$ sled je aspoň taký dlhý ako ľubovoľný z týchto $M$ sledov.

### $M$ NAJKRATŠÍCH SLEDOV

Nasledujúca metóda, ktorú navrhol Dreyfus (1969), dáva $M$ najkratších $1$—$j$ sledov pre každé $j=2, \ldots, n$. Predpokladáme, že hľadáme sledy v digrafe $G$, kde

$V_G = \{1, 2, \ldots, n\}$ a každá hrana $ij$ má dĺžku $c_{ij}$. Pritom nech $G$ je bez cyklov zápornej dĺžky.

Táto metóda vyžaduje, aby pre každé $s$ a $j$ bola množina $s$ najkratších 1—$j$ sledov jednoznačne určená. Preto musíme usporiadať sledy rovnakej (ohodnotenej) dĺžky: napr. postupnosti ich vrcholov lexikograficky usporiadame. Takto napr. z dvoch 1—6 sledov 153216 a 15346 rovnakej ohodnotenej dĺžky prehlásime prvý za kratší.

Nech $d_j(s)$ označuje dĺžku $s$-tého najkratšieho 1—$j$ sledu a nech $p_{kj}(s)$ označuje počet takých 1—$j$ sledov v množine $s$ najkratších 1—$j$ sledov, v ktorých je hrana $kj$ posledná.

Zrejme $p_{kj}(0) = 0$ a hneď vidieť, že platí rekurentný vzťah:

$$\text{(D1)} \qquad p_{kj}(s+1) = \begin{cases} p_{kj}(s)+1, & \text{ak } kj \text{ je poslednou hranou v } (s+1)\text{-om} \\ & \text{najkratšom 1—}j \text{ slede,} \\ p_{kj}(s), & \text{inak.} \end{cases}$$

Ťažšie sa počítajú hodnoty $d_j(s)$. Nech $kj$ je posledná hrana v $(s+1)$-om najkratšom 1—$j$ slede. Dĺžka 1—$k$ úseku tohto sledu je $d_k(p_{kj}(s)+1)$. Takže berúc minimum cez všetky možné $k$, získame vzťah

$$\text{(D2)} \qquad d_j(s+1) = \min_k \{d_k(p_{kj}(s)+1) + c_{kj}\}, \text{ pričom } d_1(1) = 0.$$

Vzťah (D2) však nemožno priamo využiť na počítanie, lebo $d_j(s+1)$ môže implicitne závisieť od $d_k(s+1)$, a to zasa od $d_j(s+1)$. Počet hrán v slede je však výbornou pomôckou umožňujúcou organizáciu počítania. Totiž $d_k(s+1)$ sa objavuje na pravej strane vzťahu (D2), iba ak hrana $kj$ je poslednou hranou v každom z $s$ najkratších 1—$s$ sledov. Keďže hrana $kj$ je poslednou hranou takéhoto sledu, tak jeho 1—$k$ úsek má o jednu hranu menej. Preto po počiatočnom výpočte hodnôt $d_j(1)$ (napr. pomocou Dijkstrovho alebo Fordovho algoritmu), počítame najprv tie z hodnôt $d_j(2)$, kde sa hodnoty $d_j(1)$ realizovali najmenším počtom hrán, potom najbližším väčším počtom hrán atď. Úplne analogicky počítame hodnoty $d_j(3)$ atď. Takto vzťah (D2) umožňuje vypočítať všetky $d_j(s)$. ●

Na výpočet hodnôt $d_j(1)$ ($j = 2, 3, \ldots, n$) vystačíme s $O(n^3)$ operáciami (Fordov algoritmus). Na výpočet jednej hodnoty $d_j(s)$ podľa (D2) stačí $O(n)$ operácií, čo pre všetky $j = 2, 3, \ldots, n$ znamená $O(n^2)$ operácií. Takže na výpočet $d_j(s)$ pre všetky $s = 1, 2, \ldots, M$ stačí $O(n^3 + Mn^2)$ operácií. Vhodnou štruktúrou dát možno počet operácií $O(n^2)$ v jednej iterácii zredukovať na $O(n \log n)$ (porovnaj [Lawler (1976)]).





Všetky hodnoty $d_j(s)$ treba uložiť v pamäti, čo je spolu $O(Mn)$ miest. Okrem toho v každej iterácii treba $m$ miest na hodnoty $p_{kj}(s)$. Ak predpokladáme, že pre každé $s$ $s$-tý najkratší 1—$j$ sled dáme na výstup a viac neudržujeme v pamäti, tak celkove nám stačí pamäť rozsahu $O(Mn + m)$.

## $M$ NAJKRATŠÍCH CIEST

V mnohých reálnych úlohách sa vyžadujú nie hocijaké sledy, ale cesty. Nech digraf $G$ s $V_G = \{1, 2, \ldots, n\}$ je bez záporných cyklov. Úlohou je nájsť $M$ najkratších 1—$n$ ciest.

Základ nasledovného algoritmu pochádza od Yena (1971), ale predkladaná verzia je modifikáciou, ktorú publikoval Lawler (1972).

Najprv nájdeme nejakú najkratšiu 1—$n$ cestu. Nech je to cesta $P_1 = 1v_2v_3 \ldots v_qn$, ktorá na začiatku tvorí obsah zoznamu $Z$. Túto cestu odstránime zo $Z$ a „vyrobíme" z nej niekoľko ciest takto: Nájdeme najkratšiu 1—$n$ cestu neobsahujúcu hranu $1v_2$, potom najkratšiu 1—$n$ cestu obsahujúcu hranu $1v_2$ a neobsahujúcu $v_2v_3$. Vo všeobecnosti hľadáme najkratšiu 1—$n$ cestu spomedzi 1—$n$ ciest obsahujúcich 1—$v_i$ úsek cesty $P_1$ a neobsahujúcich hranu $v_iv_{i+1}$. Jednoducho stačí nájsť najkratšiu $v_i$—$n$ cestu v digrafe $G - \{1, v_2, \ldots, v_{i-1}\} - v_iv_{i+1}$, kde $1 \leq i \leq q$, pričom $v_1 = 1$ a $v_{q+1} = n$. Takto získame nanajvýš $q$ 1—$n$ ciest (pre každé $i$ nemusí existovať) z cesty $P_1$ a všetky ich uložíme do zoznamu $Z$ spolu s informáciou: „predpísaný úsek a zakázané hrany".

Cesta $P_1$ je prvá najkratšia 1—$n$ cesta a druhá najkratšia sa nachádza v $Z$. Preto vyberieme najkratšiu z ciest v $Z$, označíme ju $P_2$ (druhá najkratšia vôbec) a odstránime ju zo $Z$. Z cesty $P_2$ vyrobíme nové cesty (podobne ako z $P_1$) a všetky dáme do $Z$. Z tých všetkých, čo sú v $Z$, nájdeme najkratšiu a označíme $P_3$ (tretia najkratšia vôbec) atď. Takto postupujeme, pokiaľ nezískame $P_M$, resp. skončíme prv, ak $Z$ bude prázdne, a teda už nebude z čoho vyrábať nové cesty.

### Algoritmus

KROK 0: Ak sú všetky dĺžky hrán nezáporné, tak pre vyhľadávanie najkratších ciest môžeme používať Dijkstrov algoritmus. V opačnom prípade jednorazové použitie Fordovho algoritmu umožňuje urobiť transformáciu dĺžok hrán na nezáporné (pozri časť 4.C), a potom už stačí používať len Dijkstrov algoritmus. Nájdeme najkratšiu 1—$n$ cestu $P_1$ a dáme ju do $Z$. Predpísaný úsek pre túto cestu je triviálny (vrchol 1) a zakázané hrany tvoria prázdnu množinu. Položíme $s := 1$.

KROK 1: Ak $Z$ je prázdne, STOP (neexistuje viac ciest). Inak nájdeme

najkratšiu z ciest v $Z$, označíme ju $P_s$ a vynecháme zo $Z$. Na výstup dáme $P_s$. Ak $s = M$, STOP (našli sme už $M$ najkratších 1—$n$ ciest).

KROK 2: Nech $P_s = 1v_2v_3 \ldots v_q n$ a nech $P_s$ má predpísaný úsek 1—$v_p$ ($p \leq q$) a nech $F_s$ je zakázaná množina šípov odchádzajúcich z $v_p$. (Pre $s = 1$ je $p = 1$ a $F_s = \emptyset$.) Potom pri každom z nasledujúcich $q - p + 1$ ohraničení nájdeme najkratšiu 1—$n$ cestu a každú spolu s ohraničením „predpísaný úsek a zakázané hrany" dáme do $Z$.

(0) Predpísaný je 1—$v_p$ úsek z $P_s$ a $F_s \cup \{v_p v_{p+1}\}$ je zakázaná množina hrán.

(1) Predpísaný je 1—$v_{p+1}$ úsek z $P_s$ a $v_{p+1}v_{p+2}$ je zakázaná hrana.

⋮

(i) ($1 \leq i < q - p$.) Predpísaný je $1 - v_{p+i}$ úsek z $P_s$ a hrana $v_{p+i}v_{p+i+1}$ je zakázaná.

⋮

(q—p) Predpísaný je 1—$v_q$ úsek z $P_s$ a hrana $v_q n$ je zakázaná.

Položíme $s := s + 1$ a ideme na krok 1. ●

Je ľahké ukázať, že uvedený systematický prieskum „nezabudne" na žiadnu krátku 1—$n$ cestu, a teda, že algoritmus pracuje správne.

Všimnime si zložitosť algoritmu. Na nájdenie $M$-tej najkratšej cesty treba vykonať $M - 1$ iterácií (kroky 1 a 2). V každej iterácii sa $O(n)$ razy používa Dijkstrov algoritmus (lebo $q - p + 1$ môže byť až $n - 1$). To je spolu $O(Mn^3)$ operácií, aj keď sme nerátali rôzne drobné výpočty: tieto sú však dominované týmto počtom. Sem sme však nezahrnuli manipuláciu so zoznamom $Z$. Vhodné je použiť na $Z$ prioritný front, ktorý umožňuje odstránenie, resp. vloženie najkratšej cesty v čase $O(\log M)$. V každej iterácii treba vložiť, resp. vybrať nie viac ako $n$ ciest: teda celkove na to stačí $O(Mn \log M)$ operácií. Keďže vždy $M \leq n!$, tak $\log M \leq n \log n$. Preto vystačíme s $O(Mn^2 \log n)$ vkladaniami, resp. vyberaniami zo $Z$. Každá cesta vyžaduje na záznam $O(n)$ operácií. Preto na toto, a teda aj celkove, vystačíme s $O(Mn^3 \log n)$ operáciami.

Na záver poznamenajme, že algoritmus postupne generuje prvú najkratšiu 1—$n$ cestu, potom druhú najkratšiu atď., takže ho môžeme hocikedy prerušiť, a teda číslo M nie je potrebné poznať vopred.





## 4. F
## Niektoré ďalšie optimálne sledy

Doteraz sme uvažovali sledy, kde hlavným, resp. jediným kritériom optimality bola ich dĺžka. Teraz uvedieme niekoľko iných kritérií.

### NAJŠIRŠIA CESTA

Daný je digraf $G$, v ktorom každá hrana $ij$ má priradené nezáporné číslo $b_{ij} = b(ij)$, tzv. *šírku* (alebo *priepustnosť, kapacitu* a pod.). Potom pod *šírkou* nejakej *cesty* rozumieme šírku najužšej hrany (t. j. hrany s minimálnou šírkou) v nej. Nech sú dané vrcholy $s, t \in V_G$: úlohou je nájsť $s$—$t$ cestu najväčšej šírky.

Úlohy takéhoto druhu možno niekedy riešiť jednoduchým spôsobom. Teraz možno postupovať napr. takto: Najprv usporiadame hrany do neklesajúcej postupnosti podľa šírky: $b(e_1) \leq b(e_2) \leq \ldots \leq b(e_m)$. V digrafe $G_0 := G$ hľadáme nejakú $s$—$t$ cestu $P_0$. Potom v digrafe $G_1 := G_0 - e_1$ hľadáme nejakú $s$—$t$ cestu $P_1$. Vo všeobecnosti hľadáme v digrafe $G_i := G_{i-1} - e_i$ $s$—$t$ cestu $P_i$. Ak $i$ je maximálne číslo, pre ktoré $P_i$ existuje, tak $P_i$ je zrejme najširšia $s$—$t$ cesta. ●

Uvedený algoritmus má zložitosť $O(m^2)$. Teraz uvedieme rýchlejší spôsob načrtnutý v práci [Edmonds, Karp (1972)] a opísaný v [Adeľson–Veľskij, Dinic, Karzanov (1975)].

Hľadanú cestu skonštruujeme pomocou značkovacej metódy. V skutočnosti nájdeme najširšiu $s$—$j$ cestu pre každé $j \in V_G$.

Na začiatku $s$ dostáva značku $(-, \infty)$ a ostatné vrcholy sú neoznačkované. Vo všeobecnosti označkovaný vrchol $j$ má značku tvaru $(i, b_j)$, čo značí, že poznáme $s$—$j$ cestu šírky $b_j$ a jej predposledný vrchol je $i$. Ak $S$ je množina vrcholov už označkovaných a $\bar{S} := V_G - S$ množina neoznačkovaných, pričom $t \in \bar{S}$, tak ďalší vrchol označkujeme takto: Nech $ij$ je taká hrana, že $i \in S$, $j \in \bar{S}$ a pritom číslo $r_j := \min\{b_i, b_{ij}\}$ je maximálne. Potom vrchol $j$ dostáva značku $(i, r_j)$, t. j. $b_j = r_j$.

Aby sme pred označkovaním vrcholu $j$ nemuseli hľadať hranu $ij$ podľa definície, tak je vhodné dávať vrcholom z $S$ aj dočasné značky $(i, r_j)$. Potom pred trvalým označkovaním zistíme len maximálne $r_j$ (na čo stačí $O(n)$ operácií) a jedna dočasná značka sa stane trvalou. Po označkovaní vrcholu $j$ však ešte treba vypočítať a pozmeniť dočasné značky $(u, r_v)$ pre $v \in \bar{S}$ takto: Zistíme $r'_v = \min\{b_j, b_{jv}\}$: ak $r'_v > r_v$, tak $(u, r_v)$ sa zmení na $(j, r'_v)$ a inak sa nezmení. Na to stačí $O(n)$ operácií. Celkove $n$ vrcholov dostane trvalú značku, a teda na celý algoritmus vystačíme s $O(n^2)$ operáciami. ●

Správnosť algoritmu je zrejmá. Poznamenajme, že ak $G$ je graf, tak

pomocou najcennejšej kostry možno nájsť najširšiu $i$—$j$ cestu pre každú dvojicu $[i, j]$ (pozri 5.Z.20).

## CESTA S NAJVÄČŠOU OČAKÁVANOU ŠÍRKOU

Nech je $G$ digraf, kde každá hrana $ij$ má dve ohodnotenia: *spoľahlivosť* $p_{ij}$, $0 < p_{ij} \leq 1$ (vyjadruje pravdepodobnosť správnej činnosti komunikačnej linky) a *šírku* $b_{ij} > 0$. Pre dané dva vrcholy $s$, $t \in V_G$ treba nájsť $s$—$t$ sled s najväčšou očakávanou šírkou. Pritom *očakávaná šírka sledu* $Q$ je

$$e(Q) = \min_{ij \in Q} b_{ij} \prod_{ij \in Q} p_{ij}.$$

V tomto vzorci je min $b_{ij}$ šírka sledu $Q$ a $\Pi\, p_{ij}$ je spoľahlivosť (pravdepodobnosť správnej činnosti) sledu $Q$.

Keďže každý optimálny sled obsahuje optimálnu cestu, budeme hovoriť len o cestách. Táto úloha zahŕňa v sebe úlohu o najspoľahlivejšej ceste (všetky $b_{ij} = 1$) aj úlohu o najširšej ceste (všetky $p_{ij} = 1$), ktoré sme už riešili. Nasledujúcu metódu dáva Christofides (1975).

**Algoritmus**

KROK 0: Nájdeme najširšiu $s$—$t$ cestu v $G$: nech jej šírka je $b^*$. Ďalej v $G$ nájdeme najspoľahlivejšiu $s$—$t$ cestu $Q_0$: nech má spoľahlivosť $p_0$ a šírku $b_0$. Nech $D := Q_0$ ($D$ je kandidát na cestu s najväčšou očakávanou šírkou). Položíme $k := 0$, $G_0 := G$.

KROK 1: Ak $e(D) \geq b^* p_k$, STOP ($D$ je optimálna cesta).
Inak utvoríme digraf $G_{k+1} := G_k - \{ij \mid b_{ij} \leq b_k\}$.
Položíme $k := k + 1$ a ideme na krok 2.

KROK 2: V $G_k$ nájdeme najspoľahlivejšiu $s$—$t$ cestu $Q_k$: nech jej spoľahlivosť je $p_k$ a šírka $b_k$. Ak $s$—$t$ cesta neexistuje, STOP ($D$ je optimálna cesta). Ak $e(D) < e(Q_k)$, tak $D := Q_k$. Ideme na krok 1. ●

Dokážeme správnosť algoritmu. Ak optimálna cesta má väčšiu očakávanú šírku, ako má $Q_0$, tak musí mať šírku väčšiu ako $b_0$, lebo $Q_0$ je najspoľahlivejšia $s$—$t$ cesta. Teda potom optimálna cesta leží v $G_1$. Analogický argument platí pre $Q_1$ atď. Keďže $D$ je optimálna cesta spomedzi $Q_0$, $Q_1$, ..., $Q_k$, tak $D$ bude výsledok. Zrejme ak $e(D) \geq b^* p_k$, tak môžeme skončiť, lebo už v $G_k$, a teda ani v $G_{k+1}$ lepšiu cestu ako $D$ nenájdeme.

Počet iterácií (kroky 1 a 2) nepresiahe $m$ a v každej iterácii stačí $O(n^2)$ operácií





(Dijkstrovým algoritmom nájdeme najspoľahlivejšiu cestu). Celkove máme $O(mn^2)$ algoritmus.

## CYKLUS S MAXIMÁLNYM ZISKOM ZA ČASOVÚ JEDNOTKU

Nasledujúci problém skúmali Dantzig, Blattner a Rao (1967) a Lawler (1967, 1976): Nákladný parník si môže vyberať prístavy, ktoré navštívi, a poradie, v ktorom ich navštívi. Ak pláva z prístavu $i$ do prístavu $j$, tak zarobí $z_{ij}$ korún (zisk) a potrebuje na to $t_{ij}$ dní (nakladanie, prevoz a vykladanie). Ktoré prístavy má parník navštíviť a v akom poradí, aby maximalizoval svoj priemerný denný zisk?

Hneď je vidieť, že riešenie problému nájdeme, ak zistíme taký cyklus, v ktorom pomer celkového zisku k celkovému času je maximálny. Parník potom bude stále obiehať tento cyklus (začať môže z ľubovoľného prístavu tohto cyklu). Ako model sa nám zrejme hodí digraf. Úlohu prevedieme na minimalizačnú tak, že definujeme ceny $c_{ij} := -z_{ij}$. Teda budeme hľadať taký cyklus $Z$ v digrafe, aby podiel

$$\Big(\sum_{ij \in Z} c_{ij}\Big) \Big/ \Big(\sum_{ij \in Z} t_{ij}\Big)$$

bol minimálny.

Ďalej budeme predpokladať, že každý cyklus $C$ má kladné sumárne časové ohodnotenie. (Pre uvedenú úlohu je to vždy pravda.)

Najprv odhadnime minimálnu hodnotu pomeru číslom $r$. Potom môžeme určiť nové ceny $\hat{c}_{ij} := c_{ij} - r t_{ij}$ a uvážime tri možné prípady pri tomto ohodnotení:

(1) Existuje záporný cyklus $Z$, t. j.

$$\sum_{Z} \hat{c}_{ij} < 0.$$

(2) Neexistuje záporný cyklus, ale existuje nulový cyklus $Z$, t. j.

$$\sum_{Z} \hat{c}_{ij} = 0.$$

(3) Každý cyklus je kladný, t. j.

$$\min_{Z} \sum_{Z} \hat{c}_{ij} > 0.$$

Prvý prípad dáva:

$$\sum_{Z} (c_{ij} - r t_{ij}) < 0,$$

a teda

$$\sum_Z c_{ij} \Big/ \sum_Z t_{ij} < r,$$

takže odhadnutá hodnota $r$ je príliš veľká a na $Z$ sa realizuje lepšia hodnota.

V druhom prípade analogicky vidíme, že $r$ je optimálna hodnota a realizuje sa cyklom $Z$.

V treťom prípade je $r$ príliš malé.

Na týchto pozorovaniach možno založiť výpočtovú procedúru: Zvolíme dostatočne veľké $r_0$: z prvého prípadu potom získame nejaké $r_1$, potom opäť z prvého prípadu $r_2$ atď., pričom $r_0 > r_1 > r_2 > \ldots$ Keďže hodnoty $r_1, r_2, \ldots$ sa realizujú na cykloch, a tých je len konečne mnoho, tak niekedy musí vzniknúť prípad 2 a máme optimum nájdené. Poznamenajme, že cykly $Z$ vieme hľadať v čase $O(n^3)$ (pozri 4.C).

Uvedená postupnosť však môže byť pomerne dlhá. K cieľu nás rýchlejšie dovedie metóda bisekcie (rozpoľovania): Predpokladáme, že poznáme interval $[a, b]$, v ktorom leží optimálna hodnota. Najprv skúšame $r = (a+b)/2$, a tým interval „neurčitosti" $[a, b]$ zmenšíme na polovičný $[a, (a+b)/2]$, ak vznikol prípad (1), resp. $[(a+b)/2, b]$, ak máme prípad (3). V prípade (2) máme dokonca optimálne riešenie. Získaný interval potom rozpolíme atď. Opakujúc $(k-1)$-krát toto delenie, získame interval neurčitosti dĺžky $(b-a)/2^k$, čo pre dostatočne veľké $k$ je už malé a v praxi stačí. Teoreticky tu však môže pre veľké $k$ ležať iba jeden podiel, lebo cyklov je len konečne mnoho. ●

Aby sme získali odhad počtu operácií, budeme predpokladať, že všetky čísla $c_{ij}$ a $t_{ij}$ sú celé a že $|c_{ij}| \leq c_0$ a $|t_{ij}| \leq t_0$. Takto pre prvý interval $[a, b]$ stačí brať $a := -nc_0$, $b := nc_0$. Okrem toho vidíme, že pre dva rôzne cykly $Z$ a $Z'$ je rozdiel optimálnych hodnôt aspoň $1/(n^2 t_0^2)$. Skutočne, ak $C/T$ a $C'/T'$ sú príslušné pomery, tak

$$\left| \frac{C}{T} - \frac{C'}{T'} \right| = \left| \frac{CT' - C'T}{TT'} \right| \neq 0.$$

Posledný podiel je najmenší, ak $CT' - C'T = 1$ a $TT' = n^2 t_0^2$.

Z toho vyplýva, že rozličných pomerov nie je viac ako $2nc_0/(1/(n^2 t_0^2)) = 2n^3 c_0 t_0^2$. Preto uvedená postupnosť $r_1, r_2, \ldots$ nemá viac prvkov, a teda taký algoritmus nevyžaduje viac ako $O(n^6 c_0 t_0^2)$ operácií. Ak $c_0$, $t_0$ nezávisia od $n$, tak to dáva $O(n^6)$ algoritmus.

Lepšia situácia je v prípade metódy bisekcie. Na túto stačí interval $[a, b]$ rozpoliť nie viac ako $\log_2 (b-a) - \log_2(1/(n^2 t_0^2))$ razy, čo je $O(\log n + \log c_0 + \log t_0)$. Nanajvýš toľkokrát treba hľadať záporný cyklus. Celkove máme teda $O(n^3(\log n + \log c_0 + \log t_0))$ algoritmus, teda polynomiálny, lebo dĺžka vstupu nášho problému je aspoň $n + \log c_0 + \log t_0$.





## 4.Z
## Cvičenia a poznámky

1. (a) Sformulujte úlohu o celočíselnom batohu max $\{c_1 x_1 + \ldots + c_p x_p \mid a_1 x_1 + \ldots + a_p x_p \leq b$, všetky $x_i \geq 0$ a celé$\}$ ako úlohu o najdlhšej ceste v acyklickom digrafe. [Návod: Môžeme predpokladať, že $a_i \neq a_j$ pre $i \neq j$. Zostrojíme digraf $G$ s $V(G) = \{0, 1, \ldots, b\}$ a $E(G) = E_1 \cup \ldots \cup E_p$, kde $E_i = \{jk \mid k - j = a_i, 0 \leq j < k \leq b\}$. Každý šíp z $E_i$ nech má dĺžku $c_i$. Ukážte, že najdlhšia cesta v $G$ začínajúca v 0 zodpovedá optimálnemu riešeniu úlohy o batohu [Garfinkel, Nemhauser (1972)]. Táto transformácia však nie je polynomiálna.]
   (b) Sformulujte úlohu o binárnom batohu ako úlohu o najdlhšej ceste vo vhodnom acyklickom digrafe s $p(b+1)$ vrcholmi [Fulkerson (1966)].
2. Ukážte, že existuje kladné reálne číslo $k$ a nekonečná postupnosť digrafov tak, že pre každý z nich platí: Počet všetkých najkratších $s$—$t$ ciest je aspoň $2^{k(m+n)}$.
3. (a) Urobte presnejší odhad počtu operácií v Dijkstrovom algoritme (pre kompletný digraf) a navrhnite implementáciu zameranú na zníženie počtu operácií. (Yen (1972) dáva modifikáciu s $n^2/2$ sčítaniami a s $n^2$ porovnaniami.)
   (b) Navrhnite $O((m+n)\log n)$ implementáciu Dijkstrovho algoritmu (výhodnú pre riedke digrafy). [Návod: Čísla $d_i$ udržiavame usporiadané podľa veľkosti (v prioritnom fronte) a v kroku 1 berieme za $d_k$ vždy jeho prvý prvok. V kroku 2 stačí $O(\log n)$ operácií na presunutie jedného čísla $d_j$ na správne miesto, čo je spolu $O(\deg^+(k) \log n)$ operácií pre jedno $k$: za celý algoritmus to bude $O(m \log n)$ operácií.]
4. Ukážte, že aj nasledujúca zjednodušená verzia Fordovho algoritmu nájde všetky vzdialenosti z vrcholu $s$.
   KROK 0: Vrchol $s$ dostane dvojicu $(0, 0)$ a každý iný vrchol dostane $(0, \infty)$.
   KROK 1: Ak $d_j \leq d_i + c_{ij}$ pre každé $ij \in E_G$, tak STOP. Inak volíme $ij \in E_G$ také, že $d_j > d_i + c_{ij}$ a urobíme:
   $d_j := d_i + c_{ij}$,
   $p_j := i$.
   Ideme na krok 1. ●

Avšak takýto algoritmus môže vyžadovať exponenciálne mnoho iterácií aj pri nezáporných dĺžkach šípov [D. B. Johnson (1977)].

5. Navrhnite $O(m+n)$ algoritmus pre nájdenie najdlhšej cesty v acyklickom digrafe s reálnymi dĺžkami hrán. (Kraje cesty nie sú dané.)
6. Uvažujte digraf projektu, kde každá činnosť $ij$ má trvanie (dĺžku) $T_{ij}$. Nech $T$ označuje *trvanie celého projektu* (t. j. dĺžka najdlhšej $s$—$t$ cesty). Každému vrcholu $i$ priradíme dve čísla: $a_i$-dĺžka najdlhšej $s$—$i$ cesty a $b_i$-dĺžka najdlhšej $i$—$t$ cesty. Potom pre každú činnosť $ij$ možno určiť: *najskôr možný začiatok* $T_{ij}^{(Z1)}$, *najneskôr možný začiatok* $T_{ij}^{(Z2)}$, *najskôr možný koniec* $T_{ij}^{(K1)}$ a *najneskôr možný koniec* $T_{ij}^{(K2)}$ (pri dodržaní technologických väzieb a za predpokladu, že projekt začína v čase 0 a končí v čase $T$). Vyjadrite tieto časy pomocou $a_i$, $a_j$, $b_i$, $b_j$. Číslo $r_{ij} := T_{ij}^{(K2)} - T_{ij}^{(Z1)} - T_{ij}$ sa nazýva *rezerva činnosti* $ij$. Ak $r_{ij}=0$, *činnosť* $ij$ sa nazýva *kritická*. Ukážte, že činnosť je kritická práve vtedy, keď leží na nejakej najdlhšej („kritickej") $s$—$t$ ceste. Výpočet uvedených časových údajov sa nazýva *časová analýza projektu* (a je známa pod názvami: CPM, resp. PERT). Ukážte, že celú časovú analýzu možno urobiť v čase $O(m+n)$.
   Poznamenajme, že tu všade sme predpokladali neohraničenosť zdrojov (strojov, pracovníkov a pod.), takže činnosti môžu prebiehať súčasne. Ak sú však zdroje ohraničené, časová analýza je NP-ťažkou úlohou, ako aj mnohé iné úlohy „rozvrhovania" (pozri napr. [Garey, Johnson (1979)]. Niektoré heuristiky možno nájsť v [Davis, Patterson (1975)].
   Úlohou skrátenia trvania projektu sa zaoberáme v 6.Z.14.
7. Navrhnite analógiu Floydovho algoritmu pre určenie všetkých vzdialeností, kde postupne pre $k=1, \ldots, n$ budete počítať $d_{ij}^{(k)} := d_{G(\{1,2,\ldots,k\})}(i,j)$. $O(n^3)$ algoritmus založený na tejto myšlienke predložil Dantzig (1966).
8. Nech acyklický digraf má kladné dĺžky $c_{ij}$ aj kladné trvania $t_{ij}$ hrán. Ďalej nech sú dané čísla $C$ a $T$. Dokážte, že problém nájsť cestu dĺžky aspoň $C$ a trvania aspoň $T$ je NP-ťažký. [Návod: Použite dôkaz vety 4.D.1.]
9. Navrhnite algoritmus pre riešenie *úlohy o preprave nadmerného nákladu:* Pre každý úsek komunikácie je daná jeho dĺžka a maximálny rozmer nákladu, ktorý tadiaľ možno prepraviť; tiež je to známe pre križovatky v jednotlivých smeroch prechádzania. (Rozmer sa obvykle skladá z niekoľkých zložiek: hmotnosť, výška, šírka a pod.). Úlohou je pre dané miesta $s$, $t$ a náklad daného rozmeru nájsť najkratšiu trasu z $s$ do $t$.



# 5
# STROMY A KOSTRY

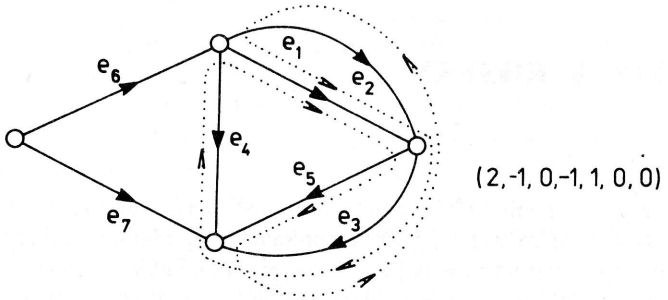
Stromy majú významné postavenie v celej teórii grafov aj v grafových algoritmoch. Zvlášť sa uplatňujú vo funkcii minimálnych podgrafov zabezpečujúcich cesty. Táto kapitola je v prevažnej miere venovaná *kostrám,* t. j. takým faktorom, ktoré sú stromy. Dominujúce postavenie má úloha o najlacnejšej kostre v grafe. Oboznámime sa aj s niektorými zovšeobecneniami, kde sa na kostru kladú ohraničenia. Tiež uvádzame algoritmus pre generovanie zoznamu $K$ najlacnejších kostier, čo môže byť istým východiskom pri hľadaní najlacnejšej kostry s komplikovanými ohraničeniami. Ďalej hľadáme tzv. najlacnejšiu zdrojovú kostru digrafu. Kapitolu končíme Steinerovým problémom, ktorý zodpovedá úlohe nájsť najlacnejšiu spojovaciu sieť daných bodov. Úvodná časť je venovaná sľúbenému postupu na určovanie nezávislých okruhov v elektrických sieťach pomocou kostier.

## 5.A
## Kostra a nezávislé polocykly

V časti 1.A sme spomenuli, ako Kirchhoff (1847) určoval nezávislé okruhy v elektrických obvodoch pri ich analýze, presnejšie, pri aplikovaní Kirchhoffovho napäťového zákona.

Analyzovanú elektrickú sieť môžeme reprezentovať súvislým pseudografom, ale ak každú hranu nejako zorientujeme, čo zodpovedá vyznačeniu elektrických prúdov, získame pseudodigraf $G$. Nech $E(G) = \{e_1, e_2, \ldots, e_m\}$. Potom ku každému uzavretému polosledu $Q$ (čo zodpovedá okruhu v elektrickej sieti) priraďme vektor $p=(p_1, p_2, \ldots, p_m)$, kde $p_i = p_i^+ - p_i^-$, pričom $p_i^+$ je počet súhlasných výskytov šípa $e_i$ v $Q$ a $p_i^-$ počet nesúhlasných výskytov šípa $e_i$ v $Q$ (pozri obr. 1). Systém uzavretých polosledov sa nazýva *nezávislý* (resp. *závislý*), ak príslušný systém vektorov je lineárne nezávislý (resp. závislý). Takto je úlohou nájsť v danom súvislom pseudodigrafe $G$ maximálny počet nezávislých uzavretých polosledov.

Kirchhoff navrhol zostrojiť najprv ľubovoľnú kostru $T$. (Ignorujúc orientáciu šípov v $G$, postup do hĺbky alebo do šírky nám v lineárnom čase nájde strom (pozri 3.A), ktorý po zavedení pôvodnej orientácie možno zobrať za $T$.) Potom pre každú chordu $e$, t. j. hranu $e \in E(G) - E(T)$, vyberieme jeden polocyklus $Z_e$




Obr. 1. Uzavretý polosled a príslušný vektor.

pseudodigrafu $T+e$. Takto získame presne $m-(n-1)$ polocyklov, ktoré tvoria lineárne nezávislý systém, lebo každý polocyklus $Z_e$ obsahuje takú hranu (hranu $e$), ktorá sa v iných polocykloch nevyskytuje. Kirchhoff dokázal, že viac nezávislých polosledov nemôže existovať:

**Veta 1.** Maximálny počet $\nu(G)$ nezávislých uzavretých polosledov v súvislom pseudodigrafe $G$ je $m-n+1$.

DÔKAZ. Postupujeme indukciou podľa $m$. Ak $m=n-1$, tak každému uzavretému polosledu zodpovedá nulový vektor. Nech $T$ je nejaká kostra v $G$ tvorená hranami $e_m, e_{m-1}, \ldots, e_{m-n+2}$ a nech $e_1, e_2, \ldots, e_{m-n+1}$ sú hrany mimo kostry. Nech $G_k = T + e_1 + e_2 + \ldots + e_k$. Podľa indukčného predpokladu $\nu(G_{k-1}) = k-1$ a nech $Z_1, Z_2, \ldots, Z_{k-1}$ sú nezávislé uzavreté polosledy v $G_{k-1}$. Pridaním hrany $e_k$ získame $G_k$ a vzniknú nové uzavreté polosledy $C_1, C_2, \ldots, C_s$ (prechádzajúce hranou $e_k$). Aspoň jeden z týchto polosledov je polocyklus. Môžeme predpokladať, že je to $C_1$ a že zložka $p_k(C_1)=1$. Zrejme $Z_1, Z_2, \ldots, Z_{k-1}, C_1$ tvoria lineárne nezávislý systém. Avšak pre každé $i \geq 2$ už $C_i$ na týchto polosledoch závisí. Skutočne, vektor $p(C_i)-p_k(C_i)p(C_1)$ zodpovedá uzavretému polosledu, ktorý vznikol z $C_i$ tým, že vždy namiesto hrany $e_k$ používame obrátený úsek polosledu $C_1$ bez $e_k$. Toto je polosled v $G_{k-1}$, a preto uvedený vektor je lineárnou kombináciou vektorov $Z_1, \ldots, Z_{k-1}$, a teda $p(C_i)$ je lineárnou kombináciou vektorov $Z_1, \ldots, Z_{k-1}, C_1$. Preto $\nu(G_k)=k$. ■

Ak $G$ je pseudomigraf s $t$ komponentmi, tak číslo $m-n+t$ sa nazýva cyklomatické číslo pseudomigrafu $G$. Podľa predchádzajúceho pre pseudodigraf $G$ platí: $m-n+t = \nu(G)$.

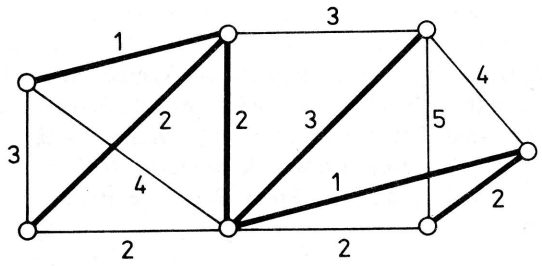




Ďalšie informácie o použití teórie grafov pri skúmaní elektrických obvodov nájde čitateľ napr. v knihe [Čulík, Doležal, Fiedler (1967)].

## 5.B
## Najlacnejšia kostra

V danom grafe $G$, kde každá hrana $e \in E(G)$ má reálne ohodnotenie (cenu) $c(e)$, treba nájsť *najlacnejšiu kostru*, t. j. takú kostru $T$ grafu $G$, že jej cena $c(T)$ (súčet cien jednotlivých hrán v $T$) je minimálna. Na obr. 1 je príklad grafu a najlacnejšej kostry (hrubé čiary).




Obr. 1. Najlacnejšia kostra.

S úlohou nájsť najlacnejšiu kostru sa možno stretnúť aj v praxi, napr. pri zostavovaní najlacnejšej súvislej siete prenosových liniek pre sieť televíznych vysielačov. Podobne je to pri rozvode elektriny, plynu a pod., ak žiadame, aby rozvetvenia neexistovali mimo zdroja a odberateľov (napr. kvôli ľahšej údržbe a kontrole). Tu všade treba nájsť najlacnejší súvislý faktor kompletného grafu, kde cena hrany $uv$ vyjadruje predpokladané náklady na vybudovanie linky z $u$ do $v$. Tieto náklady sú vždy nezáporné, a preto môžeme predpokladať, že optimálny súvislý faktor bude bez cyklov, teda kostra. (Vynechaním ľubovoľnej hrany z cyklu sa cena faktora nezväčší.)

Problém najlacnejšej kostry patrí medzi staršie úlohy na grafoch. Ako prvý sa ním zaoberal Borůvka (1926) inšpirovaný úlohou navrhovania optimálnej elektrovodnej siete. Borůvka dáva aj polynomiálny, ale dosť komplikovane formulovaný algoritmus. Jeho postup zlepšil Jarník (1930). Medzi ďalších, ktorí sa týmto zaoberali, patrí Choquet (1938), avšak až po Kruskalovi (1956) sa problém stal dostatočne známy. Dobrý prehľad starších príspevkov možno nájsť v knihe [Čulík, Doležal a Fiedler (1967)].

Poznamenajme, že úlohu možno zaviesť aj pre pseudografy, ale vždy ju možno redukovať na grafy.

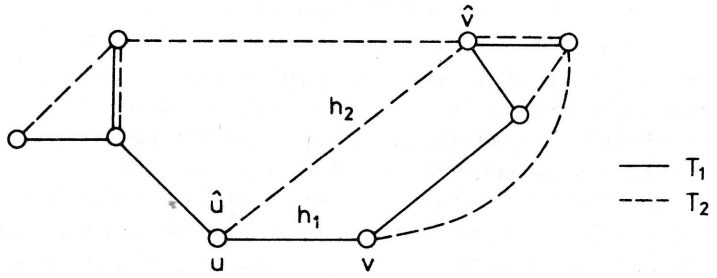
## ZÁKLADNÉ TVRDENIA

Nasledujúce výsledky možno nájsť explicitne alebo implicitne v prácach mnohých autorov. (Kvôli jednoduchosti často píšeme $G-H$ namiesto $G-E(H)$, $e \in G$ namiesto $e \in E(G)$ a pod.)

Podľa vety 1.E.3(2), ak $T$ je kostra grafu $G$, tak pre ľubovoľnú hranu $h \in G-T$ má graf $T+h$ jediný cyklus.

**Veta 1.** Nech $T_1$ a $T_2$ sú kostry grafu $G$. Potom pre každú hranu $h_1 \in T_1 - T_2$ existuje hrana $h_2 \in T_2 - T_1$ tak, že obe hrany ležia na cykle grafu $T_2 + h_1$ aj na cykle grafu $T_1 + h_2$.

DÔKAZ. Nech $h_1 = uv$. Graf $T_1 - h_1$ má práve dva komponenty: $G_u$(obsahujúci $u$) a $G_v$ (obsahujúci $v$) (porovnaj obr. 2). Nech $P_2$ je (jediná) $u$—$v$ cesta v $T_2$. Potom existuje hrana $h_2 = \hat{u}\hat{v} \in P_2$ tak, že vrchol $\hat{u} \in G_u$ a vrchol $\hat{v} \in G_v$. Potom $P_2 + h_1$ je (jediný) cyklus v $T_2 + h_1$ a zrejme obsahuje hranu $h_2$. V $T_1$ existuje jediná $\hat{u}$—$\hat{v}$ cesta $P_1$, a tá obsahuje hranu $h_1$. Preto $P_1 + h_2$ je (jediný) cyklus v $T_1 + h_2$ a obsahuje $h_1$. ■




Obr. 2. K dôkazu vety 1.

Ak $T$ je kostrou grafu $G$ a hrana $x \in G-T$, tak $T+x$ má jediný cyklus $Z$ a vynechaním ľubovoľnej hrany $y \in Z$ získame kostru $T+x-y$. Ak naviac $c(x) < c(y)$, tak $c(T+x-y) < c(T)$, a teda máme menšiu (lacnejšiu) kostru. Vtedy hovoríme, že kostru $T$ sme *zmenšili výmenou* hrany $x$ (za $y$). Dve kostry sa nazývajú *susedné*, ak možno jednu získať z druhej výmenou jednej hrany (niekedy stručne hovoríme: *sused kostry*).





**Veta 2** (*o optimalite kostry*). Kostra $T$ grafu $G$ je najlacnejšia $\Leftrightarrow$ $T$ nemožno zmenšiť výmenou hrany (t. j. ak neexistuje lacnejšia susedná kostra).

DÔKAZ. ($\Rightarrow$) Zrejmé. ($\Leftarrow$) Nech $S_0$ je nejaká najlacnejšia kostra. Ak $S_0 = T$, niet čo dokazovať, a preto nech existuje hrana $h_1 \in T - S_0$. Podľa vety 1 existuje hrana $h_2 \in S_0 - T$ tak, že $h_1$ a $h_2$ ležia aj na cykle v $S_0 + h_1$ aj na cykle v $T + h_2$. Z optimality $S_0$ máme $c(h_2) \leq c(h_1)$ a z predpokladu vety o $T$ je $c(h_2) \geq c(h_1)$, teda $c(h_1) = c(h_2)$. Preto kostra $S_1 := S_0 + h_1 - h_2$ je opäť najlacnejšia, a pritom má viac spoločných hrán s $T$ ako kostra $S_0$. Ak $S_1 \neq T$, tak analogicky možno celý postup zopakovať s $S_1$ namiesto $S_0$ a vytvoriť kostru $S_2$ atď. Takto možno vytvoriť takú najlacnejšiu kostru $S_k$, že $S_k = T$, a teda $T$ je tiež optimálna kostra. ■

Uvedená veta už naznačuje istý algoritmus: Nájdeme ľubovoľnú kostru a postupne vymieňame hrany, až získame optimálnu kostru. Avšak takýto algoritmus je zložitejší ako iné známe postupy. Väčšina známych algoritmov sa opiera o nasledujúcu vetu. Najprv zavedieme jeden pojem.

Nech $G$ je graf a $\emptyset \neq U \subset V_G$. Potom množina hrán $[U, V_G - U]_G := \{xy \mid xy \in E_G, x \in U, y \in V_G - U\}$ sa nazýva (hranový) rez grafu $G$.

**Veta 3.** Hrana $e$ grafu $G$ leží v nejakej najlacnejšej kostre $\Leftrightarrow$ $e$ je najlacnejšia hrana nejakého rezu grafu $G$.

DÔKAZ. ($\Rightarrow$) Nech $e$ leží v najlacnejšej kostre $T$. Potom $T-e$ pozostáva z dvoch stromov $T_1$ a $T_2$. Zrejme, že $[V(T_1), V(T_2)]_G$ je rezom a obsahuje $e$. Ak by v tomto reze bola lacnejšia hrana $f$, tak $T-e+f$ by bola lacnejšia kostra grafu $G$, ako je $T$.

($\Leftarrow$) Nech $e = uw$ je najlacnejšou hranou rezu $[U, W]_G$, $u \in U$, $w \in W$ a nech $T$ je najlacnejšia kostra. V $T$ existuje jediná $u$—$w$ cesta, a tá zrejme obsahuje aspoň jednu hranu $xy$ takú, že $x \in U$, $y \in W$. Podľa predpokladu je $c(uw) \leq c(xy)$, a preto kostra $T - xy + uw$ je tiež najlacnejšia a obsahuje hranu $uw = e$. ■

## VŠEOBECNÝ SPÁJACÍ ALGORITMUS

Predchádzajúca veta je ideovým základom nasledujúceho „spájacieho" algoritmu [Cheriton a Tarjan (1976)].

**Algoritmus**

KROK 0: Nech $F_0$ je nulový faktor v $G$: $i := 0$, $j := 0$, $G_0 := G$. Ak $n > 1$, ideme ľubovoľne na krok 2 alebo 3.

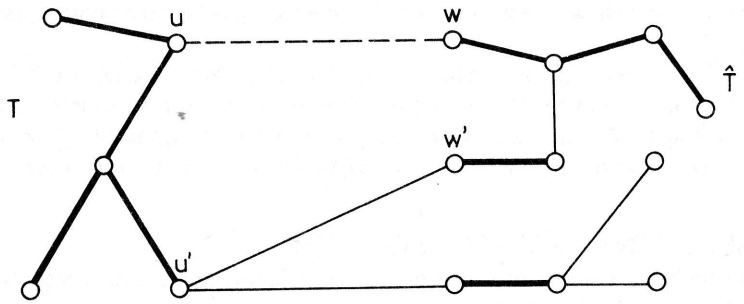
KROK 1: Ak $i = n-1$, STOP ($F_i$ je hľadaná kostra).

KROK 2 (*spájací*): Zvolíme nejaký strom $T$ lesa $F_i$ a vyberieme najlacnejšiu hranu $e$ rezu $[V(T), V_G - V(T)]_{G_j}$: $F_{i+1} := F_i + e$. Ak $e$ neexistuje, STOP ($G$ je nesúvislý, a teda nemá kostru). $i := i+1$.

KROK 3 (*očistný*): Utvoríme graf $G_{j+1}$ tak, že z $G_j$ vynecháme nejaké (žiadne, resp. aj všetky) z nestromových hrán spájajúcich vrcholy toho istého stromu v $F_i$ a tiež nejaké z hrán spájajúcich dva rôzne stromy v $F_i$ s výnimkou jednej najlacnejšej hrany: $j := j+1$ a ideme na krok 1. ●

**Veta 4.** Všeobecný spájací algoritmus pracuje správne.

DÔKAZ. (1) Najprv dokážeme, že ak sa v očistnom kroku nikdy nevynecháva hrana (t. j. $G_j = G$) a $G$ je súvislý graf, tak les $F_{n-1}$ je najlacnejšou kostrou grafu $G$. Za tým účelom dokážeme indukciou, že pre každé $i = 0, 1, \ldots, n-1$ les $F_i$ je obsiahnutý v nejakej najlacnejšej kostre. Pre $F_0$ je tvrdenie zrejmé. Uvažujme les $F_i$ ($0 \leq i < n-1$) a v súlade s krokom 2 nejaký jeho strom $T$ a najlacnejšiu hranu $e = uw$, kde $u \in V(T)$ a $w \in V_G - V(T)$: nech $w$ leží v strome $\hat{T}$. Podľa indukčného predpokladu $F_i$ leží v nejakej najlacnejšej kostre $S$. Ak hrana $uw$ leží tiež v $S$, potom $F_{i+1} \subseteq S$ a dôkaz je hotový. Inak je situácia podobná ako na obr. 3, kde hrany v $S$ sú vyznačené plnými a z toho hrany v $F_i$ hrubými čiarami. V $S$ existuje jediná $u$—$w$ cesta, a tá musí obsahovať nejakú hranu $u'w'$, kde $u' \in T$, $w' \notin T$. Z voľby hrany $uw$ máme $c(uw) \leq c(u'w')$, a preto kostra $S' = S + uw - u'w'$ je tiež najlacnejšia a obsahuje $F_{i+1}$, čo sme chceli dokázať.




Obr. 3. K dôkazu vety 4.

(2) Naše tvrdenie však platí aj vtedy, keď v očistnom kroku nejaké hrany vynechávame. Totiž algoritmus aj tak žiadnu z týchto hrán nevyberie, resp. nemusí vybrať a ak $G$ je súvislý, tak aj každý graf $G_j$ bude súvislý. ■





Uvedený algoritmus je načrtnutý príliš hrubo, takže ťažko by bolo hovoriť o jeho zložitosti. Teraz uvedieme dva staršie známe algoritmy, ktoré možno uvažovať ako špeciálne prípady všeobecného spájacieho algoritmu.

## KRUSKALOV ALGORITMUS

Kruskal (1956) navrhuje pred samotným hľadaním kostry usporiadať všetky hrany do postupnosti tak, aby $c(e_1) \leq c(e_2) \leq \ldots \leq c(e_m)$. Na to stačí $O(m \log m)$ operácií (porovnaj 2.A). Potom postupne berieme hrany $e_1, e_2, \ldots$, a skúšame ich pridať k okamžitému lesu $F_i$: ak $F_i + e_j$ je lesom, tak $F_{i+1} := F_i + e_j$, inak $e_j$ vynecháme. ●

Tento postup možno vtesnať do všeobecného spájacieho algoritmu, ak po utvorení $F_{i+1}$, v očistnom kroku vynecháme všetky hrany $e_{j+1}, \ldots, e_s$, ktoré sú chordami stromov v $F_{i+1}$, ale hrana $e_{s+1} = uw$ už spája dva rôzne stromy. Potom v spájacom kroku volíme za $T$ strom obsahujúci $u$ (alebo $w$).

Stromy v $F_i$ môžeme reprezentovať množinami ich vrcholov. Na zistenie druhu hrany $uw$ stačí zistiť, či $u$ a $w$ patria do tej istej množiny alebo nie. V druhom prípade potom príslušné množiny zjednotíme. Toto hľadanie a zjednocovanie za celý algoritmus možno ľahko urobiť v čase $O(n^2)$, ale použitím vhodných štruktúr dát a manipulácií, možno získať čas $O(m \log n)$, resp. ešte lepší [Tarjan (1975)]. Takto na celý Kruskalov algoritmus vystačíme s $O(m \log n)$ operáciami.

## PRIMOV ALGORITMUS

Nasledujúci algoritmus sa prisudzuje Primovi (1957). V skutočnosti však jeho ideu využíva už Jarník (1930). Nezávisle ho znovu objavuje aj Dijkstra (1959). Na báze tejto myšlienky bol potom zostrojený aj všeobecný spájací algoritmus, ktorého je špeciálnym prípadom, a možno ho charakterizovať takto:

V spájacom kroku vždy za $T$ volíme strom s maximálnym počtom vrcholov (ostatné sú jednovrcholové). V očistnom kroku žiadne hrany nevynechávame. ●

Odhadneme zložitosť algoritmu. Najlacnejšiu hranu vyberáme spomedzi nanajvýš $m$ hrán a takých výberov robíme $n-1$, čo spolu predstavuje nanajvýš $O(mn) \leq 0(n^3)$ operácií. Avšak už Dijkstra (1959) ukazuje nasledujúcu $O(n^2)$ implementáciu:

Nech $V(G) = \{1, 2, \ldots, n\}$, $d_j := \min \{c(ij) \mid i \in T\}$ a $t_j \in T$ je taký vrchol, kde $c(t_j, j) = d_j$. Na začiatku strom aj množina $T$ pozostáva z vrcholu 1, $t_j = 1$ a $d_j = c(1j)$ pre $j = 2, 3, \ldots, n$. Potom vo všeobecnosti, ak hľadáme najlacnejšiu hranu rezu $[T, V_G - T]_G$, stačí nájsť najmenšie $d_j$ ($j \notin T$) (na to stačí $O(n)$ operácií), nech je to $d_k$, a potom zobrať hranu $t_k k$. Potom bude $T := T \cup \{k\}$

a kladieme: ak $c(kj) < d_j$, tak $d_j := c(kj)$ a $t_j := k$, $(j \notin T)$: inak $d_j$ a $t_j$ nemeníme. Na toto tiež vystačíme s $O(n)$ operáciami. Takto na celý algoritmus ($n-1$ iterácií) stačí $O(n^2)$ operácií.

Poznamenajme, že pre husté grafy (t. j. ak $m$ je porovnateľné s $n^2$) ani lepší algoritmus nie je možný, avšak pre riedke grafy (ak $m \ll n^2$) je napr. Kruskalov algoritmus lepší. Ale aj Primov algoritmus možno implementovať na $O(m \log n)$ algoritmus [Kerschenbaum a Van Slyke (1972)].

Yao (1975) bol prvý, ktorý dal $O(m \log\log n)$ algoritmus pre najlacnejšiu kostru. V práci [Cheriton a Tarjan (1976)] možno nájsť viaceré jednoduchšie implementácie všeobecného algoritmu takejto zložitosti. V jednom z nich sa vyžaduje vyberať za $T$ vždy taký strom, ktorý má najmenší počet vrcholov.

## 5. C
## Najlacnejšia kostra s predpísaným stupňom jedného vrcholu

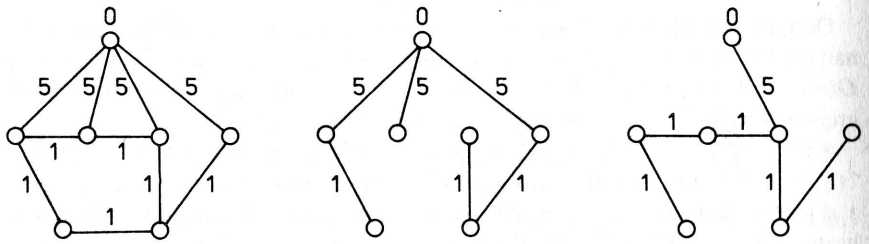
Glover a Klingman (1975) navrhli a riešili nasledujúce rozšírenie úlohy o najlacnejšej kostre.

Problém P($k$): Daný je graf $G$ s reálnymi cenami hrán, číslo $k$ a vyznačený jeden vrchol 0. Úlohou je nájsť najlacnejšiu kostru spomedzi kostier, v ktorých má vrchol 0 stupeň $k$.

Zrejme úlohy, kde žiadame, aby $\deg(0) \leq k$, resp. $\deg(0) \geq k$, sa redukujú na problém P($k$).

Takéto požiadavky na kostry vznikajú pri konštruovaní rôznych sietí. Napr. v počítačovej sieti by mal mať ústredný počítač dosť veľký „stupeň", lebo príliš málo liniek by nestačilo na prenos informácií.

Obr. 1 ilustruje problém P(3).




Obr. 1. Graf, riešenie problému P(3) a najlacnejšia kostra bez ohraničení.





Pre malé $k$ (nezávislé od $n$) možno úlohu riešiť aj prebraním $\binom{n}{k}$ možností pri vrchole 0 a použitím algoritmov pre najlacnejšiu kostru, avšak ak $k$ nie je pevné, ale napr. $n/2$, tak uvedený spôsob nedáva polynomiálny algoritmus. Postup, ktorý uvedieme neskôr, však taký algoritmus dáva v každom prípade. Na zdôvodnenie potrebujeme niekoľko tvrdení.

Nech $T$ je kostra grafu $G$ a hrana $e' \in G - T$. Potom jedinú cestu v $T$ spájajúcu kraje hrany $e'$ v $T$ označíme symbolom $T(e')$. Nasledujúcu vetu uvádzajú Čulík, Doležal a Fiedler (1967).

**Veta 1.** Nech $T$ a $T'$ sú kostry grafu $G$. Potom existuje taká bijekcia $\varphi$ množiny hrán $E(T) - E(T')$ na množinu $E(T') - E(T)$, že pre každé $e \in E(T) - E(T')$ platí: $T - e + \varphi(e)$ je kostra.

DÔKAZ. Dokážeme existenciu bijekcie $\gamma$: $E(T') \to E(T)$ takej, že pre každé $e' \in E(T') \cap E(T)$ je $\gamma(e') = e'$ a pre každé $e' \in E(T') - E(T)$ platí: $T + e' - \gamma(e')$ je kostra. Potom za $\varphi$ stačí zobrať reštrikciu zobrazenia $\gamma^{-1}$.

Ak existuje $e' \in E(T') \cap E(T)$, tak nech $e' = uv$ a utvorme kostru $\hat{T}$ z $T$ tak, že vynecháme hranu $e'$ a vrcholy $u$ a $v$ stotožníme. Úplne rovnako utvorme kostru $\hat{T}'$ z $T'$. Ak položíme $\gamma(e') = e'$ a použijeme indukčný predpoklad pre dvojicu kostier $\hat{T}$ a $\hat{T}'$, tak získame požadovanú bijekciu $\gamma$.

Takto zostáva prípad, keď $E(T') \cap E(T) = \emptyset$. Ak bijekcia $\gamma$ neexistuje, tak podľa Königovej–Hallovej vety 8.A.2 existuje $A \subseteq E(T')$ tak, že pre obal $S := T\langle \bigcup_{e' \in A} E(T(e')) \rangle$, platí:

$$(*) \qquad |A| > |E(S)|.$$

Budeme predpokladať, že množina $A$ je vybraná tak, že $|A|$ je minimálne.

Nech $V(A)$ označuje množinu všetkých krajov hrán z $A$. Zrejme $V(A) \subseteq V(S)$, $S \subseteq T$. Graf $T'\langle A \rangle \subseteq T'$ je les, a teda $|A| \leq |V(A)| - 1$. Ďalej tvrdíme, že $S$ je strom, lebo inak už pre niektorý jeho komponent možno napísať vzťah $(*)$, čo by bolo v spore s minimalitou $|A|$. Takto $|E(S)| \geq |V(A)| - 1$, čo spolu s predchádzajúcou nerovnosťou dáva: $|A| \leq |E(S)|$. Tento spor s $(*)$ dokazuje existenciu požadovanej bijekcie $\gamma$. ■

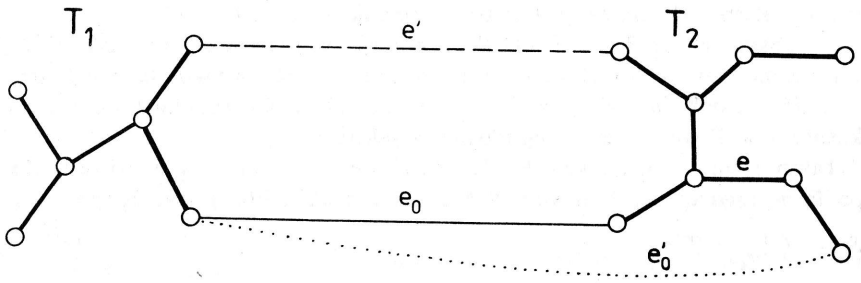
Množinu hrán grafu $G$ incidujúcich s vrcholom 0 budeme označovať symbolom $E_0$ a prvky množiny $E_0$ opatríme vždy indexom 0 (napr. $e_0$). Kostra $T$ sa nazýva *prípustná*, ak obsahuje práve $k$ hrán z množiny $E_0$.

**Veta 2.** Nech $G$ je graf a $T$ je jeho kostra. Ďalej nech hrany $e_0$, $e \in T$ a $e_0'$, $e' \in G - T$. Potom ak aspoň jeden z grafov $T + e' - e$ a $T + e_0' - e_0$ nie je kostrou, tak nasledujúce dve tvrdenia sú ekvivalentné.

(1) $T + e' - e_0$ a $T + e_0' - e$ sú kostry.

(2) $T + e' + e_0' - e - e_0$ je kostra.

DÔKAZ. (1) $\Rightarrow$ (2). Nech $T_1$ a $T_2$ sú stromy, z ktorých sa skladá les $T - e_0$. Podľa (1) je $T' := T + e' - e_0$ kostrou (pozri obr. 2, kde hrany v $T$ sú vyznačené plnými čiarami a hrany v $T_1$ a $T_2$ hrubými čiarami). Bez újmy na všeobecnosti môžeme predpokladať, že $e \in T_2$. Sú dve možnosti. (i) Jeden kraj hrany $e_0'$ patrí do $T_1$ a druhý do $T_2$ (obr. 2). Potom $T + e_0' - e_0$ je zrejme kostra a podľa predpokladu vety $T + e' - e$ už nie je kostra. Takto vidíme, že $e \notin T(e')$, ale podľa (1) $e \in T(e_0')$. Preto $T' + e_0' - e$ je kostra. (ii) Oba kraje hrany $e_0'$ patria do toho istého stromu (keďže $T + e_0' - e$ je kostra, tak patria do $T_2$). Teraz hneď vidieť, že $T' + e_0' - e$ je kostra.




Obr. 2. K dôkazu vety 2.

(2) $\Rightarrow$ (1). Nech $S = T + e' + e_0' - e - e_0$ je kostra. Potom podľa vety 1 existuje bijekcia $\varphi$: $\{e, e_0\} \to \{e', e_0'\}$ tak, že každá dvojica $(x, \varphi(x))$ dáva kostru $T + \varphi(x) - x$. Potom podľa predpokladu vety vzniká aspoň jedna z možností: $\varphi(e_0) \neq e_0'$ alebo $\varphi(e) \neq e'$. Teda $\varphi(e) = e_0'$ a $\varphi(e_0) = e'$. ■

Ak je daná kostra $T$ grafu $G$, tak hrany $e \in T$ a $e' \notin T$ nazývame *zameniteľné*, ak $T - e + e'$ je kostra grafu $G$: tiež hovoríme, že dvojica $(e, e')$ je zámenná pre $T$.

**Veta 3** (*o optimalite*). Prípustná kostra $T$ nie je optimálna pre problém P($k$) $\Leftrightarrow$ aspoň jedna z nasledujúcich troch podmienok je splnená:





(1) Existuje zámenná dvojica $(e, e')$ pre $T$ tak, že $c(e') < c(e)$, t. j. $T + e' - e$ je lacnejšia prípustná kostra.

(2) Existuje zámenná dvojica $(e_0, e_0')$ pre $T$ tak, že $c(e_0') < c(e_0)$, t. j. $T + e_0' - e_0$ je lacnejšia prípustná kostra.

3) Neplatí ani (1) ani (2) a existujú hrany $e$, $e_0 \in T$ a $e'$, $e_0' \notin T$ tak, že každá z dvojíc $(e_0, e')$, $(e, e_0')$ je zámenná pre $T$, a pritom $c(e') + c(e_0') < c(e) + c(e_0)$. Ekvivalentne: neplatí (1) ani (2) a $T + e' + e_0' - e - e_0$ je lacnejšia prípustná kostra.

DÔKAZ. ($\Leftarrow$) Ak platí (1) alebo 2, tak $T$ zrejme nie je optimálna kostra. Ak platí (3), tak neplatí (1) ani (2), a preto aspoň jeden z grafov $T + e' - e$, $T + e_0' - e_0$ nie je kostrou. Potom podľa vety 2 $T + e' + e_0' - e - e_0$ je kostra a zrejme prípustná aj lacnejšia. Obrátená časť ekvivalencie v (3) vyplýva tiež ihneď z vety 2. Takže $T$ nie je optimálna kostra.

($\Rightarrow$) Nech $T'$ je lacnejšia prípustná kostra. Potom podľa vety 1 existuje bijekcia $\varphi$: $T - T' \to T' - T$ tak, že každá dvojica $(x, \varphi(x))$ je zámenná pre $T$. Keďže vrchol 0 má rovnaký stupeň v $T$ aj v $T'$, tak tieto dvojice možno rozdeliť do troch skupín: dvojice tvaru $(e, e')$ (ako v (1)), dvojice tvaru $(e_0, e_0')$ (ako v (2)) a konečne skupina dvojíc tvaru $(e_0, e')$ a $(e, e_0')$ (ako v (3)), pričom z oboch tvarov je ich rovnaký počet. Ak žiadna z dvojíc 1. skupiny ani žiadna z dvojíc 2. skupiny, nedáva lacnejšiu kostru, tak neplatí ani (1) ani (2). Ukážeme, že platí (3). Totiž potom dvojice 3. skupiny dávajú $\Sigma [c(e) - c(e_0') + c(e_0) - c(e')] > 0$ (lebo $c(T) > c(T')$). Preto medzi nimi existuje zámenná dvojica $(e, e_0')$ a zámenná dvojica $(e_0, e')$ tak, že $c(e) + ce_0) > c(e_0') + c(e')$ (stačí zobrať zámennú dvojicu $(e, e_0')$, kde je $c(e) - c(e_0')$ maximálne a dvojicu $(e_0, e')$ maximalizujúcu $c(e_0) - c(e')$). ■

Veta 3 dáva zrejme polynomiálny algoritmus, ktorý postupne vylepšuje prípustné riešenie. Taký algoritmus by však bol dosť zložitý. Lepším sa ukazuje nasledujúci „duálny" prístup.

**Veta 4.** Nech $T$ je optimálna kostra v probléme P($k$). Potom:

(A) (*Zväčšovanie stupňa.*) Nech existujú hrany $e \in T$, $e_0' \notin T$ tak, že $T' := T + e_0' - e$ je kostra. Ak $c(e_0') - c(e)$ je minimálne, tak $T'$ je optimálna kostra pre problém P($k + 1$).

(B) (*Zmenšovanie stupňa.*) Nech existujú hrany $e_0 \in T$, $e' \notin T$ tak, že $T' := T + e' - e_0$ je kostra. Ak $c(e') - c(e_0)$ je minimálne, tak $T'$ je optimálna kostra pre problém P($k - 1$).

DÔKAZ. Pretože dôkazy častí (A) a (B) sú podobné, obmedzíme sa na dôkaz časti (A).

Ukážeme, že $T'$ nespĺňa žiadnu z podmienok (1), (2), (3) z vety 3, a tým dokážeme optimalitu kostry $T'$.

(1) Nech $e_1 \in T'$ a $e_1' \notin T'$ sú zámenné hrany pre $T'$ a $c(e_1) > c(e_1')$. Keďže $T$ je optimálna kostra, tak tieto hrany nie sú zámenné pre $T$. Takto získavame kostru $T'' := T' + e_1' - e_1 = T + e_0' + e_1' - e - e_1$. Podľa vety 1 potom každá z dvojíc hrán $(e_1, e_0')$ a $(e, e_1')$ je zámenná pre $T$ (lebo $(e_1, e_1')$ nemôže byť). Z predpokladu vety je $c(e_0') - c(e) \leq c(e_0') - c(e_1)$ a z optimality $T$ pre P($k$) máme $c(e_1') \geq c(e)$. Z týchto nerovností vidieť, že $c(e_1') \geq c(e) \geq c(e_1)$, a teda (1) neplatí.

(2) Nech $e_0 \in T'$ a $e_0'' \notin T'$ sú zámenné pre $T'$ a $c(e_0) > c(e_0'')$. Z kostry $T'$ získavame $T'' := T' + e_0'' - e_0 = T + e_0' + e_0'' - e - e_0$. Ak $e_0 = e_0'$, tak $T'' = T + e_0'' - e$ a z predpokladu vety je $c(e_0'') - c(e) \geq c(e_0') - c(e)$. Takže $c(e_0'') \geq c(e_0') = c(e_0)$, a teda (2) neplatí.

Preto nech $e_0 \neq e_0'$, a teda $e_0 \in T$. Keďže $e_0'' \notin T'$, tak aj $e_0'' \notin T$. Z vety 1 vidíme, že $(e_0, e_0')$ a $(e, e_0'')$ sú zámenné dvojice pre $T$ ($(e_0, e_0'')$ nie je taká, lebo $T$ je optimálna kostra). Z predpokladu vety je $c(e_0'') - c(e) \geq c(e_0') - c(e)$ a z optimality $T$ máme $c(e_0') \geq c(e_0)$, čo spolu dáva $c(e_0'') \geq c(e_0)$, a teda podmienka (2) ani teraz nie je splnená.

(3) Nech existujú hrany $e_0^*$, $e_1 \in T'$ a $e_0''$, $e_1' \notin T'$ tak, že $T'' := T' + e_0'' + e_1' - e_0^* - e_1$ je kostra, a pritom

(3a) $$c(e_0'') + c(e_1') - c(e_0^*) - c(e_1) < 0.$$

Ak $e_0^* = e_0'$, alebo $e = e_1'$, tak rovnaké úvahy ako v (1), resp. (2) vedú k sporu. Preto budeme predpokladať, že $e_0^* \neq e_0'$, aj $e \neq e_1'$. Odtiaľ vyplýva, že $e_1$, $e_0^* \in T$ a $e_0''$, $e_1' \notin T$. Takto budeme uvažovať kostru $T'' := T + e_0' + e_0'' + e_1' - e - e_1 - e_0^*$, kde pridávané hrany nepatria do $T$, vynechávané patria do $T$ a všetkých šesť hrán je navzájom rôznych. Podľa vety 1 možno z nich utvoriť tri dvojice zámenných hrán pre $T$. Teoreticky je 6 (=3!) možností:

(I) $(e, e_0'')$, $(e_1, e_0')$, $(e_0^*, e_1')$:
(II) $(e, e_0'')$, $(e_1, e_1')$, $(e_0^*, e_0')$:
(III) $(e, e_0')$, $(e_1, e_0'')$, $(e_0^*, e_1')$:
(IV) $(e, e_0')$, $(e_1, e_1')$, $(e_0^*, e_0'')$:
(V) $(e_1, e_0'')$, $(e, e_1')$, $(e_0^*, e_0')$:
(VI) $(e_1, e_0')$, $(e, e_1')$, $(e_0^*, e_0'')$.

Uvažujme prípad (I). Z predpokladu vety (o vzniku $T'$ z $T$) je

(3b) $$c(e_0'') - c(e) \geq c(e_0') - c(e).$$

Keďže $T + e_0' - e_1$ a $T + e_1' - e_0^*$ sú kostry (2. a 3. dvojica v (I)), tak podľa vety 2 dokážeme, že aj $\hat{T} := T + e_0' + e_1' - e_1 - e_0^*$ je kostra. K tomu stačí uvážiť, že





aspoň jeden z grafov $S_1 := T + e_1' - e_1$ a $S_2 := T + e_0' - e_0^*$ nie je kostra. To však je pravda, lebo kombinujúc (3a) a (3b) vidíme, že

(3c) $$c(e_1') - c(e_1) + c(e_0') - c(e_0^*) < 0.$$

Potom ak by $S_1$ aj $S_2$ boli kostry, tak aspoň jedna z nich by bola lacnejšia ako $T$. Takto $\hat{T}$ je prípustná kostra pre P($k$), ale podľa (3c) je lacnejšia ako optimálna kostra $T$, čo je spor. Tým je prípad (I) vylúčený.

Uvažujme prípad (II). Prvá dvojica je taká istá ako v (I) a dáva tú istú nerovnosť (3b). Kombinujúc (3a) a (3b) získame (3c), a preto aspoň jedna z kostier $T + e_1' - e_1$ a $T + e_0' - e_0^*$ (druhá a tretia dvojica v (II)) je lacnejšia ako $T$, čo je nemožné.

Ostatné prípady sa vylučujú podobne. Menovite (III) ako (I) a prípady (IV), (V) a (VI) ako (II). Sumárne potom je vylúčený prípad (3). Tým je optimalita kostry $T'$ pre problém P($k+1$) dokázaná.

Druhá časť vety (časť (B) – optimalita kostry $T'$ pre P($k-1$)) sa dokazuje analogicky a dôkaz prenechávame čitateľovi (5.Z.4). ■

Nasledujúca veta hovorí o tom, kedy možno aplikovať vetu 4.

**Veta 5.** Nech $k_1$ je počet blokov grafu $G$ obsahujúcich vrchol 0 a nech $k_2 = \deg_G(0)$. Ďalej nech $T$ je kostra grafu $G$ a $\deg_T(0) = k$. Potom $k_1 \leq k \leq k_2$ a naviac

(A) Existujú hrany $e \in T$ a $e_0' \notin T$ tak, že $T + e_0' - e$ je kostra $\Leftrightarrow k < k_2$.
(B) Existujú hrany $e_0 \in T$ a $e' \notin T$ tak, že $T + e' - e_0$ je kostra $\Leftrightarrow k_1 < k$.

DÔKAZ. Zrejme $k \leq k_2$. Naviac, pre každý blok $B$ má $B \cap T$ stupeň každého vrcholu aspoň 1, teda $k_1 \leq k$.

(A) ($\Rightarrow$) Pretože $T + e_0' - e$ má pri vrchole 0 stupeň $k+1$, tak zrejme $k < k_2$.

($\Leftarrow$) Nech hrana $e_0' \in G - T$, potom $T + e_0'$ obsahuje cyklus dĺžky aspoň 3, a ten obsahuje hranu $e$ neincidentnú s vrcholom 0. Takže $T + e_0' - e$ je kostra.

(B) ($\Rightarrow$) $T + e' - e_0$ má pri vrchole 0 stupeň $k-1$, tak podľa uvedeného $k_1 < k$.

($\Leftarrow$) Ak $k_1 < k$, tak pre nejaký blok $B$ obsahujúci vrchol 0 platí: $\deg_{B \cap T}(0) \geq 2$. Nech okolie $V_{B \cap T}(0) = \{v_1, v_2, \ldots, v_s\}$: potom $s \geq 2$. Takto $(B \cap T) - \{0\}$ má práve $s$ komponentov s vrcholovými množinami $V_1, V_2, \ldots, V_s$. Keďže vrchol 0 nie je artikuláciou v $B$, tak existuje hrana $e'$ spájajúca nejaký vrchol z $V_1$ s nejakým vrcholom z $V_i$, kde $i \neq 1$. Potom $T + e'$ obsahuje cyklus obsahujúci hrany $0v_1$ a $0v_i$. Takto stačí položiť $e_0 = 0v_i$. ■

Na základe viet 4 a 5 možno postaviť takýto algoritmus pre riešenie problému P($k$): (1) Ak $k < k_1$ alebo $k > k_2$, tak P($k$) nemá riešenie. (2) Ak $k_1 \leq k \leq k_2$, tak

nájdeme najlacnejšiu kostru $T$ (bez ohraničenia na stupeň vrcholu 0) podľa časti 5.B: nech $\deg_T(0)=s$. (2.1) Ak $s<k$, tak aplikujeme vetu 4(A) $(k-s)$-krát a získame optimálnu kostru pre P($k$). (2.2) Ak $s>k$, tak zasa $(s-k)$-násobné použitie vety 4(B) nám dá optimum pre P($k$). Podrobnejšiu implementáciu tohto postupu možno nájsť v [Glover, Klingman (1975)]. ●

Na záver poznamenajme, že možno postaviť všeobecnejšiu úlohu: V súvislom grafe $G$ je daných $t$ špeciálnych vrcholov $v_1, v_2, \ldots, v_t$: okrem toho je daných $t$ čísel $k_1, k_2, \ldots, k_t$. Úlohou je nájsť najlacnejšiu kostru grafu $G$ spomedzi kostier, v ktorých pre každé $i=1,2,\ldots,t$ má vrchol $v_i$ stupeň $k_i$. Práve sme opísali polynomiálny algoritmus pre $t=1$. Avšak už pre $t=2$ zostáva problém otvorený. Žiaľ, ak $t$ nie je pevné, tak problém je NP-ťažký, dokonca aj vtedy, ak žiadame len prípustné riešenie (5.Z.5).

## 5.D
## Kostra s minimálnym pomerom dvoch ohodnotení

Tu sa budeme zaoberať s nasledujúcim problémom, ktorý navrhol a vyriešil Chandrasekaran (1977).

Problém MP: Daný je graf $G$, kde každá hrana $e\in E(G)$ má dve reálne ohodnotenia $a_e$, $b_e$. Úlohou je nájsť kostru $T$ grafu $G$ minimalizujúcu pomer

$$(*)\qquad \sum_{e\in T} a_e \Big/ \sum_{e\in T} b_e.$$

Ako príklad na aplikáciu uvažujeme nasledujúcu úlohu. Pri konštrukcii elektrického zariadenia treba navrhnúť prepojenie $n$ daných svoriek do súvislej siete, pričom iné svorky sa už nepoužijú. Pritom za prepojenie dostaneme $p$ korún, vodič spájajúci svorky $i$ a $j$ stojí $c_{ij}$ korún a inštalácia trvá $t_{ij}$ časových jednotiek. Predpokladajme, že takýchto zariadení máme vyrábať veľmi mnoho, a preto sa kladie úloha: maximalizovať priemerný zisk za časovú jednotku.

Je zrejmé, že optimálne riešenie (prepojenie) $T$ musí byť kostrou a ak pre $e=ij$ položíme $a_e := c_e - p/(n-1)$, $b_e := t_e$, tak máme uvedený problém MP.

### TRANSFORMÁCIA NA PARAMETRICKÚ ÚLOHU O NAJLACNEJŠEJ KOSTRE

Jednoduché modifikácie algoritmov pre najlacnejšiu kostru nevedú k riešeniu problému MP. Ukážeme, že možno použiť známy trik pre transformáciu zlomko-





vej účelovej funkcie na parametrickú. K tomu budeme predpokladať, že pre každú kostru $T$ je $\sum_{e\in T} b_e>0$. K problému MP a k danému reálnemu číslu $k$ priraďme:

Problém N($k$): Treba nájsť najlacnejšiu kostru grafu $G$, kde cena každej hrany $e$ je $c_e(k) := a_e - kb_e$.

**Veta 1.** Nech $k^*$ je optimálna hodnota účelovej funkcie $(*)$ v probléme MP a nech $T^*(k)$ je riešením problému N($k$). Potom platí:

$$\sum_{e\in T^*(k)} c_e(k) \begin{cases} >0 \Leftrightarrow k^*>k, \\ <0 \Leftrightarrow k^*<k, \\ =0 \Leftrightarrow k^*=k. \end{cases}$$

DÔKAZ. Môžeme písať ekvivalencie:

$$\sum_{e\in T^*(k)} c_e(k)>0$$

$$\Leftrightarrow \sum_{e\in T} c_e(k)>0, \qquad \text{pre každé } T$$

$$\Leftrightarrow \sum_{e\in T} a_e \Big/ \sum_{e\in T} b_e>k, \qquad \text{pre každé } T$$

$$\Leftrightarrow k^*>k.$$

Tým je prvá ekvivalencia dokázaná. Dokážeme druhú:

$$k^*<k$$

$$\Leftrightarrow \text{existuje } T^*: \quad \sum_{e\in T^*} a_e \Big/ \sum_{e\in T^*} b_e<k$$

$$\Leftrightarrow \text{existuje } T^*: \quad \sum_{e\in T^*} c_e(k)<0$$

$$\Leftrightarrow \sum_{e\in T^*(k)} c_e(k)<0.$$

Tretia ekvivalencia vyplýva bezprostredne z prvých dvoch. ■

Bezprostredne máme:

**Dôsledok 1(a).** Nejaká kostra $T$ je optimálna pre problém MP $\Leftrightarrow$ $T$ je optimálna pre problém N($k^*$).

V ďalšom ukážeme, že stačí riešiť iba konečne mnoho, ba dokonca iba

polynomiálne mnoho problémov N($k$), aby sme našli $k^*$, a teda podľa dôsledku 1(a) aj riešenie problému MP.

Nech $e, f \in E(G)$. Potom funkcie $a_e - kb_e$ a $a_f - kb_f$ premennej $k$ sú alebo identické, alebo jedna je všade väčšia ako druhá, alebo majú jediný priesečník pri $k(e, f) := (a_e - a_f)/(b_e - b_f)$. Takto množina $K := \{k(e, f) \mid e, f \in E(G)\}$ má najnajvýš $m^2$ takýchto hodnôt $k$ a ľahko ich možno pomocou $O(m^2 \log m)$ operácií usporiadať do postupnosti:

$$-\infty < k_1 < k_2 < \ldots < k_r < \infty.$$

**Veta 2.** Nech $I$ je jeden z otvorených intervalov $(-\infty, k_1)$, $(k_i, k_{i+1})$ pre $1 \leq i < r$, $(k_r, \infty)$ a nech $\hat{k} \in I$. Potom každé riešenie $T^*(\hat{k})$ problému N($\hat{k}$) je riešením problému N($k$) pre každé $k$ z uzáveru intervalu $I$.

DÔKAZ. Podľa vety 5.B.2 ak $T^*(\hat{k})$ nie je optimálnym riešením problému N($k$), tak existujú zámenné hrany $e \in T^*(\hat{k})$ a $f \notin T^*(\hat{k})$ tak, že $c_f(k) < c_e(k)$. Keďže $c_f(\hat{k}) \geq c_e(\hat{k})$, tak alebo $k(e, f) \in [\hat{k}, k) \subset I$ alebo $k(e, f) \in (k, \hat{k}] \subset I$. Pretože $I$ neobsahuje žiadne $k_i$, máme spor. ■

Z nášho predpokladu, že pre každú kostru $T$ je $\Sigma\, b_e > 0$, vyplýva:

$$L_T(k) := \sum_{e \in T} (a_e - kb_e) > 0, \qquad \text{ak } k \to -\infty:$$

$$L_T(k) < 0, \qquad \text{ak } k \to +\infty.$$

Keďže $L_T(k)$ je spojitá funkcia argumentu $k$, tak pre riešenie problému MP možno dať tento algoritmus: Po nájdení intervalov $I_0 := (-\infty, k_1)$, $I_1 := (k_1, k_2)$. ..., $I_r := (k_r, \infty)$ zvolíme stredný index $j$ (t. j., $j := \lfloor r/2 \rfloor$) a nejaké $k \in I_j$. Potom nájdeme najlacnejšiu kostru $T^*(k)$ a vypočítame $L_j := L_{T^*(k)}(k_j)$ a $L_{j+1} := L_{T^*(k)}(k_{j+1})$. Ak $L_j L_{j+1} \leq 0$, tak $T^*(k)$ je riešenie problému MP. Ak $L_j > 0$ aj $L_{j+1} > 0$, tak zvolíme stredný index v množine $\{j+1, \ldots, r\}$ a ak $L_j < 0$ aj $L_{j+1} < 0$, tak v $\{0, \ldots, j\}$ a postup opakujeme. Takto po vyriešení $O(\log m)$ problémov najlacnejšej kostry získame požadované riešenie. ● Zložitosť celého postupu je $O(m^2 \log m)$ a je určená zložitosťou triedenia čísel $k_i$.

## ZLOŽITOSŤ PROBLÉMU PRI ĽUBOVOĽNÝCH OHODNOTENIACH

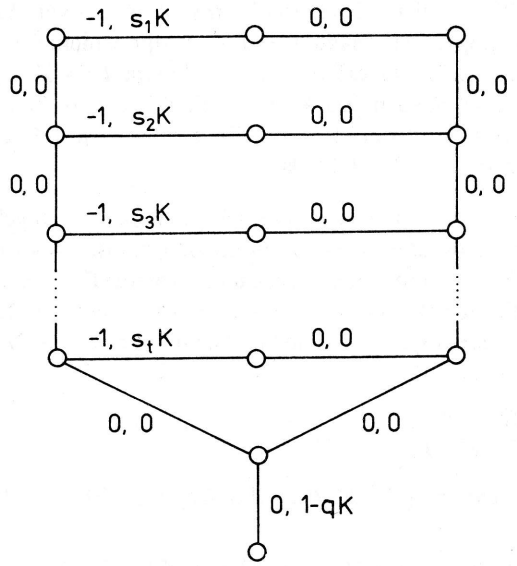
Predchádzajúci algoritmus sme odvodili za predpokladu, že pre každú kostru je $\Sigma\, b_e > 0$. Ak je $\Sigma\, b_e < 0$, tak stačí zmeniť každé $a_e$ na $-a_e$ a máme prvý prípad. Žiaľ, vo všeobecnosti platí:





**Veta 3.** Problém MP je NP-ťažký dokonca pre planárne grafy s maximálnym stupňom 3.

DÔKAZ. Uvedieme polynomiálnu transformáciu problému rozdelenia multimnožiny prirodzených čísel $s_1, s_2, \ldots, s_t$ na dve triedy $\{s_i\}_{i \in I}$ a $\{s_i\}_{i \in \bar{I}}$, každú so súčtom $q = \Sigma\, s_i/2$. Nech $K \geq 2$ a nech $G$ je graf z obr. 1, kde každá hrana $e$ má vyznačené ohodnotenia $a_e$, $b_e$. Ľahko sa ukáže, že problém rozdelenia má riešenie práve vtedy, ak optimum v probléme MP nepresahuje $-t/2$. ■




Obr. 1. K dôkazu vety 3.

## 5. E
## Generovanie zoznamu $K$ najlacnejších kostier

Niekedy potrebujeme nájsť najlacnejšiu kostru spomedzi kostier spĺňajúcich dosť komplikované ohraničenia (napr. stupne vrcholov nesmú prekročiť dané číslo a pod.). Vtedy sa často uchýlime k „preberaniu" kostier. Všetkých kostier v grafe môže byť aj nepolynomiálne mnoho, a preto je vhodné generovať a overovať kostry v smere od najlacnejšej k najdrahšej. To dáva nádej, že nebude treba prebrať všetky kostry, ale len nejakých $K$ najlacnejších.

## PRINCÍP ALGORITMU

Tu načrtneme myšlienku jedného prístupu k tejto úlohe podľa práce [Gabow (1977)]. Základom je nasledujúca veta:

**Veta 1.** Nech $G$ je graf a $IN$, $OUT \subseteq E(G)$ sú dané množiny hrán. Ďalej nech $S = \{T \mid T$ je kostra grafu $G$ a $IN \subseteq E(T) \subseteq E(G) - OUT\}$ a nech $T_0$ je najlacnejšia kostra v $S$. Potom druhú najlacnejšiu kostru v $S$ stačí hľadať medzi susedmi (susednými kostrami) kostry $T_0$.

DÔKAZ. Nech $T'$ je druhá najlacnejšia kostra v $S$. Podľa vety 5.C.1 možno hrany množiny $T_0 - T'$ popáriť s hranami $T' - T_0$ do dvojíc zámenných hrán pre $T_0$. Nech sú to $(e_1, f_1), \ldots, (e_r, f_r)$. Pre každé $j = 1, \ldots, r$ kostra $T_j := T_0 - e_j + f_j$ zrejme patrí do $S$. Keďže $T_0$ je najlacnejšia kostra v $S$, tak $c(f_j) - c(e_j) \geq 0$. Potom však $c(T_j) \leq c(T_0) + \Sigma [c(f_j) - c(e_j)] = c(T')$, a teda $T_j$ je tiež druhá najlacnejšia kostra v $S$ a pritom susedná s $T_0$. ■

Myšlienka algoritmu je takáto: Nech $S_1^1$ je množina všetkých kostier grafu $G$. Nájdeme (napr. Kruskalovým algoritmom) najlacnejšiu kostru $T_1$ v $S_1^1$. Potom nech $IN_1^1 := OUT_1^1 := \emptyset$ a druhú najlacnejšiu kostru $T_2$ hľadáme medzi susedmi kostry $T_1$ (podľa vety 1, kde $IN := OUT := \emptyset$). Nech $T_2 = T_1 - e_1^1 + f_1^1$. Potom $S_1^1$ rozdelíme na dve disjunktné množiny (sleduj obr. 1, kde vždy množina $S_i^j$ má minimálny prvok $T_i$)

$S_1^2 := \{T \mid T \in S_1^1,\ e_1^1 \in T\}$,
$S_2^2 := \{T \mid T \in S_1^1,\ e_1^1 \notin T\}$.

Ak položíme $IN_1^2 := \{e_1^1\}$, $OUT_1^2 := \emptyset$ a $IN_2^2 := \emptyset$, $OUT_2^2 := \{e_1^1\}$, tak môžeme písať

$S_1^2 = \{T \mid T$ je kostrou v $G$ a $IN_1^2 \subseteq E(T) \subseteq E(G) - OUT_1^2\}$,

$S_2^2 = \{T \mid T$ je kostrou v $G$ a $IN_2^2 \subseteq E(T) \subseteq E(G) - OUT_2^2\}$.

Vidíme, že $T_1 \in S_1^2$, $T_2 \in S_2^2$. Takto na základe vety 1 stačí hľadať druhú najlacnejšiu kostru v $S_1^2$ medzi susedmi kostry $T_1$ a tiež druhú najlacnejšiu kostru v $S_2^2$ medzi susedmi kostry $T_2$. Lacnejšia z nich je zrejme tretia najlacnejšia kostra $T_3$ grafu $G$. Ak napr. $T_3 \in S_1^2$ (ako na obr. 1) a $T_3 = T_1 - e_1^2 + f_1^2$, tak potom rozdelíme množinu $S_1^2$ na dve: množinu

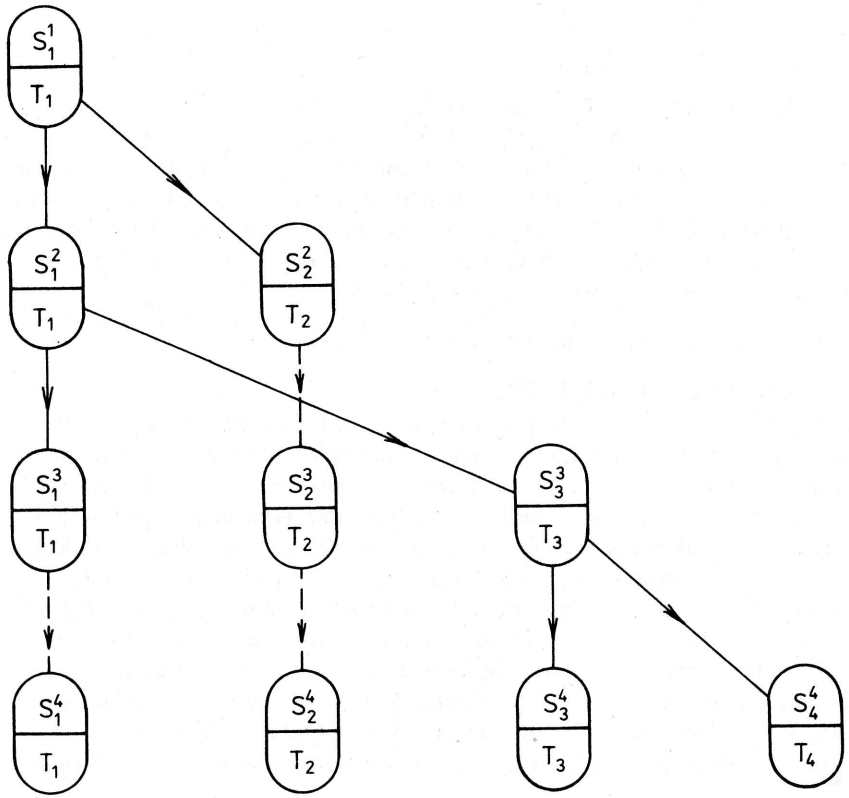
$S_1^3 := \{T \mid T \in S_1^2$ a $e_1^2 \in T\}$ obsahujúcu $T_1$ a množinu

$S_3^3 := \{T \mid T \in S_1^2$ a $e_1^2 \notin T\}$ obsahujúcu $T_3$.

Okrem toho kladieme $S_2^3 := S_2^2$ (túto množinu sme „zdedili“).








Obr. 1. Rozkladanie množiny kostier.

Nech máme vo všeobecnosti množinu všetkých kostier grafu $G$ takto rozdelenú na množiny $S_1^j$, $S_2^j$, ..., $S_j^j$, kde

(*) $S_i^j = \{T \mid T$ je kostrou grafu $G$ a $IN_i^j \subseteq E(T) \subseteq E(G) - OUT_i^j\}$.

$T_i$ je najlacnejšou kostrou v $S_i^j$ a $i$-tou najlacnejšou kostrou grafu $G$ ($i = 1, \ldots, j$). Potom v každej množine $S_i^j$ môžeme nájsť druhú najlacnejšiu kostru (podľa vety 1 stačí hľadať medzi susedmi kostry $T_i$) a najlacnejšiu z nich označiť $T_{j+1}$ a prehlásiť za $(j+1)$-ú najlacnejšiu kostru grafu $G$. Nech $T_{j+1} \in S_k^j$ a $T_{j+1} = T_k - e_k^j + f_k^j$. Potom množinu $S_k^j$ rozdelíme na dve:

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

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

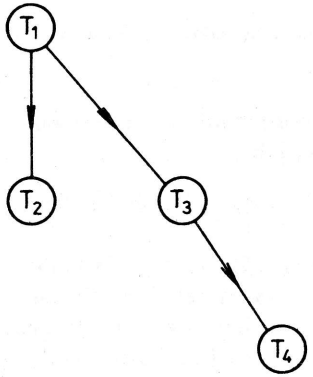
Ak definujeme $IN_k^{j+1} := IN_k^j \cup \{e_k^j\}$, $OUT_k^{j+1} := OUT_k^j$
a $IN_{j+1}^{j+1} := IN_k^j$, $OUT_{j+1}^{j+1} := OUT_k^j \cup \{e_k^j\}$,
tak $S_k^{j+1}$ a $S_{j+1}^{j+1}$ budú mať tvar (*). Tiež vidíme, že $T_k \in S_k^{j+1}$ a $T_{j+1} \in S_{j+1}^{j+1}$. Množiny $S_i^{j+1}$ $(i = 1, \ldots, j: j \neq k)$ 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 $(g, h)$.

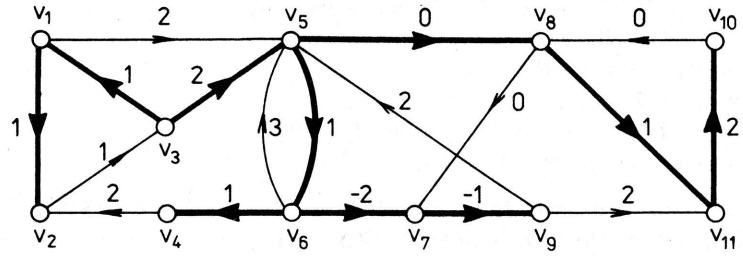
Gabow (1977) dáva podrobnú implementáciu celého algoritmu a ukazuje, že vystačíme s časom $O(Km \log m)$ (ba ešte trochu lepším). Tiež uvádza modifikáciu poskytujúcu všetky kostry grafu (nech je ich $N$) od najlacnejšej po najdrahšiu v čase $O(Nm)$ a v priestore $O(N+m)$.

## 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 $|Q| \leq n$.



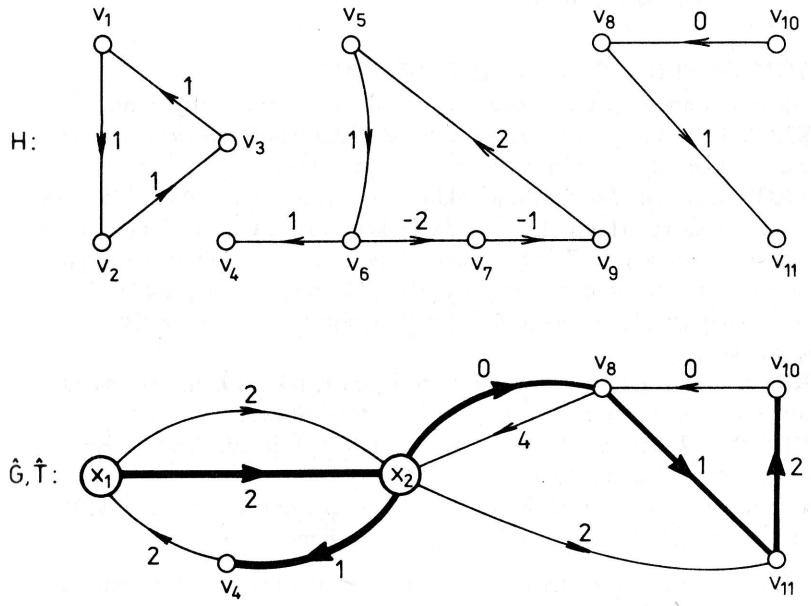


1.2: Ak $|Q| < n-1$, tak STOP: digraf $G$ nemá zdrojovú kostru.

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 $\deg_H^-(u) \leq 1$ 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.

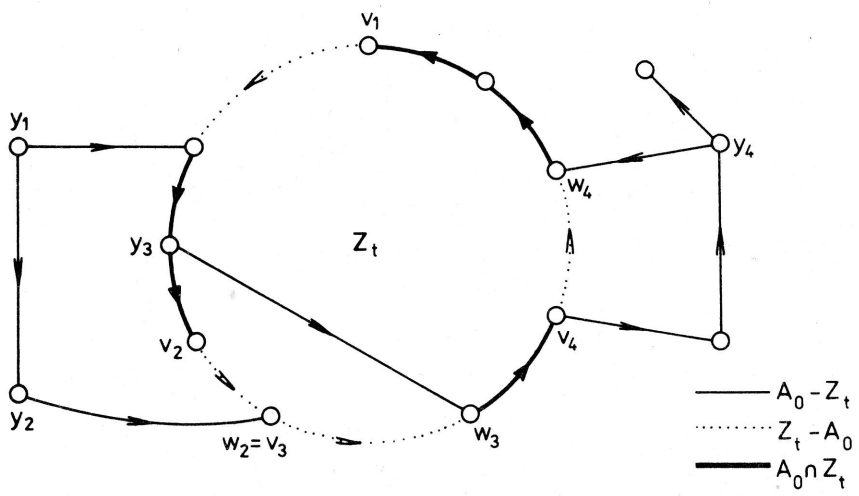
(KROK 3) Je zrejmé, že utvorený digraf $T$ z $\hat{T}$ je zdrojovou kostrou v $G$. Zostáva dokázať optimalitu kostry $T$.

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 $Z_t$.

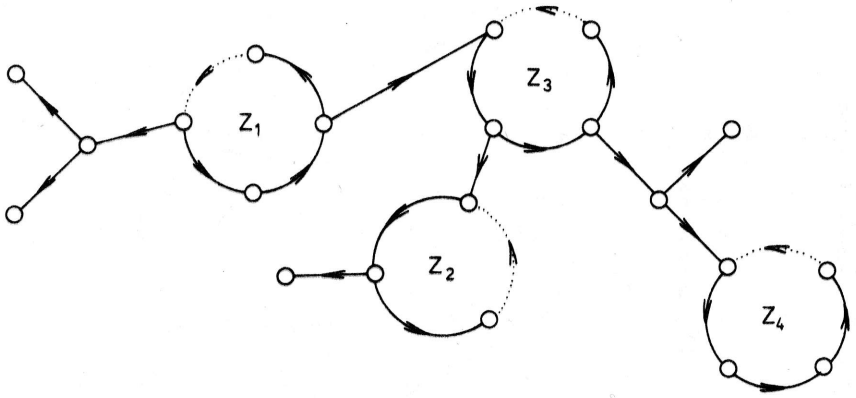
Nech $v_1w_1, \ldots, v_rw_r$ sú všetky šípy cyklu $Z_t$ v smere jeho obiehania, ktoré nepatria do $A_0$ ($w_i = v_{i+1}$ je možné). Obr. 3 ilustruje takúto situáciu. Pre spor predpokladajme, že $r \geq 2$. Uvažujme šípy $v_iw_i \in Z_t$. Máme dve možnosti, ktoré postupne rozoberieme.

(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. ■

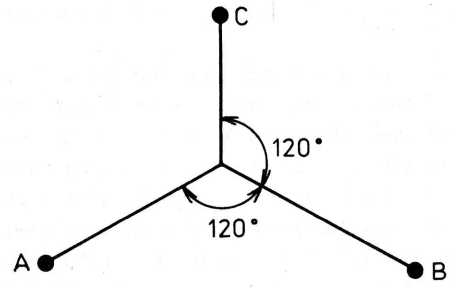
Na prechod od $G$ ku $\hat{G}$ nám stačí $O(m+n)$ operácií a tak isto je to pri prechode od $\hat{G}$ ku $G$. Takýchto prechodov môže byť nanajvýš $n-1$ ($n(\hat{G}) \leq n(G) - 1$), a teda celý algoritmus má zložitosť nanajvýš $O(mn + n^2)$. Tarjan (1977) dáva implementácie pracujúce v časoch $O(m \log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



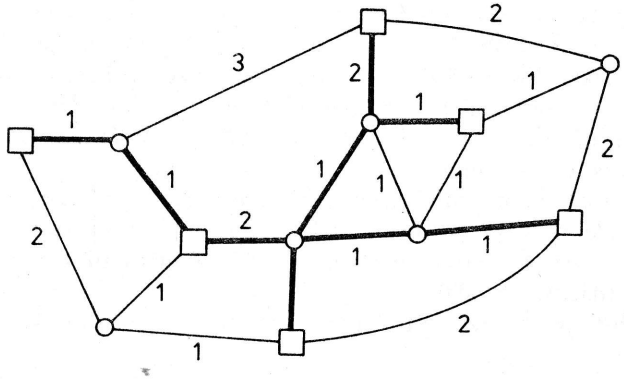

Obr. 1. Steinerov problém v rovine a jeho riešenie.

budeme venovať, sformulovali Hakimi (1971) a nezávisle Dreyfus a Wagner (1971): Daný je súvislý graf $G$, kde každá hrana $e \in E(G)$ má kladné ohodnotenie (cenu) $c(e)$: pre danú množinu $B \subseteq V(G)$, tzv. *základných vrcholov,* treba nájsť najlacnejší strom $T \subseteq G$ obsahujúci $B$.

Ľubovoľný súvislý podgraf $H \subseteq G$ obsahujúci $B$ nazývame *steinerovský podgraf*: špeciálne, ak $H$ je strom, tak sa nazýva *steinerovský strom.* Je zrejmé, že v Steinerovom probléme môžeme namiesto stromu $S$ žiadať najlacnejší steinerovský podgraf, lebo vždy bude riešením strom.

Tento problém je problémom najlacnejšej kostry, ak $B = V(G)$ a problémom najkratšej cesty, ak $|B| = 2$. Zrejme tento problém lepšie vystihuje úlohy navrhovania optimálnych spojovacích sietí ako problém najlacnejšej kostry (pozri 5.B). Na obr. 2 je príklad, kde sú štvorčekmi vyznačené základné vrcholy a najlacnejší steinerovský strom je vyznačený hrubými čiarami.

Steinerovský strom môže okrem základných vrcholov obsahovať aj iné, tzv. *steinerovské vrcholy* (porovnaj obr. 2, kde sú štyri také vrcholy).




Obr. 2. Najlacnejší steinerovský strom (hrubé čiary).

Je zrejmé, že ak v nejakom cykle $Z$ grafu pre nejakú hranu $ij = e \in Z$ platí: $c(e) > c(Z - e)$, tak hranu $e$ nikdy nepoužijeme v optimálnom steinerovskom strome, a teda ju môžeme vynechať. Iná možnosť je, že vypočítame vzdialenosť $d_G(i, j)$, pričom dĺžky hrán kladieme rovné ich cenám a potom definujeme novú cenu $c'(e) := d_G(i, j)$. Takto môžeme aj doplniť graf na kompletný (cena novej hrany bude rovná vzdialenosti jej krajov). Týmto možno zabezpečiť, že ohodnotenia tvoria metriku. Potom každému steinerovskému stromu v takomto kompletnom grafe možno ľahko priradiť steinerovský strom rovnakej ceny (pre tie isté základné vrcholy) v pôvodnom grafe.





Nasledujúci výsledok sa objavuje už v práci [Jarník, Kössler (1934)] (pre Steinerov problém v euklidovskom priestore).

**Veta 1.** Nech ohodnotenia v kompletnom grafe $K_n$ tvoria metriku a nech $p$ je počet základných vrcholov. Potom pre počet $s$ steinerovských vrcholov v najlacnejšom steinerovskom strome $T$ platí: $s \leq p - 2$.

DÔKAZ. Strom $T$ má $p + s$ vrcholov, a teda $p + s - 1$ hrán. Každý základný vrchol má stupeň aspoň 1 a každý steinerovský vrchol aspoň 3 (z predpokladu metrického ohodnotenia), takže súčet stupňov v strome $T$ je aspoň $p + 3s$. Takto $2(p + s - 1) \geq p + 3s$ odkiaľ máme požadovanú nerovnosť. ■

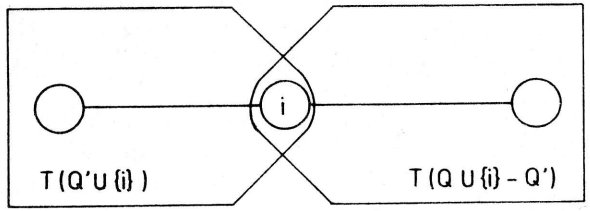
Poznamenajme, že veta 1 naznačuje takýto postup pre nájdenie najlacnejšieho steinerovského stromu: Pre každé $j = 0, 1, \ldots, p - 2$ a každú podmnožinu $M \subseteq V(G) - B$ mohutnosti $j$ nájsť v indukovanom podgrafe $G(B \cup M)$ najlacnejšiu kostru. Potom najlacnejšia zo všetkých takýchto kostier bude najlacnejším steinerovským stromom. Takýto postup však možno odporučiť iba vtedy, ak $|V(G) - B|$ je malé číslo.

## DREYFUSOVA A WAGNEROVA METÓDA

Dreyfus a Wagner (1971) predložili trochu nádejnejší (hoci tiež nepolynomiálny) algoritmus, ktorý možno odporučiť pre malé hodnoty $p := |B|$. Ak $Q \subseteq B$ a $i \in V(G)$, tak $c(Q \cup \{i\})$ bude označovať cenu najlacnejšieho steinerovského stromu $T(Q \cup \{i\})$ pre (množinu základných vrcholov) $Q \cup \{i\}$ a $c_i\{Q\}$ cenu najlacnejšieho stromu $T_i(Q)$ spomedzi steinerovských stromov pre $Q \cup \{i\}$, ktoré majú vrchol $i$ nekoncový.

Nech $i \notin Q$. Optimálny steinerovský strom pre $Q \cup \{i\}$ s nekoncovým vrcholom $i$ možno rozložiť na dva optimálne steinerovské stromy: jeden pre $Q' \cup \{i\}$ a druhý pre $Q \cup \{i\} - Q'$, kde $Q'$ je vhodná množina s $\emptyset \subset Q' \subset Q$ (obr. 3), takže

$$(*) \qquad c_i(Q) = \min_{\emptyset \subset Q' \subset Q} \{c(Q' \cup \{i\}) + c(Q \cup \{i\} - Q')\}.$$




Obr. 3. Rozklad steinerovského stromu $T_i(Q)$.

Pre získanie vzťahu pre $c(Q\cup\{i\})$ predpokladajme, že $G$ je kompletný graf s metrickým ohodnotením. Potom pre optimálny steinerovský strom $T(Q\cup\{i\})$ máme tri možnosti:

(1) Vrchol $i$ je nekoncový. Potom $c(Q\cup\{i\})=c_i(Q)$.

(2) Vrchol $i$ je koncový a jeho jediný sused $j\notin Q$. Potom vrchol $j$ je steinerovský vrchol v $T(Q\cup\{i\})$, takže (z predpokladu metrického ohodnotenia) má stupeň aspoň 3. Takto $T(Q\cup\{i\})$ možno rozložiť na hranu $ij$ a strom $T(Q\cup\{j\})$, v ktorom $j$ má stupeň aspoň 2. Preto $c(Q\cup\{i\}) = c_{ij}+c_j(Q)$.

(3) Vrchol $i$ je koncový a jeho sused (v strome) $j\in Q$. Potom $c(Q\cup\{i\})=c_{ij}+c(Q)$.

Sumarizujúc tieto tri prípady, môžeme písať:

$$(**)\qquad c(Q\cup\{i\})=\min\begin{cases}\min\limits_{j\notin Q}\{c_{ij}+c_j(Q)\},\\ \min\limits_{j\in Q}\{c_{ij}+c(Q)\}.\end{cases}$$

Poznamenajme, že predchádzajúci vzorec zahŕňa aj prípad (1) ($c_{ii}=0$, $j=i\notin Q$).

Ak položíme $c(U)=0$, pre $|U|=1$, tak rekurentné vzťahy (*) a (**) umožňujú vypočítať postupne až $c(B)$ a samozrejme súčasne nájsť aj optimálny strom. Avšak exponenciálny počet operácií nedovoľuje realizovať výpočet pre väčšie $|B|$. ●

## ZLOŽITOSŤ PROBLÉMU

Steinerov problém v grafoch patrí medzi ťažké problémy, ako dokázal Karp (1972).

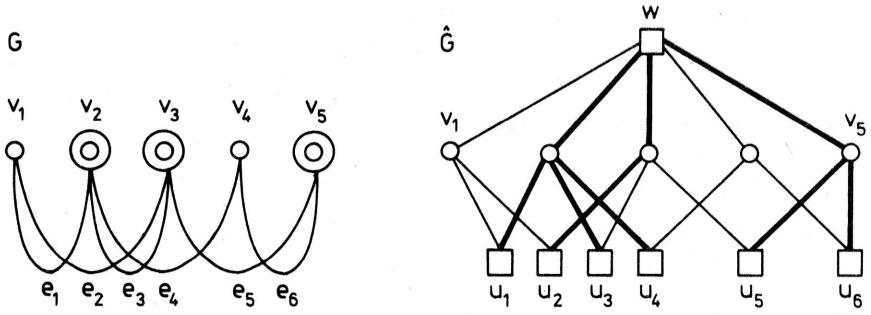
**Veta 2.** Steinerov problém v grafoch je NP-ťažký dokonca pre bipartitné neohodnotené grafy (všetky $c(e)=1$).

DÔKAZ. Uvedieme polynomiálnu transformáciu problému vrcholového pokrytia hrán grafu na náš problém. Najprv do každej hrany $e_j$ daného grafu $G$ vsunieme nový vrchol $u_j$ ($j=1, \ldots, m$). Potom pridáme nový vrchol $w$, ktorý spojíme s každým pôvodným vrcholom $v_i\in V(G)$. Vzniknutý graf $\hat G$ je zrejme bipartitný. Ďalej nech $\hat B=\{w, u_1, \ldots, u_m\}$ je množina základných vrcholov v $\hat G$ (pozri obr. 4). Tvrdíme, že hrany grafu $G$ možno pokryť s $k$ vrcholmi práve vtedy, ak v $\hat G$ existuje steinerovský strom ceny $m+k$.

K pokrývajúcim vrcholom zostrojíme steinerovský strom ako na obr. 4. To možno aj obrátiť, ak každý vrchol $u_j$ má v strome stupeň 1. To však možno zariadiť: ak obe hrany $v_pu_j$, $u_jv_q$ patria do stromu, tak jednu z nich vynecháme a pridáme alebo hranu $wv_p$ alebo $wv_q$ tak, aby sme získali strom. ■








Obr. 4. K dôkazu vety 1.

V 5.Z.15 čitateľ uvidí, že problém zostáva NP-ťažký dokonca pre planárne grafy. Tam sa v skutočnosti ukáže NP-obťažnosť rektilineárneho problému [Garey, Johnson (1977)]. Dôkaz pre problém v rovine pozri v [Garey, Graham, Johnson (1977)].

## APROXIMÁCIA POMOCOU KOSTRY

Hoci problém najlacnejšieho steinerovského stromu je NP-ťažký, predsa možno dať hrubú aproximáciu optima v polynomiálnom čase.

K danému grafu $G$ s množinou základných vrcholov $B$ priraďme kompletný graf $K(B)$ s vrcholovou množinou $B$ a každej jeho hrane $uv$ priraďme cenu $c_{K(B)}(uv) := d_G(u, v)$, pričom pri výpočte vzdialenosti v $G$, po dĺžkami hrán rozumieme ich ceny. Nech $P_{uv}$ je nejaká $u$—$v$ cesta v $G$ dĺžky $d_G(u, v)$. Nech $T$ je nejaká najlacnejšia kostra grafu $K(B)$. Potom, ak nahradíme každú hranu $uv\in T$ cestou $P_{uv}$, získame nejaký steinerovský podgraf $H\subseteq G$ pre $B$, lebo $T$ je súvislý graf a obsahuje $B$. Konečne, z $H$ vyberieme steinerovský strom $T'$ pre $B$ (napr. zoberieme ľubovoľnú kostru $T'$ grafu $H$).

Nech $S$ je najlacnejší steinerovský strom pre $B$. Potom zrejme $c(S)\leq c(T')\leq c(H)\leq c(T)$. Hoci niekedy môže byť $c(T')<c(T)$, v najhoršom prípade máme len rovnosť. Tu vzniká prirodzená otázka: aký môže byť pomer $c(T)$ a $c(S)$ (pre netriviálny prípad, keď $|B|>1$). Presnejšie, nech $\varrho_{\text{graph}}$ je minimálne číslo také, že pre všetky príklady Steinerovho problému platí: $c(T)/c(S)\leq\varrho_{\text{graph}}$. Prv ako dáme na otázku odpoveď, poznamenajme, že uvedený spôsob aproximácie pomocou kostry bol študovaný najprv pre Steinerov problém v rovine, resp. v $n$-rozmernom euklidovskom priestore $R^n$ a tiež pre rektilineárny problém. Tam tiež má zmysel nájsť korešpondujúce čísla $\varrho_n$ a $\varrho_{\text{rect}}$. Graham a Hwang (1976) ukázali, že

$$\varrho_n\leq\sqrt{3}=1{,}73\ldots$$

a z príkladov Chunga a Gilberta (1976) máme

$$\varrho_n \geq (4-\sqrt{2})/\sqrt{3} = 1{,}49\ldots, \text{ ak } n \to \infty.$$

Ďalej Chung a Hwang (1978) dokázali, že

$$1{,}16\ldots = 2/\sqrt{3} \leq \varrho_2 \leq 3/(2\sqrt{3}+2-\sqrt{7+2\sqrt{3}}) = 1{,}34\ldots$$

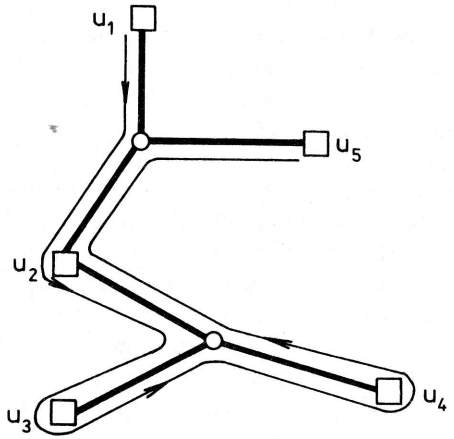
Konečne Hwang (1976) zistil, že

$$\varrho_{\text{rect}} = 3/2.$$

Žiaľ, pre grafy je toto číslo väčšie [Plesník (1981)]:

**Veta 3.** $\varrho_{\text{graph}} = 2$.

DÔKAZ. Najprv dokážeme, že $\varrho_{\text{graph}} \leq 2$. Zrejme v najlacnejšom steinerovskom strome $S$ je každý koncový vrchol základný: nech $u_1$ je jeden z nich. Nech $W$ je sled v $S$ začínajúci v $u_1$, obsahujúci všetky vrcholy v $S$, prechádzajúci každou hranou nanajvýš 2 razy a končiaci v nejakom základnom vrchole. Taký sled $W$ možno získať napr. Tarryho prieskumom, ako ilustruje obr. 5. Nech $u_1, u_2, \ldots, u_k$ je postupnosť základných vrcholov v tom poradí, v akom sa v slede $W$ prvýkrát vyskytujú. Tento $u_1$—$u_k$ sled $W$ možno zrejme rozložiť na $k-1$ ciest: $u_1$—$u_2$ cesta $Q_{12}$, $u_2$—$u_3$ cesta $Q_{23}$, …, $u_{k-1}$—$u_k$ cesta $Q_{k-1,k}$. Vzdialenosť $d_G(u_i, u_{i+1}) \leq c(Q_{i,i+1})$ pre každé $i = 1, 2, \ldots, k-1$. Nech $T$ je najlacnejšia kostra v $K(B)$ a nech




Obr. 5. Strom $S$ a sled $W$.



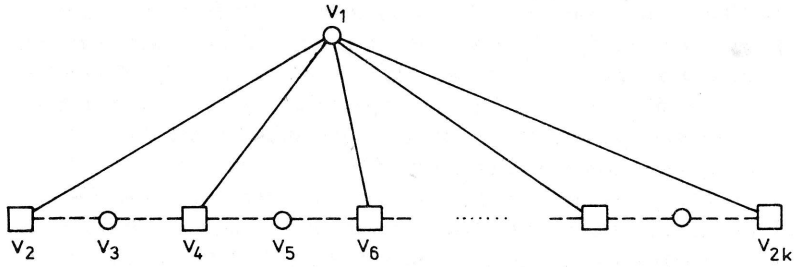


$\tilde{T}$ je kostra v $K(B)$ pozostávajúca z hrán $u_1u_2$, $u_2u_3$, …, $u_{k-1}u_k$. Potom môžeme písať:

$$c(T) \leq c(\tilde{T}) = \Sigma\, c_{K(B)}(u_iu_{i+1}) = \Sigma\, d_G(u_i, u_{i+1}) \leq \Sigma\, c(Q_{i,i+1}) = c(W) < 2c(S),$$

čo sme mali dokázať.

Teraz dokážeme, že $\varrho_{\text{graph}} \geq 2$. K tomu uvažujme graf $G$ z obr. 6 s $2k$ vrcholmi, z ktorých je $k$ základných (vyznačené štvorčekmi) a každá hrana má cenu 1. Najlacnejší steinerovský strom $S$ je vyznačený plnými čiarami a najlacnejšia kostra $T$, resp. korešpondujúci strom $T'$, prerušovanými čiarami. Vidíme, že $c(S) = k$ a $c(T) = 2k-2$. Teda podiel $c(T)/c(S) \to 2$ ak $k \to \infty$. ■




Obr. 6. K dôkazu vety 3.

Poznamenajme, že v uvedenej metóde nie je potrebné konštruovať graf $K(B)$, ale najlacnejšiu kostru možno konštruovať priamo z grafu $G$. Dokonca môžeme hneď konštruovať steinerovský strom $T'$ napodobňujúc nejaký algoritmus pre najkratšiu kostru. Takto to robia Takahashi a Matsuyama (1980), ktorí používajú Primov algoritmus. Vychádzajú zo stromu $T_1$ pozostávajúceho z jedného vrcholu $u \in B$. Ak už máme nejaký strom $T_i$, tak nájdeme vrchol $x \in B - V(T_i)$ najbližší k stromu $T_i$, a potom nejakú najkratšiu $x$—$V(T_i)$ cestu pridáme k $T_i$, čím získame strom $T_{i+1}$. Pre nejaké $i \leq k$ už $T_i$ zrejme zahŕňa $B$, a teda je steinerovským stromom.

Iná metóda spočíva v tom, že ak $r$ je dĺžka (t. j. cena) najlacnejšej hrany grafu, tak pre každý základný vrchol $u$ všetky body hrán do vzdialenosti $r$ od $u$ stiahneme do $u$. Tým získame „menší" graf a postup môžeme opakovať, až získame graf s jediným základným vrcholom, z ktorého potom postupne získame steinerovský strom pre pôvodný graf. Tento postup je podrobne opísaný v práci [Plesník (1981)], kde je aj ukázané, že získame strom, ktorého cena je vždy menšia ako dvojnásobok optima. Žial. obr. 6 aj tu ukazuje, že pomer k optimu ide k 2 pre $k \to \infty$. Polynomiálny algoritmus s lepším pomerom však nie je známy.

## 5.Z
## Cvičenia a poznámky

1. Dokážte nasledujúcu analógiu vety 5.A.1: Maximálny počet nezávislých uzavretých sledov v silne súvislom pseudodigrafe je rovný $m-n+1$. [Návod: Nech $e=uv$ je neslučka ležiaca v cykle; vynechajte $e$ a vrcholy $u$, $v$ stotožnite; použite indukciu.] [Berge (1958).]
2. Dokážte, že počet kostier grafu môže byť nepolynomiálny, avšak vždy ohraničený číslom $2^m$.
   V teórii grafov sa dokazuje, že kompletný graf $K_n$ má práve $n^{n-2}$ kostier (pozri napr. [Harary (1969)], [Moon (1970)], [Nešetřil (1979)], a i.)
3. (a)Ukážte, že nasledujúci algoritmus [Kruskal (1956)] nájde najlacnejšiu kostru. V každom kroku vynecháme z grafu najdrahšiu možnú hranu, pre ktorú zvyšok zostane ešte súvislý. Po $m-(n-1)$ krokoch máme výsledok.
   (b) Dokážte, že pred hľadaním najlacnejšej kostry v kompletnom grafe môžeme vynechať ľubovoľných $n-2$ najdrahších hrán.
4. (a) Vyplňte medzery v dôkaze vety 5.C.4.
   (b) Dajte podrobnejšiu implementáciu algoritmu pre P($k$) z časti 5.C.
5. Ukážte, že problém nájsť kostru s predpísanými stupňami nepresahujúcimi 2 je NP-ťažký dokonca pre planárne kubické grafy. [Návod: Uvážte problém hamiltonovskej cesty.]
   Poznamenajme, že problém nájsť kostru s maximálnym počtom koncových vrcholov je tiež NP-ťažký [Garey, Johnson (1979)].
6. Rozhodnite, či vety 5.C.2 až 4 platia aj vtedy, ak $E_0$ je ľubovoľná množina hrán (nie nevyhnutne okolitých hrán jedného vrcholu).
7. Uveďte príklad ukazujúci, že nasledujúca modifikácia Kruskalovho algoritmu nedáva $\varrho$-aproximačný algoritmus pre problém MP z 5.D pri žiadnom pevnom $\varrho$: V každom kroku vyberáme do kostry takú hranu, aby sme získali les s minimálnou parciálnou účelovou funkciou.
8. Ukážte, že v algoritme z časti 5.E (poznámky o implementácii) platí to, čo sa tam tvrdí, t. j.: $T_i=T$, $IN_i^*=I_1\cup I_2$, $OUT_i^*=O_1$.
9. Nech $T_1, \ldots, T_p$ je $p$ najlacnejších kostier grafu $G$. Dokážte, že $(p+1)$-ú najlacnejšiu kostru stačí hľadať medzi susedmi kostier $T_1, \ldots, T_p$.
10. Pretransformujte všeobecnú úlohu nájsť najlacnejšiu zdrojovú kostru digrafu na úlohu s kladnými cenami.
11. Uvažujte úlohu nájsť najlacnejšiu zdrojovú kostru digrafu s predpísaným zdrojom. Ukážte: (a) Túto úlohu možno previesť na úlohu s voľným zdrojom. (b) Pre túto úlohu možno modifikovať algoritmus z 5.F, krok 2 tak, že





$c(\hat{e}_w):=c(e_w)-c(e'_w)+k_t$, kde $k_t$ je ľubovoľné číslo priradené cyklu $Z_t$ (napr. $k_t=0$). Ďalej ukážte, že ak zvolíme $k_t=c(Z_t)$, tak $c(T)=c(\hat{T})$.

12. (a) Ukážte, že úlohu nájsť najlacnejší faktor digrafu, ktorého každý komponent je zdrojový strom, možno previesť na úlohu o zdrojovej kostre pridaním nového vrcholu (zdroja).
    (b) Ukážte, že ak v úlohe (a) žiadame práve $k$ komponentov, tak v algoritme z 5.F stačí v kroku 1 vybrať $n-k$ najlacnejších hrán a potom pre $\hat{G}$ hľadať tiež „$k$-zdrojový les".
13. Dokážte, že v rektilineárnom Steinerovom probléme sa môžeme obmedziť len na také vertikálne a horizontálne priamky roviny, ktoré prechádzajú základnými bodmi [Hanan (1966)].
14. Dokážte, že rektilineárny Steinerov problém v rovine je NP-ťažký. [Návod: Urobíme nasledujúcu polynomiálnu transformáciu z problému súvislého vrcholového pokrytia planárneho grafu s maximálnym stupňom 4 (2.Z.11). Nech daný graf $G$ má $n$ vrcholov a $m$ hrán. Uvažujme v rovine štvorčekovú mriežku M s mrežovými bodmi tvaru $(6in^2, 6jn^2)$, pre všetky celočíselné $i$ a $j$. (a) Najprv zakreslíme diagram grafu $G$ v mriežke M tak, že vrcholy umiestnime v mrežových bodoch, hrany pozostávajú z čiar mriežky a výsledok je rovinný diagram. (To možno urobiť postupným pridávaním stien v polynomiálnom čase). Tie čiary z mriežky, ktoré sme použili na zakreslenie, nazvime hranové segmenty. (b) Množinu $B$ základných bodov pre Steinerov problém zostrojíme takto:
    Pre každý mrežový bod $p_i$ zodpovedajúci vrcholu $v_i$ grafu $G$ vynecháme zo všetkých hranových segmentov pri $p_i$ úseky bodov so vzdialenosťou od $p_i$ menšou ako 2. Nech $L$ je sumárna dĺžka zvyšných častí hranových úsekov a nech $B$ je množina všetkých bodov na nich s celočíselnými súradnicami. Dokážte, že $G$ má súvislé vrcholové pokrytie mohutnosti $k$ $\Leftrightarrow$ existuje rektilineárny steinerovský strom ceny $c:=L+2m+2(k-1)$.] [Garey, Johnson (1977).]
15. Dokážte, že Steinerov problém v grafoch je NP-ťažký dokonca pre planárne neohodnotené grafy (všetky $c_{ij}=1$). [Návod: Použite 5.Z.14.]
16. Dokážte, že ak $uv$ je najlacnejšia hrana súvislého grafu a $u$, $v$ sú základné vrcholy, tak existuje najlacnejší steinerovský strom obsahujúci hranu $uv$.
17. Uvážte nasledujúce zosilnenie metódy kostry pri hľadaní najlacnejšieho steinerovského stromu pre $B$ v grafe $G$: Nech je dané pevné číslo $q$. Utvoríme všetky možné $K(B\cup Q)$, kde $Q\subseteq V(G)-B$ a $|Q|=q$, nájdeme v nich najlacnejšie kostry a vyberieme najlacnejšiu z nich. Ukážte na príklade neohodnoteného grafu, že ani tento postup nedáva $\varrho$-aproximáciu s $\varrho<2$.

18. Sformulujte Steinerov problém v grafoch ako 0—1 úlohu lineárneho programovania.
19. Nech hrana $e$ grafu $G$ má dĺžku $a(e)$ a cenu $c(e)$. Pre dané číslo $B$ (rozpočet) treba nájsť faktor $F$ grafu $G$ s minimálnym priemerom (daným dĺžkami hrán) a taký, že jeho cena (súčet cien jeho hrán) nepresahuje $B$. Dokážte, že tento problém je NP-ťažký. [Návod: Transformujte problém minimálneho vrcholového pokrytia hrán.]

    V [Plesník (1981a)] je dokázané, že dokonca aj vtedy, keď $B$ nás obmedzuje len na kostry je $\varrho$-aproximačný problém NP-ťažký pre $\varrho < 2$.

    Ak namiesto minimálneho priemeru žiadame, aby $\Sigma\, d_F(u, v)$ bolo minimálne (sumujeme cez všetky dvojice vrcholov $(u, v)$), tak tiež ide o NP-ťažký problém dokonca aj vtedy, keď všetky dĺžky a ceny hrán sú jednotkové a $B$ nás obmedzuje na kostry, t. j. $B = n - 1$ [Johnson, Lenstra, Rinnooy Kan (1978)].
20. Nech je daný graf $G$, kde každá hrana $ij$ má šírku $b_{ij}$. Dokážte, že stačí nájsť jednu najcennejšiu kostru v $G$, kde cena hrany $ij$ je $b_{ij}$, aby sme určili najširšiu $i$—$j$ cestu v $G$ (pozri 4.F) pre každú dvojicu $[i, j]$.

# 6
# TOKY

Pojem „tok", ktorý budeme študovať, je abstrakciou ustálených fyzikálnych tokov (tok vody, elektriny, automobilov, materiálu a pod.) v príslušných sieťach (vodovodná, elektrická, cestná, dopravná sieť a pod.). Dôležitým podnetom pre vznik teórie boli problémy z dopravy produktov k spotrebiteľom. Takéto úlohy možno formulovať ako úlohy lineárneho programovania. Avšak v pojmoch tokov možno ľahko formulovať mnoho zaujímavých a dôležitých kombinatorických úloh, ktoré často vôbec nesúvisia s reálnymi tokmi. Teória tokov má aplikácie aj v teórii grafov.

V tejto kapitole sa venujeme hlavne úlohe o maximálnom toku (preprava maximálneho množstva hmoty pri obmedzených priepustnostiach prepravných liniek). Jedno zovšeobecnenie tejto úlohy je prepraviť predpísané množstvo hmoty pri minimálnych nákladoch. Obe tieto úlohy majú tú peknú vlastnosť, že medzi optimálnymi riešeniami existuje aj celočíselné optimum (pri celočíselných vstupných údajoch) a existujú polynomiálne algoritmy pre jeho nájdenie. Uvedieme viaceré aplikácie týchto úloh. Okrem jednoduchých variácií základnej úlohy si všimneme aj viacproduktové toky a ďalšie zovšeobecnenia.

## 6.A
## Úloha o maximálnom toku

Nech $G$ je digraf, v ktorom sú vyznačené dva vrcholy: *vstup* $s$ a *výstup* $t$: ostatné vrcholy sú tzv. *prechodové*. Ďalej nech každá hrana $e \in E(G)$ má priradené nezáporné reálne číslo $b(e)$, tzv. *kapacitu* (tiež: *priepustnosť* alebo *šírku*) hrany $e$. (Kapacita vyjadruje maximálne množstvo hmoty, ktoré možno za časovú jednotku po danej komunikácii prepraviť.) Zjednodušene často píšeme $b_{ij}$ namiesto $b(ij)$.

Funkcia $x$: $E(G) \to R$ sa nazýva *tok*, presnejšie $s$—$t$ tok v $G$, ak spĺňa podmienky:

$$0 \leq x_{ij} \leq b_{ij}, \quad ij \in E(G) \quad \text{(kapacitné ohraničenia)},$$

$$\sum_{j \in V^-(i)} x_{ji} - \sum_{j \in V^+(i)} x_{ij} = 0, \quad i \in V(G) - \{s, t\} \quad \text{(podmienky kontinuity)}.$$

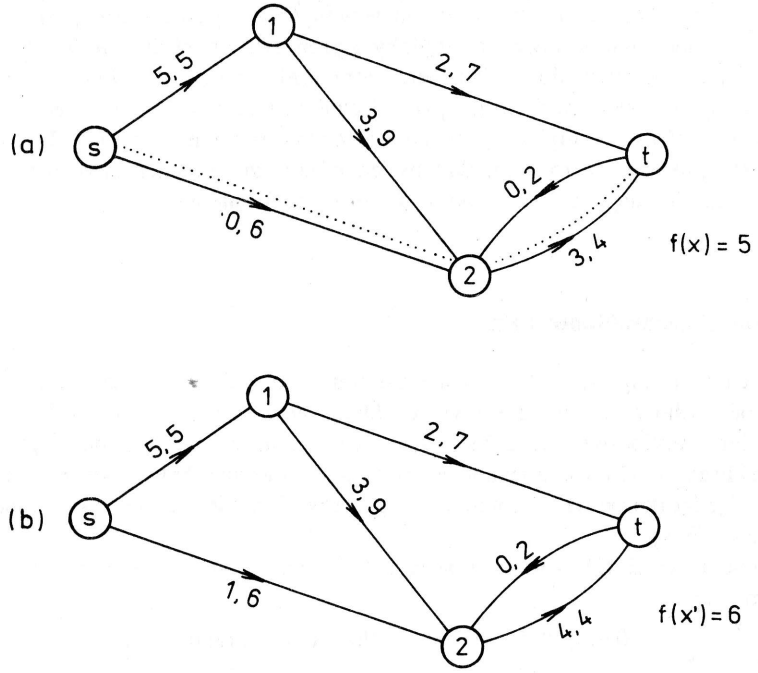
Číslo $x_{ij}$ je množstvo hmoty, ktoré „tečie" šípom $ij$. Podmienky kontinuity vyjadrujú jednoduchý fakt, že to, čo do vrcholu priteká, to z neho aj odteká : teda je to vlastne Kirchhoffov prúdový zákon.

Číslo

$$f(x) := \sum_{j \in V^-(t)} x_{jt} - \sum_{j \in V^+(t)} x_{tj}$$

nazývame *veľkosť* (alebo *hodnota*) $s$—$t$ toku $x$. $f(x)$ je vlastne „čisté" množstvo, ktoré pritečie do $t$. Vzhľadom na podmienky kontinuity vidíme, že $f(x)$ je „čistý odtok" z $s$, t. j. $f(x) = -(\Sigma x_{js} - \Sigma x_{sj})$.

Pod úlohou *o maximálnom toku* rozumieme nájsť $s$—$t$ tok s maximálnou veľkosťou. Taký tok sa nazýva *maximálny*. Medzi prvých, ktorí túto úlohu študovali explicitne v digrafoch, patria Ford a Fulkerson (1956), hoci tento problém vystupoval v rôznych úlohách o preprave už prv.




Obr. 1. (a) Tok $x$ (druhé číslo je kapacita hrany).
(b) Tok $x'$ získaný z $x$ zväčšením po ceste $s2t$.

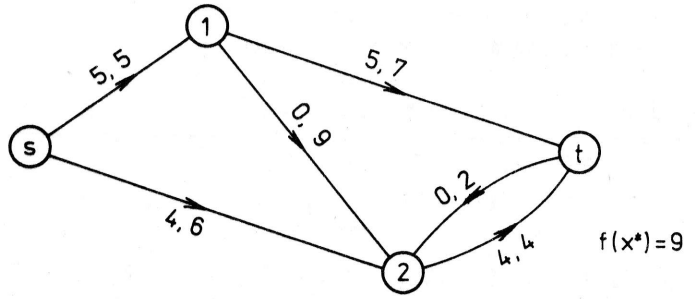




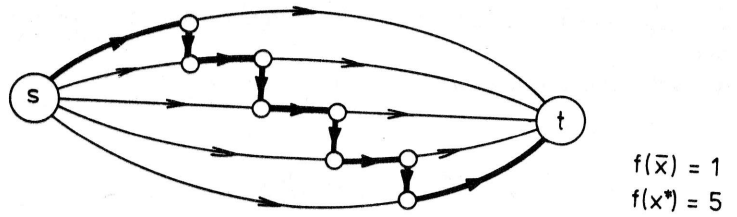
Vidíme, že kapacitné ohraničenia a podmienky kontinuity sú lineárne (v $x_{ij}$) a účelová funkcia $f(x)$ je tiež lineárna. Preto úloha o maximálnom toku je úlohou lineárneho programovania, avšak dosť špeciálnou, a preto má zmysel hľadať špeciálne algoritmy na jej riešenie. V tejto úlohe lineárneho programovania je množina prípustných riešení neprázdna (napr. existuje *nulový tok*, t. j. $x_{ij} := 0$ pre všetky $ij \in E(G)$), polyedrická, teda uzavretá, a ohraničená (kapacitné ohraničenia). Preto spojitá (dokonca lineárna) funkcia $f(x)$ má tam maximum. Teda maximálny tok vždy existuje.

Príklad na obr. 1(a) ukazuje tok $x$ veľkosti 5 : na každej hrane $ij$ je dvojica čísel $(x_{ij}, b_{ij})$. Na obr. 1(b) je tok $x'$, ktorý sme získali z $x$ tak, že na ceste $s2t$ sme zväčšili tok $x$ o jednotku. Avšak takýmto spôsobom už nemožno získať väčší tok z $x'$, keďže na každej $s$—$t$ ceste existuje aspoň jedna hrana $ij$ taká, že $x'_{ij} = b_{ij}$. Takáto hrana sa nazýva *nasýtená* a cesta obsahujúca nasýtenú hranu *blokovaná*. Taký tok, pri ktorom sú všetky $s$—$t$ cesty blokované, nazývame *blokujúci tok*.

Žiaľ, blokujúci tok nemusí byť maximálny (obr. 2 ukazuje maximálny tok $x^*$ pre digraf z obr. 1 : má veľkosť 9). Z obr. 3 je vidieť, že blokujúcim tokom $\bar{x}$ nemožno aproximovať maximálny tok $x^*$ dokonca ani v acyklickom digrafe, kde sú všetky




Obr. 2. Maximálny tok $x^*$.




Obr. 3. Blokujúci tok $\bar{x}$ (všetky $b(e) = 1$ : na hrubých šípoch je $\bar{x}(e) = 1$, inde 0).

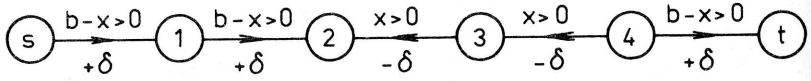
kapacity jednotkové (na hrubo vyznačených hranách $ij$ je $\bar{x}_{ij}=1$ a inde $\bar{x}_{ij}=0$: nájdite tok $x^*$ s $f(x^*)=5$).

## VETA O MAXIMÁLNOM TOKU A MINIMOVOM REZE

Ford a Fulkerson (1956) navrhli silnejší prostriedok na zväčšovanie daného toku $x$. Nech $P$ je nejaká polocesta, nech $P^+$ je množina súhlasných a $P^-$ množina nesúhlasných šípov polocesty $P$. Čísla $b_{ij}-x_{ij}$ pre $ij\in P^+$ a $x_{ij}$ pre $ij\in P^-$ sa nazývajú *rezervy* na poloceste $P$: minimálna z rezerv na $P$ sa nazýva *rezerva* polocesty $P$. Ak je rezerva nejakej polocesty $P$ kladná, tak $P$ sa nazýva *rezervná polocesta*. Rezervná $s$—$t$ polocesta sa nazýva aj *zväčšujúca polocesta pre* $x$. Tento názov je oprávnený, lebo ak $\delta>0$ je rezerva $s$—$t$ polocesty $P$, tak funkcia $x'$: $E(G)\to R$, kde

$$x'_{ij}=\begin{cases} x_{ij}+\delta, & \text{pre } ij\in P^+,\\ x_{ij}-\delta, & \text{pre } ij\in P^-,\\ x_{ij} \end{cases}$$

inak je opäť tokom v $G$ a $f(x')=f(x)+\delta$. Totiž kapacitné ohraničenie aj podmienky kontinuity sa dodržia, ako vidieť z príkladu na obr. 4 (napr. vo vrchole 1 bude o $\delta$ viac pritekať, ale aj o $\delta$ viac odtekať).




Obr. 4. Zväčšujúca polocesta.

Ak $G$ je digraf a $A$, $B\subseteq V(G)$, tak $(A, B)_G:=(A\times B)\cap E(G)$. Nech $S\subset V(G)$, $\bar{S}:=V(G)-S$ a nech $s\in S$, $t\in\bar{S}$. Potom množinu hrán $H:=(S,\bar{S})_G$ nazývame (hranový) $s$—$t$ *rez*. Je zrejmé, že každá $s$—$t$ cesta obsahuje aspoň jednu hranu z ľubovoľného $s$—$t$ rezu. Obr. 5 ilustruje tento pojem. *Kapacita* $b(H)$ rezu $H$ je súčet kapacít jednotlivých hrán rezu. Rez s minimálnou kapacitou sa nazýva *minimový*.

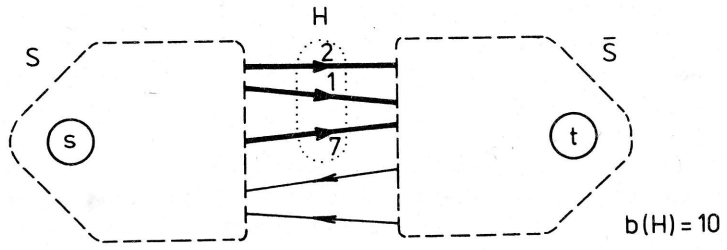
Pre ľubovoľnú funkciu $h$: $E(G)\to R$ a množiny $A$, $B\subseteq V(G)$ budeme stručne písať:

$$h(A,B)=\sum_{ij\in(A,B)G} h_{ij}.$$

Ak $x$ je nejaký $s$—$t$ tok a $(S,\bar{S})_G$ nejaký $s$—$t$ rez, tak zrejme $f(x)=x(S,\bar{S})$








Obr. 5. $s$—$t$ rez kapacity 10.

$-x(\bar{S},S)\leqslant b(S,\bar{S})$, lebo platia podmienky kontinuity a kapacitné ohraničenia. Táto nerovnosť nám pripomína tzv. slabú vetu o dualite v lineárnom programovaní. Preto očakávame, že v extrémnych prípadoch dosiahneme rovnosť. Skutočne je to tak a nasledujúca veta môže byť aj odvodená z vety o dualite v lineárnom programovaní, ale tu uprednostníme pôvodný grafový dôkaz [Ford, Fulkerson (1956)], ktorý je jednoduchší a konštruktívny.

**Veta 1.** Veľkosť maximálneho $s$—$t$ toku je rovná kapacite minimového $s$—$t$ rezu.

DÔKAZ. Najprv poznamenajme, že veta má vždy zmysel, lebo maximálny tok vždy existuje (maximalizujeme spojitú funkciu na kompakte, ako sme už spomínali: neskoršie zaručíme maximálny tok aj algoritmicky) a minimový rez tiež (lebo je len konečne mnoho rezov).

Nech $x^*$ je maximálny $s$—$t$ tok a $b^*$ nech je kapacita minimového $s$—$t$ rezu. Podľa predchádzajúceho je $f(x^*)\leqslant b^*$. Teraz nájdeme taký $s$—$t$ rez $H$, že $f(x^*)=b(H)$, a tým bude dôkaz hotový. Nech $S\subseteq V(G)$ je množina vrcholov obsahujúca $s$ a každý taký vrchol $v$, pre ktorý existuje rezervná $s$—$v$ polocesta. Zrejme $t\notin S$, lebo inak by existovala zväčšujúca $s$—$t$ polocesta pre $x^*$, čo protirečí maximalite toku $x^*$. Teda $t\in\bar{S}$. Každá hrana $ij$ idúca z $S$ do $\bar{S}$ je nasýtená, lebo inak z existencie rezervnej $s$—$i$ polocesty by vyplývala existencia rezervnej $s$—$j$ polocesty, čo je nemožné ($j\in\bar{S}$). Analogicky na každej hrane idúcej z $\bar{S}$ do $S$ je tok $x$ nulový. Takto $f(x^*)=x^*(S,\bar{S})=b(S,\bar{S})$. ■

Z algoritmického hľadiska je významné nasledujúce tvrdenie [Ford, Fulkerson (1957)].

**Veta 2** (*o optimalite toku*). Nejaký $s$—$t$ tok $x$ je maximálny $\Leftrightarrow$ neexistuje rezervná $s$—$t$ polocesta pre $x$.

DÔKAZ. ($\Rightarrow$) Pomocou zväčšujúcej polocesty by sa dal tok $x$ zväčšiť.

($\Leftarrow$) Nech $S := \{v|$ existuje rezervná $s$—$v$ polocesta pre $x\}$. Potom tak ako v predchádzajúcom dôkaze vidíme, že $f(x) = b(S, \bar{S})$, a teda $x$ je maximálny tok a $(S, \bar{S})_G$ minimový rez. ■

## FORDOV A FULKERSONOV ALGORITMUS

Veta 2 nám ponúka nasledujúci spôsob hľadania maximálneho toku: Nájdeme nejaký tok (napr. nulový) a ten postupne zväčšujeme po zväčšujúcich polocestách. Otázkou zostáva hľadanie zväčšujúcich polociest.

Ford a Fulkerson (1957, 1962) navrhli postupne generovať množinu vrcholov dosiahnuteľných z $s$ po rezervných polocestách. K tomu sa používajú značky vrcholov: každý označkovaný vrchol $j \neq s$ má značku tvaru $(i^+, \delta_j)$ alebo $(i^-, \delta_j)$, ktorá značí, že sme našli rezervnú $s$—$j$ polocestu s rezervou $\delta_j$ a posledný šíp tejto polocesty je v prvom prípade $ij$ a v druhom $ji$. Vrchol $s$ má značku $(-, \infty)$ (trvale). Vo všeobecnosti zoberieme ľubovoľný označkovaný, ale nepreskúmaný vrchol $i$ a preskúmame ho takto:

Každý neoznačkovaný vrchol $j \in V^+(i)$ dostane značku

$$(i^+, \min\{\delta_i, b_{ij} - x_{ij}\}), \qquad \text{ak } x_{ij} < b_{ij},$$

a každý neoznačkovaný vrchol $j \in V^-(i)$ značku

$$(i^-, \min\{\delta_i, x_{ji}\}), \qquad \text{ak } x_{ji} > 0.$$

(Značky priraďujeme postupne, takže aj vrchol $j \in V^+(i) \cap V^-(i)$ dostane len jednu značku.) Tým je vrchol $i$ preskúmaný.

Ak $t$ dostane nejakú značku, tak značkovanie skončíme a identifikujeme rezervnú $s$—$t$ polocestu, po ktorej vrchol $t$ dostal značku. To je ľahké zistiť podľa prvých zložiek značiek: posledným šípom tejto polocesty bude šíp $zt$, ak $t$ má značku $(z^+, \delta_t)$ a šíp $tz$, ak $t$ má značku $(z^-, \delta_t)$ atď. Po tejto poloceste potom zväčšíme tok o $\delta_t$ (čo je minimálna rezerva na nej) a získame nový tok $x'$. Potom všetky značky zotrieme a značkujeme znova (pre tok $x'$).

Ak sa pri nejakom značkovaní vrchol $t$ neoznačkuje, to značí, že neexistuje rezervná $s$—$t$ polocesta a príslušný tok je podľa vety 2 maximálny. Ak $S$ je množina označkovaných vrcholov, tak z dôkazu vety 1 vidíme, že potom $(S, \bar{S})_G$ je minimový rez. ●

Všimnime si zložitosť Fordovho a Fulkersonovho algoritmu. Ak máme k dispozícii zoznamy okolí jednotlivých vrcholov, tak na celé značkovanie potrebné na nájdenie zväčšujúcej polocesty vystačíme s $O(m)$ operáciami. Identifikáciu zväčšujúcej $s$—$t$ polocesty a zväčšenie toku po nej možno urobiť v čase $O(n)$.





Sumárne jedna iterácia nepresiahne $O(m)$ operácií a celý algoritmus $qO(m)$ operácií, kde $q$ je počet iterácií. Žiaľ, $q$ môže byť veľmi veľké, dokonca nekonečné, ako vidieť z Fordovho a Fulkersonovho príkladu (1962) (pozri 6.Z.1), kde sa ani v limite nezíska maximálny tok.

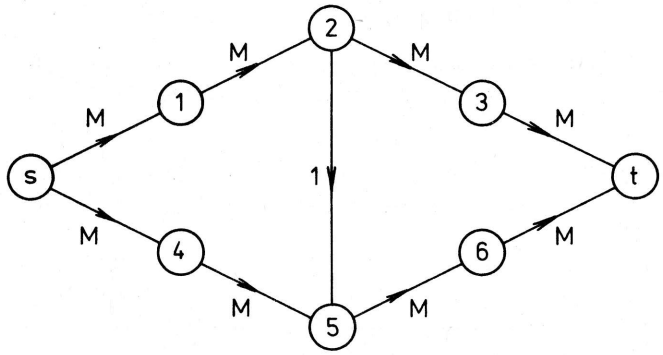
Ukazuje sa však, že v prípade celočíselných kapacít (a aj v prípade racionálnych, resp. všeobecnejšie, ak podiel ľubovoľných dvoch kapacít je racionálne číslo — súmerateľné kapacity) je tento algoritmus konečný, ako vidieť z nasledujúceho výsledku, resp. jeho dôkazu.

**Veta 3.** Nech sú všetky kapacity celé čísla a nech $f^*$ je hodnota maximálneho toku. Potom ak začíname s celočíselným tokom (napr. nulovým), tak Fordov a Fulkersonov algoritmus nájde po nanajvýš $f^*$ iteráciách maximálny tok. Naviac tento tok bude celočíselný.

DÔKAZ. V každej iterácii je $\delta$ celé číslo a veľkosť toku vzrastie o $\delta$, čo je aspoň jednotka. ■

Spomínaný Fordov a Fulkersonov príklad podnietil niektorých autorov k hľadaniu konečných modifikácií algoritmu (napr. Hu (1969), ale Poinstein (1972) zistil jeho nekonečnosť a sám navrhol modifikáciu, ktorá je už konečná).

Táto otázka nie je však až taká dôležitá, lebo pre praktické úlohy je konečnosť vždy zaručená (veta 3). Horšie je to, že počet iterácií môže byť až $f^*$, ako vidieť z príkladu na obr. 6, ktorý dali Edmonds a Karp (1970, 1972): Nech $M$ je veľké celé číslo. Vychádzajúc z nulového toku, zväčšíme ho o 1 po poloceste ($s$, 1, 2, 5, 6, $t$). Tento zase o 1 po poloceste ($s$, 4, 5, 2, 3, $t$), potom o 1 po poloceste ($s$, 1, 2,




Obr. 6. Fordov a Fulkersonov algoritmus tu urobí $2M$ iterácií.

5, 6, $t$) atď. Teda celkove takto bude treba $2M$ iterácií ($f^* = 2M$), čo exponenciálne závisí od vstupnej informácie, ktorá má dĺžku $O(\log_2 M)$ bitov. Preto Fordov a Fulkersonov algoritmus nie je polynomiálny, hoci sa v praxi ukazuje pomerne rýchly. Polynomiálne modifikácie uvedieme v nasledujúcej časti.

## 6.B
## Polynomiálne algoritmy pre maximálny tok

Tu uvedieme tri modifikácie Fordovho a Fulkersonovho algoritmu zamerané na zmenšenie počtu iterácií. Prvý z nich (algoritmus najkratších polociest) vystačí vždy s polynomiálnym počtom iterácií, zatiaľ čo dva ďalšie sú založené na celočíselnosti kapacít, inak dávajú maximálny tok len v limite.

### ALGORITMUS NAJKRATŠÍCH POLOCIEST

Ak Fordov–Fulkersonov algoritmus bude zväčšovať tok vždy po najkratšej zväčšujúcej poloceste, tak vystačíme s polynomiálnym počtom zväčšovaní toku. ●

Túto myšlienku navrhli a zrealizovali Edmonds a Karp (1970, 1972) a Dinic (1970).

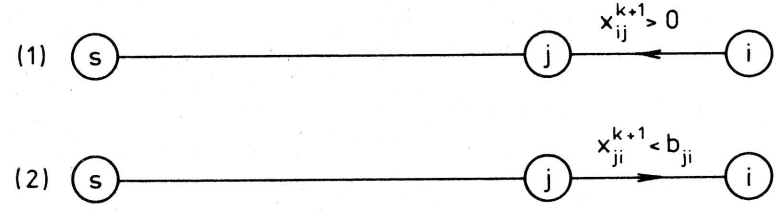
Pod dĺžkou polocesty budeme ako obvykle rozumieť počet jej hrán (ohodnotenie je kapacita). Zaveďme označenia:

$\sigma^k(i)$ = dĺžka najkratšej rezervnej $s$—$i$ polocesty po $k$ zväčšeniach toku,
$\tau^k(i)$ = dĺžka najkratšej rezervnej $i$—$t$ polocesty po $k$ zväčšeniach toku.

**Lema 1(a).** Pre každé $k$ a každý vrchol $i$ platí:

$$\sigma^{k+1}(i) \geq \sigma^k(i) \quad \text{a} \quad \tau^{k+1}(i) \geq \tau^k(i).$$

DÔKAZ. Prvú nerovnosť dokážeme sporom. Predpokladajme, že existuje $i$ a $k$ tak, že $\sigma^{k+1}(i) < \sigma^k(i)$. Budeme predpokladať, že $i$ je tak vybrané, že $\sigma^{k+1}(i)$ je




Obr. 1. K dôkazu lemy 1(a).





minimálne. Nech $j$ je predposledný vrchol nejakej najkratšej rezervnej $s$—$i$ polocesty $Q$ pre tok $x^{k+1}$ (t. j. po $k+1$ zväčšeniach toku) (obr. 1). Potom $\sigma^{k+1}(i) = \sigma^{k+1}(j) + 1$ a z voľby vrcholu $i$ je $\sigma^{k+1}(j) \geq \sigma^k(j)$, čo spolu dáva:

$$(*) \qquad \sigma^{k+1}(i) \geq \sigma^k(j) + 1.$$

Pre poslednú hranu polocesty $Q$ sú dve možnosti ilustrované na obr. 1.

(1) Nech $ij \in Q$. Nech tok $x^k$ bol zväčšený ($(k+1)$-vé zväčšenie) po zväčšujúcej $s$—$t$ poloceste $P$. Ak hrana $ij$ nepatrila do $P$, alebo bola nesúhlasná v $P$, tak $x_{ij}^k > 0$ (lebo $x_{ij}^{k+1} > 0$). Preto $\sigma^k(i) \leq \sigma^k(j) + 1$, čo spolu s $(*)$ dáva $\sigma^{k+1}(i) \geq \sigma^k(i)$ – spor s predpokladom. Ak hrana $ij$ bola súhlasná v $P$, tak bolo $\sigma^k(j) = \sigma^k(i) + 1$, a tým skôr získame spor.

Prípad (2) a dôkaz pre $\tau$ sú analogické, a preto ich prenechávame čitateľovi. ■

**Veta 1.** Ak tok vždy zväčšujeme po najkratšej rezervnej $s$—$t$ poloceste, tak na nájdenie maximálneho toku vystačíme s $O(mn)$ zväčšeniami toku (nezávisle od počiatočného toku).

DÔKAZ. Ako vieme, na každej zväčšujúcej poloceste je aspoň jedna hrana taká, na ktorej sa tok zmení na nulový, resp. rovný kapacite. Každú takú hranu nazveme *kritickou hranou* zväčšujúcej polocesty.

Nech $ij$ je kritická (súhlasná alebo nesúhlasná) hrana v $(k+1)$-ej zväčšujúcej poloceste $P$. Potom dĺžka polocesty $P$ je $m(P) = \sigma^k(i) + \tau^k(i) = \sigma^k(j) + \tau^k(j)$ (z definície čísel $\sigma$ a $\tau$). Ak sa hrana $ij$ znova vyskytne v nejakej (povedzme) $(r+1)$-ej ($r > k$) zväčšujúcej poloceste $Q$ a je to najskorší ďalší výskyt ($r$ je minimálne), tak sa tam objaví v opačnom smere. Teda ak $\tau^k(i) = \tau^k(j) + 1$, tak $\sigma^r(i) = \sigma^r(j) + 1$. Podľa lemy 1(a) máme $\sigma^r(j) \geq \sigma^k(j)$ a $\tau^r(i) \geq \tau^k(i)$, čo všetko spolu dáva:

$$m(Q) = \sigma^r(i) + \tau^r(i) \geq \sigma^k(j) + \tau^k(j) + 2 = m(P) + 2.$$

Každá polocesta má dĺžku nanajvýš $n-1$, a teda každá hrana môže byť kritickou nanajvýš $(n/2)$-krát. Keďže každá zväčšujúca polocesta obsahuje aspoň jednu kritickú hranu, tak môžeme urobiť nanajvýš $mn/2$ zväčšení toku. ■

Pre nájdenie najkratšej rezervnej $s$—$t$ polocesty pre tok $x^k$ možno upresniť značkovaciu metódu Forda a Fulkersona tak, že bude sledovať postup do šírky. Prehľadnejšie však je k toku $x^k$ zostrojiť tzv. *rezervný digraf* $G^k$, kde $V(G^k) = V(G)$ a $ij \in E(G^k)$ práve vtedy, keď alebo $ij \in E(G)$ a $x_{ij} < b_{ij}$ alebo $ji \in E(G)$ a $x_{ji} > 0$. Rezervné polocesty v $G$ zrejme zodpovedajú cestám v $G^k$.

Na zostrojenie digrafu $G^k$ stačí $O(m)$ operácií a na nájdenie najkratšej $s$—$t$ cesty v $G^k$, a tým zväčšujúcej polocesty v $G$ stačí tiež $O(m)$ operácií (algoritmus Moora). Na zväčšenie toku stačí $O(n)$ operácií. Takto uvážiac vetu 1 na celý algoritmus stačí $O(m^2n)$ operácií. Toto sa pre riedke digrafy redukuje na $O(n^3)$, ale pre husté máme len $O(n^5)$. (Zadeh (1972) ukázal, že v najhoršom prípade odhad $O(n^5)$ nemožno znížiť.)

Dinic (1970) zrealizoval myšlienku metódy najkratších polociest v $O(mn^2)$ algoritme tým, že nachádza všetky zväčšujúce polocesty danej dĺžky naraz. Ďalší pokrok vychádza z jeho práce. Karzanov (1974) dáva $O(n^3)$ algoritmus a Čerkasskij (1977) – $O(n^2m^{1/2})$. Posledné dva algoritmy používajú predtok (ignorujúci podmienky kontinuity), ktorý sa potom „vyrovnáva" na tok. Sleator a Tarjan (1980) dávajú $O(mn \log n)$ algoritmus. Toky v planárnych digrafoch skúma napr. Kučera (1981).

## ALGORITMUS NAJŠIRŠÍCH POLOCIEST

V tejto modifikácii Fordovho a Fulkersonovho algoritmu každé zväčšovanie toku sa uskutočňuje po najširšej rezervnej $s$—$t$ poloceste. ● (Šírka polocesty je rovná minimálnej rezerve hrany tejto polocesty.) Teda v každej iterácii sa tok maximálne zväčšuje. Takýto algoritmus uvádzajú Edmonds a Karp (1972).

V časti 4.F je daný $O(n^2)$ algoritmus pre nájdenie najširšej $s$—$t$ cesty. Ak ho použijeme v rezervnom digrafe $G^k$ toku $x^k$, tak získame požadovanú najširšiu rezervnú $s$—$t$ polocestu. Odhadnime počet iterácií.

**Veta 2.** Nech sú všetky kapacity celé čísla a nech $f^*$ je hodnota maximálneho toku. Potom algoritmus najširších polociest nájde maximálny tok nanajvýš po $1 + m \ln f^*$ zväčšovaniach toku.

DÔKAZ. Nech $x$ je nejaký tok a nech najširšia rezervná $s$—$t$ polocesta pre $x$ má rezervu $\delta > 0$. Odhadnime rozdiel $f^* - f(x)$. Nech množina $S$ obsahuje vrchol $s$ a všetky také vrcholy $v$, pre ktoré existuje rezervná $s$—$v$ polocesta s rezervou väčšou ako $\delta$. Zrejme $t \in \bar{S} := V(G) - S$, a teda $(S, \bar{S})_G$ je $s$—$t$ rez. Preto ak hrana $uv \in (S, \bar{S})_G$, tak $b(uv) - x(uv) \leq \delta$ a ak $uv \in (\bar{S}, S)_G$, tak $x(uv) \leq \delta$ (inak by sme $S$ mohli zväčšiť). Takto môžeme písať:

$$f^* - f(x) \leq b(S, \bar{S}) - f(x) = b(S, \bar{S}) - [x(S, \bar{S}) - x(\bar{S}, S)] =$$

$$= [b(S, \bar{S}) - x(S, \bar{S})] + x(\bar{S}, S) \leq \delta |(S, \bar{S})_G| + \delta |(\bar{S}, S)_G| \leq m\delta.$$

Z toho vyplýva, že pre nový tok $x'$ platí: $f^* - f(x) \leq m[f(x') - f(x)]$. Odtiaľ:

$$f^* - f(x') \leq \left(1 - \frac{1}{m}\right) [f^* - f(x)].$$





Ak štartujeme z toku $x^0$, tak po $k$ zväčšeniach toku získaný tok $x^k$ spĺňa vzťah:

$$f^* - f(x^k) \leq \left(1 - \frac{1}{m}\right)^k [f^* - f(x^0)] \leq \left(1 - \frac{1}{m}\right)^k f^* \leq e^{-k/m} f^*.$$

Z predpokladu celočíselnosti kapacít vyplýva, že ak $f^* - f(x^k) < 1$, tak tok $x^k$ už je maximálny. Toto zaručíne, ak $-k/m + \ln f^* < 0$, t. j. ak $k > m \ln f^*$. ■

V každej iterácii je najprácnejšie nájsť najširšiu rezervnú $s$—$t$ polocestu. Ako sme už uvádzali, na to máme $O(n^2)$ algoritmus. Takto na celý algoritmus vystačíme pri celočíselných kapacitách s $O(n^2m \ln f^*)$ operáciami, čo značí, že algoritmus je polynomiálny. Teoreticky to presahuje zložitosť najlepších známych algoritmov, ale v praxi sa tento algoritmus veľmi dobre osvedčil.

## ALGORITMUS ZJEMŇOVANIA STUPNICE

Tento algoritmus, skúmaný v prácach [Edmonds a Karp (1972)] a [Dinic (1973)] využíva takúto vlastnosť: Ak všetky kapacity sú násobkami celého čísla $p$, tak Fordov a Fulkersonov algoritmus konštruuje toky, ktoré sú násobkami čísla $p$.

Nech $\delta_{max}$ je šírka najširšej $s$—$t$ cesty v digrafe $G$, kde pod šírkou hrany rozumieme jej kapacitu. Položme

$$q = \lfloor \log_2 \delta_{max} \rfloor.$$

Teraz kapacitu každej hrany $e$ zaokrúhlime nadol pri stupnici s „jednotkou" $2^q$, t. j. $b(e)$ zmeníme na $b_q(e) := 2^q \lfloor b(e)/2^q \rfloor$. Potom Fordovým–Fulkersonovým algoritmom nájdeme maximálny tok $x_q^*$ v digrafe $G$ s kapacitami $b_q(e)$. Ďalej zjemníme stupnicu tak, že namiesto kapacity $b_q(e)$ dáme $b_{q-1}(e) := b_q(e) + 2^{q-1} [(b(e) - b_q(e))/2^{q-1}]$ (t. j. $b(e)$ zaokrúhlime nadol v stupnici s „jednotkou" $2^{q-1}$). Potom vychádzajúc z $x_q^*$ nájdeme (Fordovým–Fulkersonovým algoritmom) maximálny tok $x_{q-1}^*$ v digrafe $G$ pri kapacitách $b_{q-1}(e)$. Ďalšie zjemnenie dáva $b_{q-2}(e) := b_{q-1}(e) + 2^{q-2} \lfloor (b(e) - b_{q-1}(e))/2^{q-2} \rfloor$ a tok $x_{q-2}^*$ atď. Pre celočíselné kapacity $b(e)$ bude $b_0(e) = b(e)$, a preto $x_0^*$ je hľadaný maximálny tok. Pre neceločíselné kapacity možno pokračovať (niekedy do nekonečna) a získať postupnosť $x_1^*, x_2^*, \ldots$, ktorá konverguje k maximálnemu toku $x^*$ pre pôvodné kapacity. ● Máme však:

**Veta 3.** Ak sú všetky kapacity celé čísla, tak algoritmus zjemňovania stupnice nájde maximálny tok po nanajvýš $m(q + 1)$ zväčšovaniach toku.

DÔKAZ. Uvažujme situáciu pri kapacitách $b_k(e)$ po nájdení maximálneho toku $x_k^*$. Nech množina $S$ pozostáva z vrcholu $s$ a všetkých takých vrcholov $v$,

pre ktoré existuje rezervná $s$—$v$ polocesta s rezervou aspoň $2^k$. Zrejme $t \in \bar{S} := V(G) - S$, lebo inak by sa tok $x_k^*$ dal zväčšovať. Takto pre každú hranu $e_1 \in (S, \bar{S})_G$ je $b_k(e_1) - x_k^*(e_1) < 2^k$ a pre každú hranu $e_2 \in (\bar{S}, S)_G$ je $x_k^*(e_2) < 2^k$. Preto $b_k(e_1) - x_k^*(e_1) = 0$ aj $x_k^*(e_2) = 0$. Potom pri zjemnení stupnice bude $b_{k-1}(e_1) - x_k^*(e_1)$ alebo nula alebo $2^{k-1}$. Preto pri kapacitách $b_{k-1}(e)$ po každom zväčšení toku veľkosť toku vzrastie aspoň o $2^{k-1}$. Pre optimálny tok $x^*$ pri pôvodných kapacitách $b(e)$ platí:

$$f(x^*) \leq b(S, \bar{S}) \leq f(x_k^*) + 2^{k-1}|(S, \bar{S})_G|.$$

Teda nanajvýš $|(S, \bar{S})_G| \leq m$ zväčšovaní toku sa uskutoční pri zostrojovaní toku $x_{k-1}^*$ z toku $x_k^*$ pri kapacitách $b_{k-1}(e)$.

Úplne analogicky sa ukáže, že tok $x_q^*$ možno získať z počiatočného nulového toku $x_{q+1}^*$ pomocou nanajvýš $m$ zväčšovaní. Preto na prechod od $x_{q+1}^*$ ku $x_0^* = x^*$ vystačíme s $m(q+1)$ zväčšovaniami toku. ■

Ako vieme, na každé zväčšovanie vystačíme s $m$ operáciami, a preto na celý algoritmus s $O(m^2(1 + \log_2 \delta_{max}))$ operáciami. (Na zistenie $\delta_{max}$ stačí $O(n^2)$ operácií.) Keďže $\delta_{max} \leq b_{max}$ (maximálna kapacita hrany), algoritmus je zrejme polynomiálny.

## 6.C
## Variácie úlohy o maximálnom toku

Niektoré obmeny úlohy o maximálnom toku možno previesť na základnú úlohu, resp. analogicky riešiť, ale iné môžu byť veľmi ťažké.

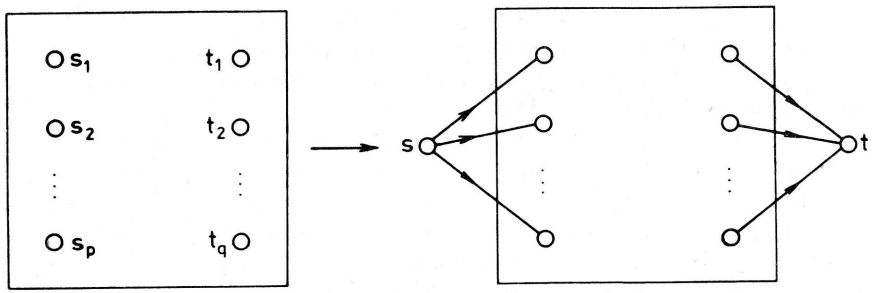
### TOK Z VIACERÝCH VSTUPOV DO VIACERÝCH VÝSTUPOV

V prípade, že je daných viac vstupov $s_1, s_2, \ldots, s_p$ a viac výstupov $t_1, t_2, \ldots, t_q$, tak problém nájsť maximálny sumárny tok z $\{s_1, s_2, \ldots, s_p\}$ do $\{t_1, t_2, \ldots, t_q\}$ možno ľahko redukovať na úlohu s jedným vstupom a jedným výstupom tak, ako ukazuje obr. 1. Tu predpokladáme, že tok môže tiecť z ľubovoľného vstupu do ľubovoľného výstupu. (Inak ide o tzv. viacproduktové toky, pozri 6.E.) Vrcholy $s$ a $t$ sú nový vstup a výstup, pričom hrany $ss_i$ aj $t_j t$ majú nekonečné kapacity. Ak je však vstup (výrobca) limitovaný, tak to možno vyjadriť kapacitou hrany $ss_i$: analogicky pre výstupy (spotrebiteľov).

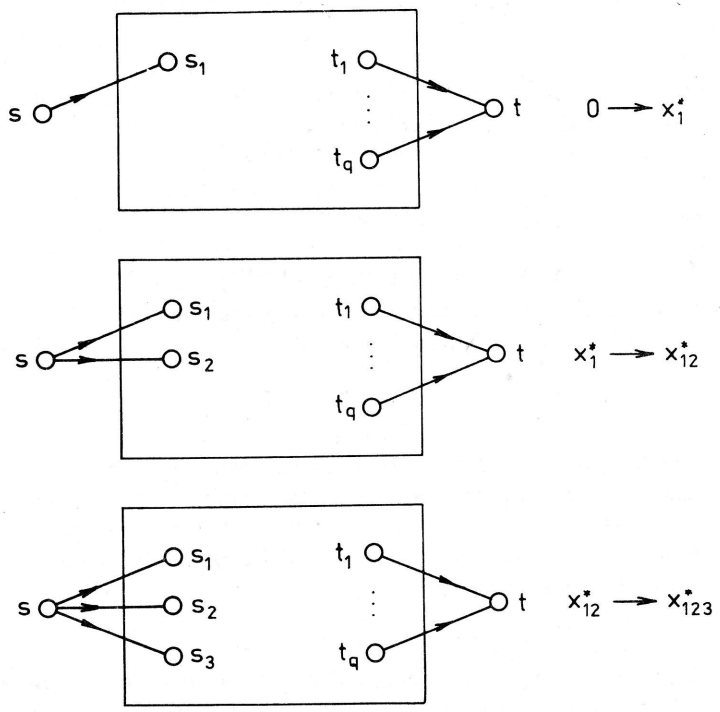
Niekedy sa hľadá taký maximálny tok, ktorý spĺňa priority vstupov, resp. výstupov [Dinic (1973)], [Minieka (1973)]. Nech $x$ je tok zo vstupov $s_1, s_2, \ldots, s_p$








Obr. 1. Redukcia na úlohu s jedným vstupom a výstupom.




Obr. 2. Postupné konštruovanie lexikograficky maximálneho toku podľa vstupov.

do výstupov $t_1, t_2, \ldots, t_q$ a $f_{si}^+(x)$ označuje čistý odtok z $s_i$ a $f_{tj}^-(x)$ čistý prítok do $t_j$. Potom tok $x$ sa nazýva *lexikograficky maximálny podľa vstupov*, ak vektor $(f_{s1}^+(x), f_{s2}^+(x), \ldots, f_{sp}^+(x))$ je lexikograficky maximálny. Analogicky definujeme *lexikograficky maximálny tok podľa výstupov.*

Je zrejmé, že lexikograficky maximálny tok podľa vstupov (alebo výstupov) je maximálny, lebo inak by existovala zväčšujúca $s_i$—$t_j$ polocesta, a teda by sme našli tok lexikograficky väčší.

Lexikograficky maximálny tok podľa vstupov možno nájsť postupne tak, ako je ukázané obr. 2. Všetky výstupy sú spojené do umelého výstupu $t$, pričom šípy $t_j t$ majú nekonečné kapacity. Najprv spojíme umelý vstup $s$ len s vrcholom $s_1$ pomocou šípa $ss_1$ (nekonečnej kapacity) a nájdeme maximálny $s$—$t$ tok $x_1^*$. Potom pridáme hranu $ss_2$ a vychádzajúc z $x_1^*$, nájdeme maximálny $s$—$t$ tok $x_{12}^*$ a z neho po pridaní $ss_3$ maximálny $s$—$t$ tok $x_{123}^*$ atď. Zrejme tok $x_{12\ldots p}^*$ je lexikograficky maximálny podľa vstupov.

Symetrická konštrukcia nám zasa dá lexikograficky maximálny tok $y_{12\ldots q}^*$ podľa výstupov.

Pretože toky $x_{12\ldots p}^*$ a $y_{12\ldots q}^*$ sú maximálne $s$—$t$ toky, existuje $s$—$t$ rez $(S, \bar{S})_G$ tak, že každá hrana $ij \in (S, \bar{S})_G$ je nasýtená aj jedným aj druhým tokom a na každej hrane $ij \in (\bar{S}, S)_G$ sú obidva toky nulové. Takto možno obidva toky skombinovať a získať tok $z$ definovaný takto: $z(ij) = x_{12\ldots p}^*(ij)$, ak obidva vrcholy $i, j \in S$ a $z(ij) = y_{12\ldots q}^*(ij)$ inak. Vidíme, že tok $z$ je lexikograficky maximálny súčasne podľa vstupov aj podľa výstupov. ●

## HRANOVÉ A VRCHOLOVÉ KAPACITY

Nech v základnej úlohe o maximálnom toku má navyše aj každý vrchol $j \in V(G) - \{s, t\}$ predpísanú kapacitu $w_j$ a žiadajme, aby

$$\sum_{i \in V^-(j)} x_{ij} \leq w_j .$$

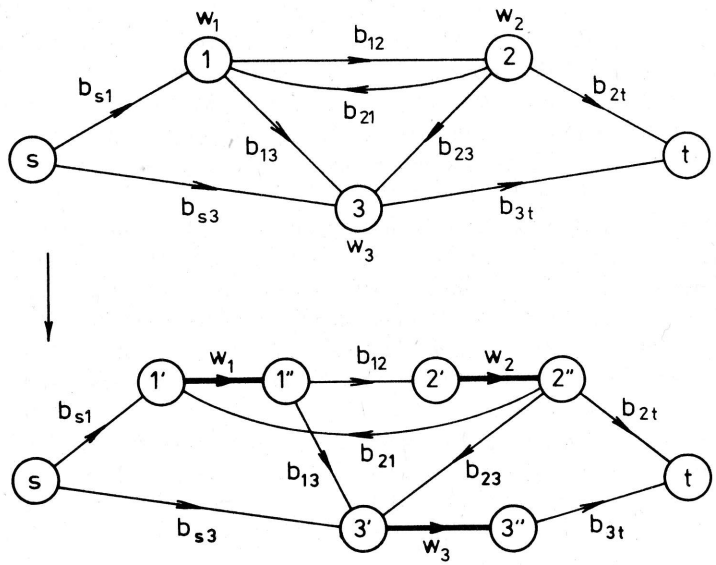
(To zodpovedá napr. obmedzenej priepustnosti cestných križovatiek.) Takúto úlohu možno previesť na základnú tak, že každý vrchol $j \neq s, t$ nahradíme dvoma vrcholmi $j'$ a $j''$, ktoré spojíme šípom $j'j''$ kapacity $w_j$. Potom všetky tie šípy, ktoré prichádzali do $j$, budú prichádzať do $j'$ a tie, čo odchádzali z $j$, budú odchádzať z $j''$. Obr. 3 ukazuje túto transformáciu.

## DOLNÉ MEDZE NA TOK

Predpokladajme, že každá hrana $ij$ digrafu má okrem kapacity $b_{ij}$ priradené ešte číslo $a_{ij}$ — dolnú medzu na tok. Takto namiesto ohraničenia $0 \leq x_{ij} \leq b_{ij}$ žiadame, aby $a_{ij} \leq x_{ij} \leq b_{ij}$.

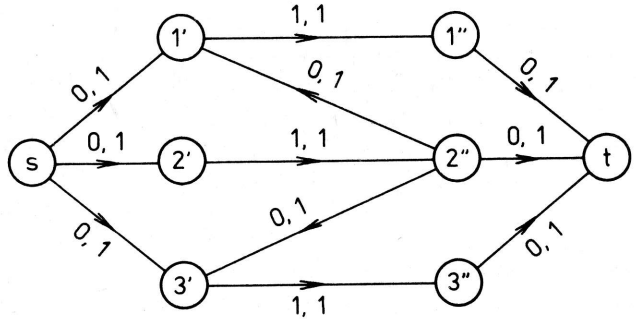







Obr. 3. Odstránenie vrcholových kapacít.

Ak boli všetky $a_{ij} = 0$, tak vždy existoval nejaký tok (napr. nulový), ale teraz sme postavení aj pred úlohu nájsť nejaký tok, čo formulujeme ako úlohu nájsť *prípustný tok*. Okrem úlohy nájsť maximálny tok tu má zmysel aj úloha nájsť *minimálny tok*, ako vidieť z nasledujúceho problému, ktorý riešili Dantzig a Fulkerson (1954).

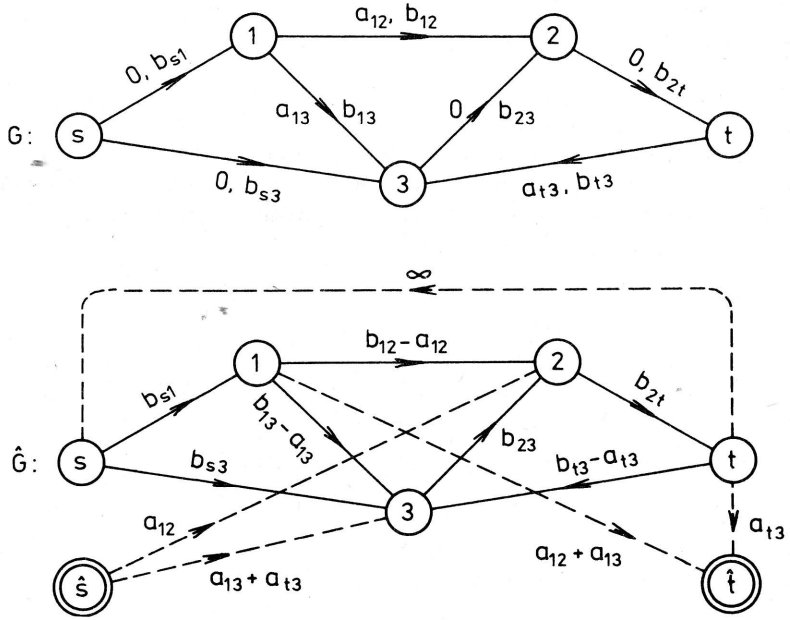



Obr. 4. K úlohe pokrytia letov najmenším počtom lietadiel.

Daný rozvrh $p$ letov treba realizovať s najmenším počtom lietadiel (rovnakých). K tejto úlohe priraďme digraf, kde každému letu $i$ zodpovedá hrana $i'i''$. Okrem toho, ak lietadlo po lete $i$ môže uskutočniť let $j$, digraf bude obsahovať šíp $i''j'$. Konečne pridáme vrchol $s$ a šípy $si'$ a vrchol $t$ a šípy $i''t$ (pozri príklad na obr. 4). Ak všetky kapacity hrán sú jednotkové a dolné medze na tok nulové okrem hrán $i'i''$, kde sú jednotky, tak veľkosť minimálneho celočíselného $s$—$t$ toku je rovná minimálnemu počtu potrebných lietadiel.

Prípustný tok možno nájsť pretransformovaním na úlohu s nulovými dolnými medzami: K danému digrafu $G$ zostrojíme digraf $\hat{G}$ tak, že pridáme šíp $ts$ nekonečnej kapacity, nový vstup $\hat{s}$, nový výstup $\hat{t}$ a pre každý šíp $ij$, kde $a_{ij}>0$ zmeníme kapacitu $b_{ij}$ na $b_{ij}-a_{ij}$ a pridáme šípy $\hat{s}j$ a $i\hat{t}$ s kapacitami $a_{ij}$: násobné šípy potom nahradíme jedným so sumárnou kapacitou. (Porovnaj obr. 5, kde v digrafe $G$ je na každej hrane $ij$ dvojica $(a_{ij}, b_{ij})$ a na každej hrane v $\hat{G}$ len kapacita hrany, lebo dolné medze sú nulové.)

Teraz je ľahko skonštruovať ku každému prípustnému $s$—$t$ toku $x$ v $G$ $\hat{s}$—$\hat{t}$ tok $y$ v $\hat{G}$ nasycujúci všetky hrany $\hat{s}j$ aj všetky hrany $i\hat{t}$ (stačí položiť $y_{ij} := x_{ij} - a_{ij}$ a zvyšok $a_{ij}$ odviesť do $\hat{t}$ a do $j$ priviesť $a_{ij}$ jednotiek z $\hat{s}$). Tok $y$ je potom




Obr. 5. Odstránenie dolných medzí.





maximálny. Obrátene, z $\hat{s}$—$\hat{t}$ toku $y$ v $\hat{G}$, ktorý nasycuje všetky hrany $\hat{s}j$ (a teda všetky hrany $i\hat{t}$), ľahko zostrojíme prípustný tok $x$ pre $G$ ($x_{ij} := y_{ij} + a_{ij}$).

Sumarizujúc vidíme, že pre nájdenie nejakého prípustného toku v $G$, stačí nájsť maximálny $\hat{s}$—$\hat{t}$ tok $y$ v $\hat{G}$: ak nasycuje všetky $\hat{s}j$ hrany, tak zostrojíme prípustný tok $x$ v $G$ ako hore; ak všetky hrany nenasycuje, tak prípustný tok v $G$ neexistuje.

Vychádzajúc z prípustného toku $x$ v $G$ možno nájsť maximálny tok upraveným algoritmom Forda a Fulkersona: na nesúhlasných šípoch rezervných polociest musí byť $x_{ij}>a_{ij}$ (namiesto $x_{ij}>0$). Analogicky možno zostrojiť algoritmus pre nájdenie minimálneho toku.

Iný spôsob hľadania maximálneho, resp. minimálneho $s$—$t$ toku sa nám ponúka, ak si uvedomíme, že po nájdení toku $y$ v $\hat{G}$ nasycujúceho hrany $\hat{s}j$, hodnota $y_{ts}$ je rovná veľkosti $s$—$t$ toku $x$ v $G$. Preto vychádzajúc z $y$ nájdeme v $\hat{G}$ maximálny $s$—$t$ tok $y^*$ nasycujúci hrany $\hat{s}j$ (nájdeme lexikograficky maximálny tok podľa vstupov $\hat{s}$, $s$ a výstupov $\hat{t}$, $t$ pričom $\hat{s}$ a $\hat{t}$ sú prioritné). Položiac $x^*_{ij} := y^*_{ij} + a_{ij}$ pre $ij \in E(G)$ získame maximálny $s$—$t$ tok v $G$ s veľkosťou $y^*_{ts} + f^+_s(y^*)$, kde $f^+_s(y^*)$ je čistý odtok z $s$ v $\hat{G}$ pri toku $y^*$.

Ak chceme nájsť minimálny $s$—$t$ tok, tak vychádzajúc z toku $y$ v $\hat{G}$, hľadáme lexikograficky maximálny tok $y^*$ podľa vstupov $\hat{s}$, $t$ a výstupov $\hat{t}$, $s$ ($\hat{s}$ a $\hat{t}$ sú prioritné), pričom kapacitu šípa $ts$ položíme rovnú $y_{ts}$. Toku $y^*$ potom zodpovedá tok $x^*$ v pôvodnom digrafe, ktorý má veľkosť $y^*_{ts} - f^+_t(y^*)$, kde $f^+_t(y^*)$ je čistý odtok z $t$ toku $y^*$ v $\hat{G}$. ●

Z uvedených postupov bezprostredne vyplýva:

**Veta 1.** Ak sú všetky kapacity aj dolné medze šípov digrafu celočíselné a existuje prípustný tok $x$, tak existuje celočíselný maximálny tok $x_1$ a celočíselný minimálny tok $x_2$. Naviac, zložitosť algoritmov pre $x$, $x_1$ a $x_2$ je taká istá ako pre základnú úlohu o maximálnom toku.

## TOKY PO REBRÁCH

Toky si možno celkom dobre predstaviť aj po rebrách. Pritom tok môže tiecť po rebre v hociktorom, ale iba v jednom smere a nesmie prekročiť kapacitu rebra. Veľkosť $f(x)$ nejakého $s$—$t$ toku $x$ je rovná čistému odtoku z $s$, a teda čistému prítoku do $t$.

K úlohe nájsť maximálny tok v grafe vedie napr. úloha: Obojsmerné ulice mesta treba zjednosmerniť tak, aby bol zabezpečený maximálny dopravný tok z miesta $s$ do miesta $t$.

Aj tu stačí každé rebro $e=[i, j]$ s kapacitou $b[i, j]$ nahradiť dvojicou protišípov $ij$ a $ji$ s kapacitami $b_{ij} = b_{ji} = b[i, j]$. Potom po nájdení toku $x$ v digrafe stačí tento tok upraviť a máme tok v pôvodnom grafe. Totiž, ak $x_{ij}$ aj $x_{ji}$ sú kladné, tak

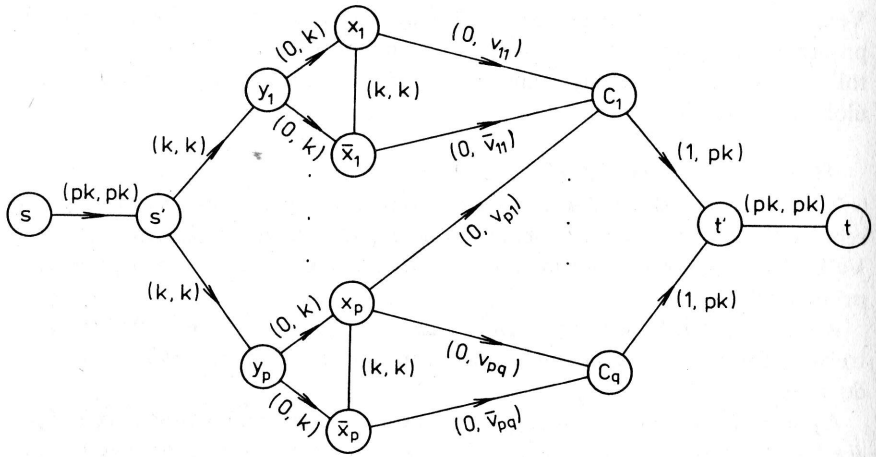
vyrobíme nový tok $x'$ definovaný takto: Nech $z(e) = \min\{x_{ij}, x_{ji}\}$: potom položíme $x'_{ij} = x_{ij} - z(e)$ a $x'_{ji} = x_{ji} - z(e)$ (veľkosť toku sa týmto nezmení).

Avšak úplne iná je situácia, ak pre každé rebro máme danú aj dolnú medzu na tok [Itai (1978)]:

**Veta 2.** Problém zistiť (nie nutne celočíselný) prípustný tok v grafe, kde každé rebro má kapacitu a dolnú medzu na tok, je NP-ťažký, dokonca pri celočíselných dátach.

DÔKAZ. Najprv transformujeme problém splniteľnosti boolovskej konjunktívnej normálnej formy $F$ na problém splniteľnosti formy s rovnakým počtom výskytov každého symbolu. Nech sa symbol $x_i$ vyskytuje celkove $k_i$-krát a symbol $\bar{x}_i$ $\bar{k}_i$-krát. Položme $k := 1 + \max\{k_i, \bar{k}_i | 1 \leq i \leq p\}$, kde $p$ je počet premenných v $F$. Utvorme konjunktívnu normálnu formu $\tilde{F} = F \wedge D_1 \wedge D_2 \wedge \ldots \wedge D_p$, kde $D_i :=$ $(x_i \vee \ldots \vee x_i \vee \bar{x}_i \vee \ldots \vee \bar{x}_i)$, pričom $x_i$ je v $D_i$ $(k - k_i)$-krát a $\bar{x}_i$ $(k - \bar{k}_i)$-krát Zrejme každý symbol sa vyskytuje v $\tilde{F}$ presne $k$-krát. Forma $\tilde{F}$ je zrejme splniteľná práve vtedy, keď $F$ je splniteľná.

Nech $\tilde{F} = C_1 \wedge C_2 \wedge \ldots \wedge C_q$ a každý zo symbolov $x_1, \bar{x}_1, \ldots, x_p, \bar{x}_p$ sa v $\tilde{F}$ vyskytuje práve $k$-krát. Problém splniteľnosti $\tilde{F}$ pretransformujeme najprv na problém prípustného toku v migrafe a ten potom na náš problém. K $\tilde{F}$ priraďme migraf načrtnutý na obr. 6, kde na každej hrane je dvojica: (dolná medza na tok, kapacita). Číslo $v_{ij}$ udáva počet výskytov symbolu $x_i$ v $C_j$, analogicky slúži $\bar{v}_{ij}$ pre $\bar{x}_i$.




Obr. 6. K transformácii splniteľnosti na prípustný tok.



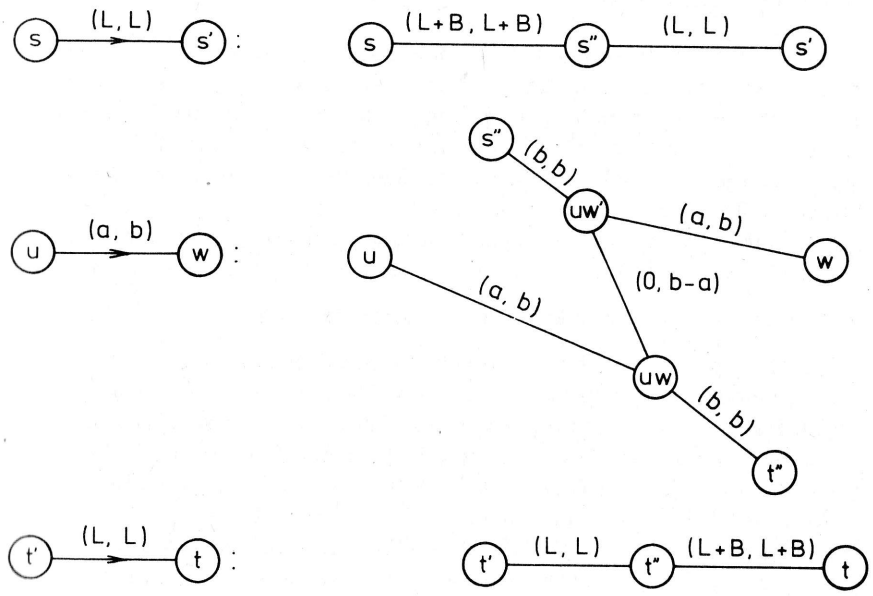


Ak existuje prípustný tok, tak každé rebro $x_i\bar{x}_i$ musí byť nasýtené. Preto celý tok z $y_i$ prechádza alebo do $\bar{x}_i$ a potom do $x_i$ (vtedy $x_i$ označíme ako pravdivé), alebo do $x_i$ a potom do $\bar{x}_i$ (vtedy je $\bar{x}_i$ pravdivé). V prvom prípade každý šíp $x_iC_j$ sa nasýti: inak bude šíp $\bar{x}_iC_j$ nasýtený. Keďže dolná medza na šípe $C_jt$ je 1, tak pre každé $C_j$ existuje prichádzajúci šíp s kladným tokom (čo označuje pravdivosť $C_j$). Teda forma $\tilde{F}$ je splniteľná.

Ak je forma $\tilde{F}$ splniteľná, tak prípustný $s$—$t$ tok zrejme existuje.

Teraz ešte ukážeme, ako nahradiť šípy rebrami, a tým previesť problém z migrafu na graf. Pridajme dva nové vrcholy $s''$ a $t''$. Šíp (s, s') nahradíme rebrami $[s, s'']$ a $[s'', s']$ a analogicky šíp $(t', t)$, ako ukazuje obr. 7, kde $B$ je súčet kapacít všetkých šípov migrafu okrem $(s, s')$ a $(t', t)$, t. j. $B = (5 + q)pk$ a $L = pk$ je požadovaná veľkosť toku. Ostatné šípy $(u, w)$ nahradíme tak, ako ukazuje obr. 7, kde $uw$ a $uw'$ sú nové vrcholy pre šíp $(u, w)$.

Nech $x$ je prípustný $s$—$t$ tok v získanom grafe. Môžeme predpokladať, že $x(s, s'') \geq 0$, a teda potom aj $x(t'', t) \geq 0$ (inak tok na každom rebre obrátime). Vrchol $t$ susedí iba s $t''$, a preto $x(t', t'') = L$ a $x(uw, t'') = b$. Takto $x(u, w) \geq 0$, a preto $a \leq x(u, uw) \leq b$, $0 \leq x(uw', uw) = b - x(u, uw) \leq b - a$. Analogicky




Obr. 7. Nahradenie šípov rebrami.

vidíme, že $x(s'', uw') = b$ a $x(uw', w) = x(u, uw) \geq a$, a preto celkový tok z $u$ do $w$ je nezáporný a spĺňa ohraničenia na tok po šípe v migrafe.
K danému toku na šípe ľahko zostrojíme tok na rebrách. ■

Hoci nevieme dobre hľadať ani celočíselný tok, máme aspoň nasledujúcu existenčnú vetu.

**Veta 3.** Ak v grafe $G$ s celočíselnými kapacitami a dolnými medzami rebier existuje prípustný tok, tak v ňom existuje celočíselný maximálny tok aj celočíselný minimálny tok.

DÔKAZ. Ku každému toku $y$ v $G$ možno priradiť orientáciu $G_y$ grafu $G$ tak, že hranu zorientujeme v smere toku po nej: takto $y$ je tokom v digrafe $G_y$. Obrátene, každý tok v $G_y$ je aj tokom v $G$. Existuje len konečne mnoho orientácií grafu $G$ a v každej z nich podľa vety 1 existuje celočíselný maximálny tok aj celočíselný minimálny tok, kedykoľvek existuje prípustný tok. ■

## 6.D
## Najlacnejší maximálny tok

Tokmi možno reprezentovať „tok" produktov od výrobcov k spotrebiteľom, avšak z ekonomického hľadiska je dôležité uskutočniť prepravu čo najlacnejšie. Totiž niektoré prepravné linky môžu mať dostatočne veľkú kapacitu, ale drahú prepravu (napr. letecké linky). Preto zavádzame na každú hranu $ij$ aj cenu $c_{ij}$ prepravy za jednotku toku. Úlohou je minimalizovať celkové prepravné náklady. Presnejšie, budeme uvažovať takúto úlohu:

Nech každý šíp $ij$ digrafu $G$ má kapacitu $b_{ij} \geq 0$ a cenu $c_{ij}$ (môže byť aj záporná). Pre daný vstup $s$ a výstup $t$ treba nájsť *najlacnejší maximálny* $s$—$t$ tok, t. j. taký maximálny $s$—$t$ tok $x$, pre ktorý je $\sum_{ij \in E(G)} x_{ij} c_{ij}$ minimálne.

Pre túto špeciálnu úlohu lineárneho programovania existuje veľmi mnoho metód, najmä pre jej dopravné verzie. Tzv. klasickú dopravnú úlohu, o ktorej pojednávame neskôr, uvažuje už Hitchcock (1941) a odvtedy sa tejto problematike venovalo mnoho autorov: Dantzig (1951); Flood (1953); Ford a Fulkerson (1956a); Orden (1956); Berge (1958); Bílý, Fiedler a Nožička (1958); Fiedler a Nožička (1961); Ford a Fulkerson (1962) a i. Prehľad viacerých metód nájde čitateľ napr. v [Goľštejn, Judin (1969)]. Väčšina prác využíva dualitu z lineárneho programovania (takýto postup budeme neskôr ilustrovať na Hitchcockovej dopravnej úlohe). V práci [Bílý et al.] je podrobne rozpracovaný grafový postup.





### ALGORITMUS POLOCYKLOV A POLOCIEST

Pre riešenie uvedenej úlohy nájsť najlacnejší maximálny tok dáme metódu, ktorá je kombináciou algoritmov z prác [Klein (1967)] a [Busacker, Gowen (1961)]. V prvej z nich sa najprv nájde maximálny tok, ktorého cena sa potom zmenšuje zmenou toku na polocykloch zápornej ceny; tento prístup možno nazvať primárnym. V druhej práci je duálny prístup: Daný optimálny tok veľkosti $h$ sa zväčšuje po najlacnejších $s$—$t$ polocestách.

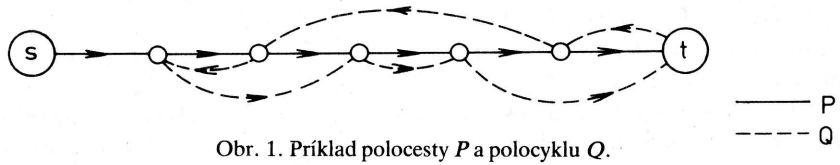
*Cena* polocyklu alebo polocesty $P$ je definovaná ako súčet cien súhlasných šípov mínus súčet cien nesúhlasných šípov. (Cena predstavuje čistú zmenu ceny toku, ak na $P$ zväčšíme tok o jednotku.) Tok $x$ môžeme zväčšiť na polocykle alebo poloceste o $\delta > 0$, ak $x(e) + \delta \leq b(e)$ na každom súhlasnom šípe $e$ a ak $x(e) - \delta \geq 0$ na každom nesúhlasnom šípe $e$. Nasledujúca veta sa explicitne alebo implicitne objavuje vo viacerých prácach. Časť (a) možno nájsť v [Busacker, Saaty (1965)] a časť (b) v [Jewell (1958)] a v [Busacker, Gowen (1961)].

**Veta 1.** Nech $x$ je tok veľkosti $f(x) = h$. Potom platí:
(a) Tok $x$ má minimálnu cenu (spomedzi tokov veľkosti $h$) práve vtedy, keď $x$ nemožno zväčšiť na polocykle zápornej ceny.
(b) Ak $x$ je tok minimálnej ceny a zväčšíme ho o $\delta$ po najlacnejšej $s$—$t$ poloceste $P$, tak získame tok $x'$ minimálnej ceny (spomedzi tokov veľkosti $h + \delta$).

DÔKAZ. (a) ($\Rightarrow$) Toto tvrdenie je zrejmé. ($\Leftarrow$) Nech existuje tok $y$ veľkosti $h$, ale menšej ceny, ako má $x$. Uvažujme rozdiel $z^0 = y - x$. Funkcia $z^0$ spĺňa podmienky kontinuity v každom vrchole (aj v $s$ a $t$), avšak nemusí byť tokom, lebo niekde môže „tiecť" proti šípke. Nech $D^0$ je digraf obsahujúci všetky vrcholy základného digrafu a $ij \in E(D^0)$ práve vtedy, keď $z^0_{ij} > 0$, alebo keď $z^0_{ji} < 0$. V $D^0$ je $z^0$ už tokom s nulovou veľkosťou $f(z^0)$ (tzv. *cirkulácia*), a teda $z^0$ možno rozložiť do cyklov digrafu $D^0$. Skutočne, $D^0$ obsahuje nejaký cyklus $C_1$: nech $\delta_1 = \min \{z^0_{ij} | ij \in C_1\}$. Ak zmenšíme tok $z^0$ na $C_1$ o $\delta_1$, získame tok $z^1$. Nech $D^1$ obsahuje len tie šípy $ij$ digrafu $D^0$, kde $z^1_{ij} > 0$ ($D^1$ má aspoň o jeden šíp menej ako $D^0$). Ak $E(D^1) \neq \emptyset$, tak v $D^1$ nájdeme cyklus $C_2$, na ktorom možno $z^1$ zmenšiť o nejaké $\delta_2$ atď.

Takto vidíme, že $z^0$ je súčtom tokov na cykloch $C_1, C_2, \ldots, C_k$. Každý z týchto cyklov $C_i$ zodpovedá v pôvodnom digrafe polocyklu $Q_i$, na ktorom možno zmeniť tok $x$ o $\delta_i$. Keďže $x + z^0 = y$ a tok $y$ má menšiu cenu ako $x$, tak niektorý z polocyklov $Q_i$ má zápornú cenu, čo je spor s predpokladom. Teda $x$ má minimálnu cenu.

(b) Podľa prvej časti vety stačí dokázať, že $x'$ nemožno zväčšiť na polocykle $Q$ zápornej ceny. Keďže pre $x$ je to pravda, tak $Q$ a polocesta $P$ majú aspoň jednu




Obr. 1. Príklad polocesty $P$ a polocyklu $Q$.

spoločnú hranu $e$, pričom $P$ prechádza cez $e$ jedným a $Q$ druhým smerom. Na obr. 1 sú dve také hrany: orientácia hrán nie je vyznačená: šípky udávajú smer prechodu polocesty $P$ a polocyklu $Q$ hranami. Polocestu $P$ spolu s polocyklom $Q$ a vyznačenými smermi môžeme pokladať za multidigraf $H$, v ktorom $P$ a $Q$ sú hranovo disjunktné, $\deg^+_H(s) = \deg^-_H(s)+1$, $\deg^-_H(t) = \deg^+_H(t)+1$ a $\deg^+_H(u) = \deg^-_H(u)=2$ pre každé $u \neq s, t$.

Sumárna cena $c(H)$ hrán v $H$ je rovná $c(P)+c(Q)$. Keďže $c(Q)<0$, tak $c(H)<c(P)$. Každý 2-cyklus, ktorý zodpovedá jedinej hrane v $G$ ($P$ a $Q$ ňou prechádzajú v rôznych smeroch) prispieva k $c(H)$ nulou, preto po vynechaní hrán takýchto 2-cyklov z $H$, získame multidigraf $H'$ s $c(H') = c(H)$. Ak $H'$ obsahuje cyklus $Z$, tak tento zodpovedá polocyklu v $G$, na ktorom možno tok $x$ zväčšiť, a preto $c(Z) \geqslant 0$. Po vynechaní hrán takýchto cyklov z $H'$ získame $H''$ s $c(H'') \leqslant c(H') = c(H) < c(P)$. Avšak $H''$ pozostáva už len z nejakej $s$—$t$ cesty $D$ a izolovaných vrcholov. $D$ zodpovedá rezervnej $s$—$t$ poloceste a $c(D) = c(H'') < c(P)$, čo je spor s voľbou $P$. ■

Rezervné polocykly zápornej ceny a rezervné $s$—$t$ polocesty minimálnej ceny môžeme zrejme hľadať pomocou metód pre najkratšie cesty, buď priamo v $G$, resp. prehľadnejšie v pomocnom kompletnom digrafe $\hat{G}$, v ktorom dĺžky jednotlivých šípov $(i, j)$ sú dané takto:

$$\hat{a}_{ij} = \begin{cases} c_{ij}, & \text{ak } x_{ij} < b_{ij},\ x_{ji} = 0: \\ -c_{ji}, & \text{ak } x_{ij} = b_{ij},\ x_{ji} > 0: \\ \min\{c_{ij}, -c_{ji}\}, & \text{ak } x_{ij} < b_{ij},\ x_{ji} > 0: \\ +\infty, & \text{inak.} \end{cases}$$

Potom najkratšia $s$—$t$ cesta v $\hat{G}$ zodpovedá najlacnejšej rezervnej $s$—$t$ poloceste v $G$ pre tok $x$. Záporným cyklom v $\hat{G}$ zodpovedajú rezervné polocykly zápornej ceny.

Na základe vety 1 ihneď máme:

**Algoritmus**

KROK 0: Nájdeme nejaký tok (nulový alebo aj maximálny).

KROK 1 *(zmena toku po záporných polocykloch):*





1.1: V príslušnom digrafe $\hat{G}$ hľadáme záporný cyklus. Ak neexistuje, ideme na krok 2.

1.2: Zmeníme tok na zápornom polocykle, čím získame lacnejší tok rovnakej veľkosti. Ideme na 1.1.

KROK 2 *(zväčšovanie toku po najlacnejších polocestách):*

2.1: V digrafe $\hat{G}$ hľadáme najkratšiu $s$—$t$ cestu. Ak žiadna neexistuje, STOP (okamžitý tok je najlacnejší maximálny tok).

2.2: Tok zväčšíme po nájdenej najlacnejšej rezervnej $s$—$t$ poloceste. Ideme na 2.1. ●

**Veta 2.** Ak v úlohe nájsť najlacnejší maximálny tok sú všetky kapacity celočíselné, tak (bez ohľadu na ceny) existuje taký najlacnejší maximálny tok, ktorý je celočíselný. Naviac, predchádzajúcim algoritmom vieme taký tok nájsť.

DÔKAZ. Ak v kroku 0 vyjdeme z celočíselného toku, tak každá zmena toku v krokoch 1 alebo 2 bude celočíselná. Celočíselných tokov je len konečne mnoho a žiaden sa v algoritme nevyskytuje 2-krát, lebo v kroku 1 sa vždy prechádza k lacnejším tokom a v kroku 2 zasa k väčším tokom. ■

Týmto je zaručená konečnosť algoritmu aj v prípade racionálnych kapacít, čo v praxi stačí. K zložitosti algoritmu sa dostaneme neskôr.

## DOPRAVNÁ ÚLOHA

Nech je daný digraf $G$, v ktorom každá hrana $ij$ má predpísanú kapacitu $b_{ij}$ a cenu $c_{ij}$ za jednotku toku. Nech $G$ má vstupy $s_1, \ldots, s_p$ a výstupy $t_1, \ldots, t_q$ s predpísanými požiadavkami $a_1, \ldots, a_p$, resp. $b_1, \ldots, b_q$. Úlohou je prepraviť „produkt“ zo vstupov na výstupy v požadovaných množstvách a pri minimálnych nákladoch. Inými slovami, v $G$ treba nájsť tok $x$ zo vstupov do výstupov taký, že čistý odtok $f^+_{s_i}(x) = a_i$ a čistý prítok $f^-_{t_j}(x) = b_j$ a pritom $\Sigma\, c_{ij}x_{ij}$ je minimálne.

Takáto úloha sa nazýva *dopravná*. (Iné názvy: sieťová dopravná úloha, úloha dopravy okľukou.)

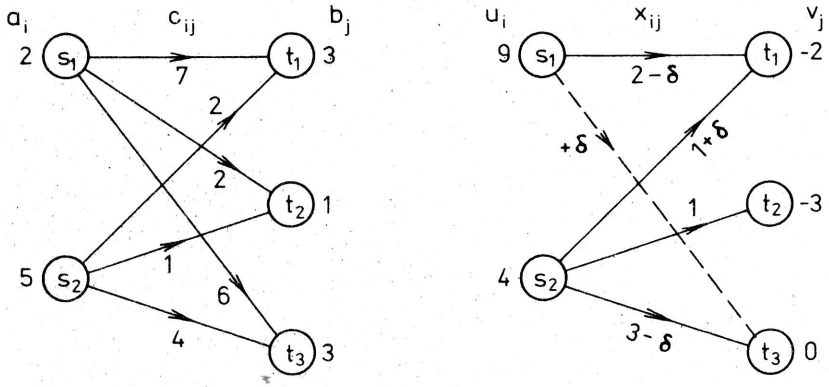
Analogicky ako v 6.C možno túto úlohu previesť na úlohu o najlacnejšom maximálnom $s$—$t$ toku, ak ku $G$ pridáme vstup $s$, výstup $t$ (pôvodné vstupy a výstupy sa stanú prechodovými vrcholmi), hrany $ss_i$ kapacity $a_i$ a ceny 0 a hrany $t_jt$ kapacity $b_j$ a ceny 0. Je zrejmé, že dopravná úloha má prípustné riešenie práve vtedy, keď v novom digrafe maximálny $s$—$t$ tok nasycuje všetky hrany $ss_i$ a $t_jt$. Potom reštrikcia najlacnejšieho maximálneho $s$—$t$ toku na pôvodný digraf je riešením dopravnej úlohy.

Medzi najznámejšie špeciálne dopravné úlohy patrí tzv. Hitchcockova

dopravná úloha (iný názov: klasická dopravná úloha). Hitchcock (1941) formuloval a riešil dopravnú úlohu, kde $G$ je bipartitný digraf s $V(G) = S \cup T$ a $E(G) = \{s_i t_j \mid s_i \in S,\ t_j \in T\}$ a všetky kapacity hrán sú nekonečné. Pre túto špeciálnu úlohu existuje mnoho algoritmov, a tu uvedieme simplexovú metódu, ktorú vypracoval Dantzig (1951), ako špecializáciu primárnej simplexovej metódy známej z lineárneho programovania. Nasledujúci výklad je založený na teórii duality (veta 2.B.1).

Nech $x_{ij}$ označuje množstvo prepravované z $s_i$ do $t_j$. Potom úlohu môžeme zapísať takto (porovnaj obr. 2):

$$\begin{cases} \text{(a)} \quad x_{i1} + x_{i2} + \ldots + x_{iq} = a_i \qquad (i = 1, 2, \ldots, p): \\ \text{(b)} \quad x_{1j} + x_{2j} + \ldots + x_{pj} = b_j \qquad (j = 1, 2, \ldots, q): \\ \qquad x_{ij} \geq 0 \quad (i = 1, 2, \ldots, p: \ j = 1, 2, \ldots, q); \\ \qquad \text{minimalizovať } \sum_{i,j} c_{ij} x_{ij}. \end{cases}$$




Obr. 2. Príklad Hitchcockovej dopravnej úlohy; kostra a zmena riešenia $((\hat{i}, \hat{j}) = (1, 3), \delta = 2, (\bar{i}, \bar{j}) = (1, 1))$.

K ohraničeniam (a) priraďme premenné $u_i$ a k (b) $v_j$. Potom duálna úloha je takáto:

$$\begin{cases} u_i + v_j \leq c_{ij} \quad (i = 1, 2, \ldots, p: \ j = 1, 2, \ldots, q): \\ \text{maximalizovať } \left( \sum_i a_i u_i + \sum_j b_j v_j \right). \end{cases}$$

Prípustné riešenia $x$ a $(u, v)$ sú optimálne práve vtedy, keď sú splnené podmienky ortogonality:





$$(*) \qquad (u_i + v_j - c_{ij}) x_{ij} = 0 \quad (i = 1, 2, \ldots, p: \ j = 1, 2, \ldots, q).$$

Postup bude takýto:

(1) Nájdeme prípustné riešenie, ktoré bude zodpovedať nejakej kostre digrafu $G$: Na hranách kostry je $x_{ij} \geq 0$, ale mimo kostry je vždy $x_{ij} = 0$. Takéto riešenie $x$ sa nazýva *bázické riešenie* a možno ho nájsť postupným nasycovaním vrcholov: Zvolíme dvojicu $(i, j)$ a položíme $x_{ij} = \{\min a_i, b_j\}$ a do kostry berieme hranu $s_i t_j$. Tým sa niektorý z vrcholov $s_i$, $t_j$ nasýti, a preto ho z digrafu vynecháme. Druhý z vrcholov vždy ponecháme a jeho požiadavku zmenšíme o $x_{ij}$. Potom postup opakujeme na redukovanom digrafe atď., až pokiaľ má digraf hrany. Takto sme získali kostru a príslušné bázické riešenie. (Prvý vynechaný vrchol bude koncový, a teda indukciou vidíme, že získame kostru.)

(2) Ak máme nejakú kostru $T$ a príslušné bázické riešenie $x$, tak nájdeme $u_i$ a $v_j$ spĺňajúce podmienky ortogonality $(*)$. Keďže na hranách kostry môžu byť všetky $x_{ij} > 0$, musíme určiť $u_i$ a $v_j$ tak, aby $u_i + v_j - c_{ij} = 0$. To sa docieli tým, že zvolíme napr. $v_q = 0$ a ostatné hodnoty $u_i$ a $v_j$ už vypočítame postupne cez hrany kostry $T$. (Výpočet je jednoznačný, lebo v $T$ existuje jediná $t_q$—$s_i$, resp. $t_q$—$t_j$ polocesta.)

(3) Ak pre všetky $i$ a $j$ máme $u_i + v_j \leq c_{ij}$, t. j. ak $(u, v)$ je prípustné riešenie duálnej úlohy, tak (keďže podmienky $(*)$ sú splnené) $x$ je optimálnym riešením primárnej a $(u, v)$ duálnej úlohy.

(4) Ak existuje $(\hat{i}, \hat{j})$ tak, že $u_{\hat{i}} + v_{\hat{j}} > c_{\hat{i}\hat{j}}$, zmeníme riešenie $x$ a (v súlade so simplexovou metódou) utvoríme novú kostru $T'$ obsahujúcu hranu $s_{\hat{i}} t_{\hat{j}}$. Nech $C$ je (jediný) polocyklus v $T + s_{\hat{i}} t_{\hat{j}}$ uvažovaný tak, že šíp $s_{\hat{i}} t_{\hat{j}}$ je v ňom súhlasný. Vidíme, že ak hodnoty $x_{ij}$ na súhlasných šípoch polocyklu zväčšíme o $\delta \geq 0$ a na nesúhlasných šípoch zmenšíme o $\delta$, tak všetky požiadavky $a_i$, $b_j$ budú splnené. Naviac, ak zvolíme $\delta$ maximálne možné a také, aby $x_{ij} - \delta \geq 0$ na všetkých nesúhlasných šípoch, získame nové prípustné riešenie $x'$: $x'_{ij} = x_{ij} + \delta$ na súhlasných šípoch, $x'_{ij} = x_{ij} - \delta$ na nesúhlasných šípoch polocyklu $C$ a $x'_{ij} = x_{ij}$ inak. Týmto aspoň na jednom nesúhlasnom šípe $(\bar{i}, \bar{j}) \in C$ máme $x'_{\bar{i}\bar{j}} = 0$. Položiac $T' = T + s_{\hat{i}} t_{\hat{j}} - s_{\bar{i}} t_{\bar{j}}$, získame novú kostru $T'$, takže $x'$ je nové bázické riešenie a ideme na krok (2). ●

Dôvod na zavedenie hrany $s_i t_j$ do kostry je ten, že cena riešenia poklesne o $\delta(u_i + v_j - c_{ij})$, čo pri $\delta > 0$ je skutočný pokles účelovej funkcie v dopravnej úlohe (6.Z.10).

Každej kostre zodpovedá jediné bázické riešenie. Skutočne: na každej koncovej hrane kostry je bázické riešenie jednoznačne určené (je rovné požiadavke koncového vrcholu) a tvrdenie vyplýva indukciou. Preto ak vždy bude $\delta > 0$, tak žiadne bázické riešenie, a teda ani žiadna kostra, sa nemôže zopakovať. Keďže

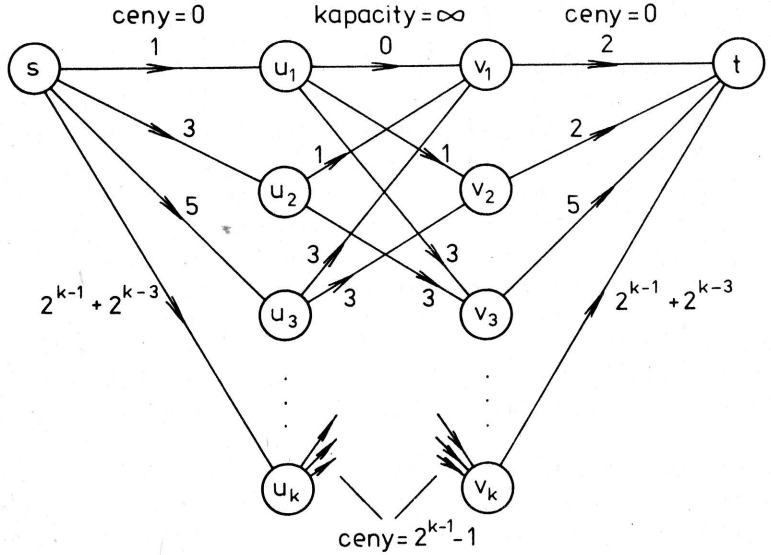
kostier je len konečne mnoho, tak z toho vyplýva konečnosť algoritmu. Žiaľ, ak sa vyskytne $\delta = 0$, tak môže vzniknúť zacyklenie. Taký príklad možno nájsť v [Gass (1964)], hoci v praktických úlohách sa zacyklenie nevyskytlo. Teoreticky sa možno poistiť proti zacykleniu nepatrným pozmenením (perturbáciou) požiadaviek $a_i$ a $b_j$ (ak $\varepsilon > 0$ je dostatočne malé, tak v úlohe s požiadavkami $a_1 + \varepsilon, \ldots, a_p + \varepsilon, b_1, b_2, \ldots, b_{q-1}, b_q + p\varepsilon$ budú vždy hranám kostry zodpovedať kladné hodnoty riešenia, a preto sa zacyklenie nevyskytne).

Hoci sa všeobecná dopravná úloha zdá byť omnoho zložitejšia, predsa ju možno polynomiálne pretransformovať na Hitchcockovu dopravnú úlohu (6.Z.7).

## POZNÁMKY O ZLOŽITOSTI

Hoci sa algoritmus polocyklov a polociest v praxi osvedčil, nie je polynomiálny, ako ukázal Zadeh (1973). Tu načrtneme jeho príklad, na ktorom ukazuje, že uvedený algoritmus môže vyžadovať exponenciálne mnoho zmien toku.

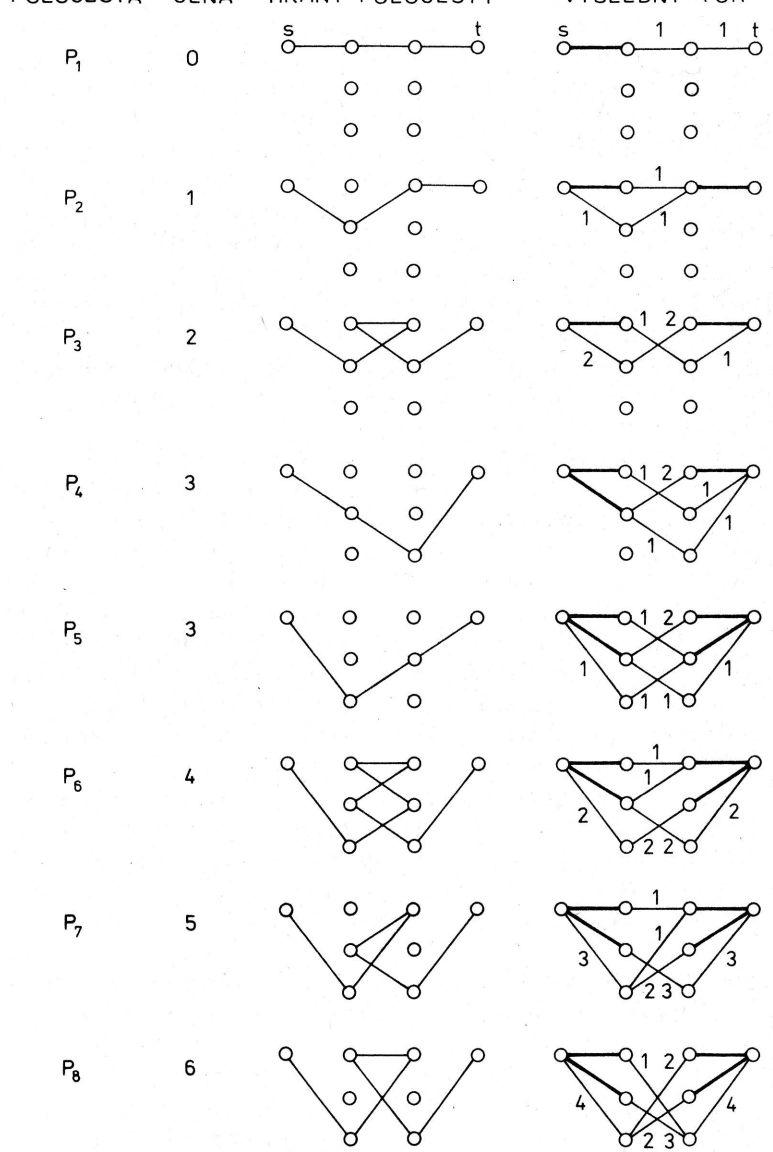
Nech $G_k$ je digraf s $2k+2$ vrcholmi $s, t, u_1, \ldots, u_k, v_1, \ldots, v_k$ a množinou šípov $\{su_i \mid 1 \leq i \leq k\} \cup \{v_j t \mid 1 \leq j \leq k\} \cup \{u_i v_j \mid 1 \leq i \leq k,\ 1 \leq j \leq k,\ i \neq j\} \cup \{u_1 v_1\}$. Kapacity a ceny sú vyznačené na obr. 3. Všetky šípy $su_i$ aj $v_j t$ majú ceny 0 a všetky šípy $u_i v_j$ majú kapacity $\infty$, (v skutočnosti stačí brať $b(u_i, v_j) = \min\{b(s, u_i), b(v_j, t)\}$). Pre $j \geq 3$ je $b(s, u_j) = b(v_j, t) = 2^{j-1} + 2^{j-3}$ a $c(u_i, v_j) = c(u_j, v_i) = 2^{j-1} - 1$, ak $i < j$.




Obr. 3. Digraf $G_k$ („exponenciálny" príklad).








Obr. 4. Postupnosť zväčšujúcich polociest v $G_3$ (nasýtené hrany sú hrubé).

Predpokladajme, že v kroku 0 algoritmu zvolíme nulový tok ako počiatočný. Keďže všetky ceny sú nezáporné, tak žiadny polocyklus zápornej ceny neexistuje, a preto ideme na krok 2, v ktorom sa bude tok zväčšovať až na maximálny tok. Postupnosť zväčšujúcich $s$—$t$ polociest $P_i$ v $G_3$, ich ceny a príslušné výsledné toky sú vyznačené na obr. 4. Tu vidíme, že maximálny tok v $G_3$ má veľkosť 8 a na nájdenie najlacnejšieho maximálneho toku sme potrebovali 8-krát zväčšovať tok. Zadeh (1973) ukazuje, že vo všeobecnosti v $G_k$ treba urobiť $2^k+2^{k-2}-2$ zväčšovaní toku po najlacnejších rezervných $s$—$t$ polocestách $P_i$, čo je exponenciálne mnoho vzhľadom na rozmer úlohy (zapísaný binárne). Postupnosť polociest $P_i$ má viaceré pekné vlastnosti, ktoré uľahčili dokázať toto tvrdenie. Napr.: (1) postupnosť cien polociest $P_i$ je neklesajúca a ak dve polocesty majú rovnakú cenu, tak majú aj rovnaký počet hrán, (2) každé zväčšovanie zväčšuje tok o 1, (3) alebo sa práve jedna hrana nasýti, alebo práve na jednej hrane sa tok stane nulovým, (4) existuje vždy práve jedna taká hrana tvaru $u_iv_j$ s nulovým tokom, ktorá je v zväčšujúcej poloceste.

Zadeh ukázal modifikácie digrafu $G_k$, ktoré slúžia ako zlé príklady pre ďalšie algoritmy. Napríklad, ak by sme v uvedenom algoritme v kroku 0 našli maximálny tok, tak v ďalšom sa uplatní len zmena toku na lacnejší cez polocykly zápornej ceny (krok 1). Vtedy stačí pridať ku $G_k$ šíp $\bar{s}s$ kapacity $2^k + 2^{k-2} - 2$ a ceny 0 a šíp $st$ kapacity $2^k + 2^{k-2}-2$ a veľmi veľkej ceny (napr. takej, aký je súčet všetkých cien ostatných hrán), kde $\bar{s}$ je nový vstup. Ak v kroku 0 nájdeme $\bar{s}$—$\bar{t}$ tok veľkosti $2^k + 2^{k-2}-2$ (maximálny), tak ho možno meniť na záporných polocykloch $C_1$, $C_2$, ..., kde $C_i = P_i ts$ a zrejme má zápornú cenu a dokonca je to najlacnejší rezervný polocyklus. Takto postupne meníme tok vždy o jednotku na polocykloch $C_1$, $C_2$, ... a podľa predchádzajúceho treba takýchto zmien $2^k + 2^{k-2}-2$. Tým sa tok na šípe $st$ stane nulovým a získame najlacnejší maximálny tok.

Aj iné metódy sú exponenciálne: primárno-duálna Fordova a Fulkersonova metóda (1962) a simplexová metóda [Dantzig (1962)]. Zadehov príklad po úprave slúži tiež ako exponenciálny príklad na simplexovú metódu riešenia Hitchcockovej dopravnej úlohy (6.Z.11). Takéto príklady možno nájsť aj v [Adeľson–Veľskij et al. (1975)], kde sa diskutujú aj ďalšie metódy (metóda potenciálov, maďarská metóda).

Edmonds a Karp (1970, 1972) však dávajú polynomiálny algoritmus pre najlacnejší maximálny tok použitím myšlienky zjemňovania stupnice v primárno-duálnej metóde. Ukazujú, že na Hitchcockov dopravný problém (s $p \leq q$) stačí $q[2+\log_2(\Sigma\ a_i/q)]$ zmien toku, ak sú všetky požiadavky $a_1, \ldots, a_p$ a $b_1, \ldots, b_q$ celočíselné. Takéto algoritmy majú však zatiaľ len teoretický význam, totiž v praktických úlohách sa klasické algoritmy správajú lepšie. Spomedzi klasických metód sa primárno-duálna metóda pokladala za najlepšiu, ale Glover, Karney





a Klingman (1974) testovaním na množstve príkladov zistili, že primárna simplexová metóda [Dantzig (1962)] je najrýchlejšia.

### NAJLACNEJŠIE ZVÄČŠENIE KAPACÍT ŠÍPOV

Predpokladajme, že v danom digrafe $G$ máme zabezpečiť $s$—$t$ tok veľkosti $f$, pričom v prípade potreby možno kapacity hrán zväčšiť. Nech šíp $ij$ má kapacitu $b_{ij}$ a nech $c_{ij}>0$ je cena každého zväčšenia kapacity o jednotku. Úlohou je minimalizovať sumárnu cenu za zväčšenie kapacít.

Túto úlohu aj nasledovnú ideu riešenia navrhol Fulkerson (1959). Môžeme predpokladať, že v $G$ existuje $s$—$t$ cesta (inak je a zostane maximálny tok len nulovej veľkosti). Ku každému šípu $ij$ pridajme nový násobný šíp $\bar{ij}$ kapacity $\infty$ a ceny $c_{ij}$: pôvodný šíp $ij$ bude mať pôvodnú kapacitu $b_{ij}$, ale cenu 0. Okrem toho pridajme nový šíp $t\hat{t}$ kapacity $f$ a ceny 0, kde $\hat{t}$ je nový výstup. Je zrejmé, že stačí nájsť najlacnejší maximálny $s$—$\hat{t}$ tok v utvorenom multidigrafe $\hat{G}$ a pôvodná úloha je vyriešená. Totiž ak potečie tok šípom $\bar{ij}$, tak už šíp $ij$ musí byť nasýtený, lebo má menšiu (nulovú) cenu. Pri počítaní nemusíme mať skutočne dva násobné šípy, ale len jeden s nelineárnou, avšak konvexnou a po častiach lineárnou cenou

$$\hat{c}_{ij}(x_{ij}) := \begin{cases} 0, & \text{ak } x_{ij} \leq b_{ij}; \\ c_{ij}(x_{ij} - b_{ij}), & \text{ak } x_{ij} > b_{ij}. \end{cases}$$

Výpočet potom prebieha simulovaním výpočtu s násobnými šípmi.

Na tej istej myšlienke možno modifikovať algoritmus polocyklov a polociest aj pre prípad, že ceny sú konvexné funkcie zložené aj z viac ako dvoch lineárnych častí (6.Z.6).

## 6.E
## Viacproduktové toky v grafoch

Nech je daný graf $G$, kde každé rebro $[u, v]$ má kapacitu $b[u, v]$ a nech je daných $k$ vstupov $s_1, s_2, \ldots, s_k$ a $k$ výstupov $t_1, t_2, \ldots, t_k$. Potom $k$-tica $(x_1, x_2, \ldots, x_k)$ sa nazýva *$k$-produktový tok* (presnejšie, $(s_1, \ldots, s_k)-(t_1, \ldots, t_k)$ tok) v grafe $G$, ak pre každé $i=1, 2, \ldots, k$ je $x_i$ jednoproduktový $s_i$—$t_i$ tok a pre každé rebro $[u, v] \in E(G)$ platí:

$$|x_1(u, v)| + \ldots + |x_k(u, v)| \leq b[u, v].$$

Poznamenajme, že ak po rebre $[u, v]$ tečie tok $x_i$ veľkosti $h$ v smere od $u$ ku $v$, tak kladieme $x_i(u, v) = h \geq 0$ a $x_i(v, u) = -h \leq 0$. Nech $f(x_i)$ označuje čistý odtok toku $x_i$ z $s_i$.

Zavedenie dolných medzí na tok je tiež možné, ale to nebudeme robiť, lebo potom dokonca pre jednoproduktový tok máme NP-ťažký problém (veta 6.C.2).

Zavádzame dve úlohy: (1) Dané sú *požiadavky* $d_1, \ldots, d_k$: treba nájsť taký $(s_1, \ldots, s_k)-(t_1, \ldots, t_k)$ tok $(x_1, \ldots, x_k)$, že pre každé $i=1, 2, \ldots, k$ je (čistý odtok z $s_i$) $f(x_i) \geq d_i$. (2) Treba maximalizovať $f(x_1) + \ldots + f(x_k)$.

V ďalšom sa sústredíme najmä na 2-produktové toky, keďže tu sa dosiahol významný úspech. Najprv riešime úlohu (2).

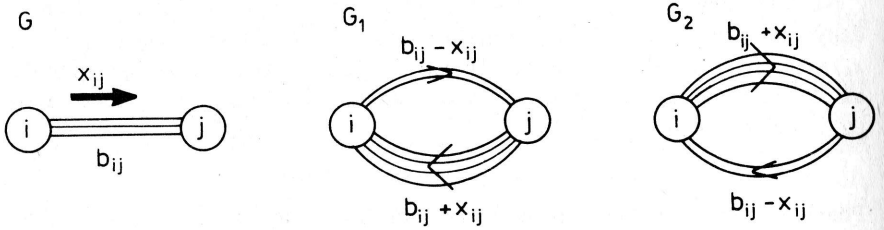
## MAXIMÁLNY 2-PRODUKTOVÝ TOK V GRAFE

Ako prvý túto úlohu vyriešil Hu (1963). Konečnosť jeho algoritmu je zaručená iba pre celočíselné kapacity rebier, avšak jeho myšlienka, zmeniť na cykle maximálny $s_1$—$t_1$ tok a umožniť väčší $s_2$—$t_2$ tok, bola základom pre neskoršie polynomiálne algoritmy [Čerkasskij (1973)], [Itai (1978)], [Sakarovitch (1973)], [Adeľson–Veľskij, Dinic, Karzanov (1975)]. V posledných dvoch prácach autori uvádzajú algoritmy pozostávajúce z riešení niekoľkých jednoproduktových úloh. Sakarovitch používa na odvodenie teóriu lineárneho programovania. Tu uvádzame grafový postup podľa [Adeľson–Veľskij et al.]

Predpokladáme, že graf $G$ s kapacitami rebier $b[i, j]$ má dané 4 rozličné vrcholy: vstupy $s_1$, $s_2$ a výstupy $t_1$, $t_2$.

Ak je daný $s_1$—$t_1$ tok $x$ v grafe $G$, tak z $G$ zostrojíme dva digrafy $G_1$, $G_2$ nahradením každého rebra $[i, j]$ dvojicou protišípov $(i, j)$ a $(j, i)$. $G_1$ má kapacity $b_1(i, j) := b[i, j] - x(i, j)$ (keďže $x(j, i) = -x(i, j)$, tak $b_1(j, i) = b[i, j] + x(i, j)$): $G_2$ má kapacity $b_2(i, j) := b_1(j, i)$ (porovnaj obr. 1, kde šírky hrán zodpovedajú kapacitám, $b[i, j] = b_{ij}$ a $x(i, j) = x_{ij}$)).

K prechodu od nejakého toku $y$ v digrafe $G_1$ alebo $G_2$ na tok v grafe $G$ je užitočné definovať funkciu $\tilde{y}(i, j) := y(i, j) - y(j, i)$.




Obr. 1. Kapacity šípov v digrafoch $G_1$ a $G_2$.





**Algoritmus**

KROK 1: V grafe $G$ nájdeme maximálny $s_1$—$t_1$ tok $x$.

KROK 2: Pomocou $G$ a $x$ zostrojíme digraf $G_1$ a určíme jeho kapacity $b_1$. Potom v $G_1$ nájdeme maximálny $s_2$—$t_2$ tok $y_1$.

KROK 3: Zostrojíme digraf $G_2$ a určíme kapacity $b_2$. Potom v $G_2$ nájdeme maximálny $s_2$—$t_2$ tok $y_2$.

KROK 4:

4(a): Ak veľkosť toku $f(y_1) \leq f(y_2)$, tak položíme $z_1 := y_1$ a v $G_2$ zmenšíme tok $y_2$ na $s_2$—$t_2$ tok $z_2$ veľkosti $f(z_2) = f(z_1)$.

4(b): Ak $f(y_1) > f(y_2)$, tak položíme $z_2 := y_2$ a v $G_1$ zmenšíme tok $y_1$ na $s_2$—$t_2$ tok $z_1$ veľkosti $f(z_1) = f(z_2)$.

KROK 5: Položíme $x_1 := x + (\tilde{z}_1 - \tilde{z}_2)/2$, $x_2 := (\tilde{z}_1 + \tilde{z}_2)/2$ a STOP: $(x_1, x_2)$ je maximálny dvojproduktový tok v $G$. ●

Dokážeme správnosť algoritmu. Najprv dokážeme, že (1) podmienky kontinuity sú splnené pre $x_1$ aj pre $x_2$. Potom overíme (2) dodržanie kapacitných ohraničení a nakoniec ukážeme (3) maximalitu dvojproduktového toku $(x_1, x_2)$.

(1) Keďže $V^+_{G_1}(i) = V^-_{G_1}(i) = V_G(i)$ pre každé $i$, tak máme:

$$\sum_{j \in V_G(i)} \tilde{z}_1(i, j) = \sum_{j \in V_G(i)} [z_1(i, j) - z_1(j, i)] =$$

$$= \sum_{j \in V^+_{G_1}(i)} z_1(i, j) - \sum_{j \in V^-_{G_1}(i)} z_1(j. i) =$$

$$= \begin{cases} 0, & \text{ak } i \neq s_2, t_2: \\ f(z_1) & \text{pre } i = s_2: \\ -f(z_1) & \text{pre } i = t_2. \end{cases}$$

Analogický výsledok platí pre $\tilde{z}_2$, takže podmienky kontinuity sú splnené. Naviac vidíme, že $f(x_1) = f(x)$, a teda $x_1$ je tiež maximálny $s_1$—$t_1$ tok v $G$.

(2) Treba ukázať, že pre ľubovoľnú hranu $[i, j] \in E(G)$ je $|x_1(i, j)| + |x_2(i, j)| \leq b_{ij}$. Vynechávajúc argument $(i, j)$, máme: $|x_1| + |x_2| \leq \max \{|x_1 + x_2|, |x_1 - x_2|\}$. Preto stačí ukázať, že

$$|x_1 + x_2| = |x + \tilde{z}_1| \leq b \text{ a } |x_1 - x_2| = |x - \tilde{z}_2| \leq b.$$

Keďže $z_1$ je tokom v $G_1$ a $z_2$ v $G_2$, tak z definície čísel $\tilde{z}_1$, $\tilde{z}_2$ máme (pozri obr. 1):

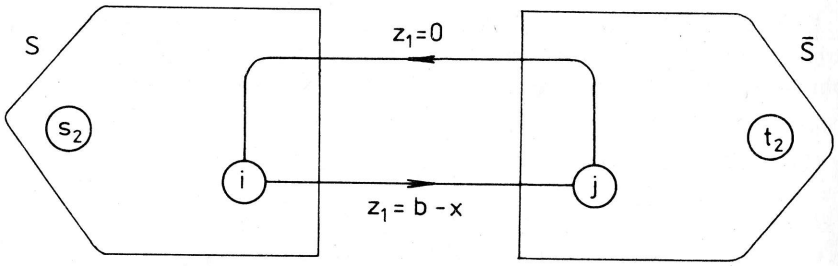
$$-b = x - (b + x) \leq x + \tilde{z}_1 \leq x + b - x = b,$$

$$-b = x-(b+x) \leq x-\tilde{z}_2 \leq x+b-x = b,$$

čo sme mali ukázať.

(3) Pre dôkaz maximality toku $(x_1, x_2)$ uvážime dva prípady, zodpovedajúce krokom 4(a) a 4(b).

(a) $f(y_1) \leq f(y_2)$. Potom $z_1 = y_1$. K maximálnemu $s_2$—$t_2$ toku $z_1$ v $G_1$ existuje minimový $s_2$—$t_2$ rez $(S, \bar{S})_{G_1}$ s kapacitou $f(z_1)$. Ako vieme, na každom šípe $ij \in (S, \bar{S})_{G_1}$ je $z_1(i, j) = b_1(i, j) = b[i, j] - x(i, j)$ a $z_1(j, i) = 0$ (pozri obr. 2). Preto pre $ij \in (S, \bar{S})_{G_1}$ máme: $x_1 + x_2 = x + \tilde{z}_1 = x + z_1 = x + b - x = b$: takto aj $|x_1| + |x_2| = b$ a $x_1, x_2 \geq 0$. Teda oba toky $x_1$, $x_2$ v $G$ smerujú z $S$ do $\bar{S}$ a spolu nasycujú všetky hrany rezu $[S, \bar{S}]_G$. Ak na všetkých hranách tohto rezu je $x_1 = 0$, tak $f(x_2) = b[S, \bar{S}]$, a preto $x_2$ je maximálny $s_2$—$t_2$ tok v $G$. Pretože v (1) sme ukázali, že $x_1$ je maximálny $s_1 - t_1$ tok v $G$, tak zrejme $(x_1, x_2)$ je maximálny dvojproduktový tok v $G$.




Obr. 2. K dôkazu správnosti algoritmu (prípad (a)).

Zostal nám prípad, keď na nejakom rebre $ij$, $i \in S$, $j \in \bar{S}$ máme $x_1(i, j) > 0$. Keďže sme ukázali, že na všetkých hranách rezu tok $x_1$ tečie len z $S$ do $\bar{S}$, tak zrejme $s_1 \in S$ a $t_1 \in \bar{S}$. Potom však $[S, \bar{S}]_G$ je rezom oddeľujúcim $\{s_1, s_2\}$ a $\{t_1, t_2\}$, a teda $(x_1, x_2)$ je maximálny tok, lebo ho nasycuje.

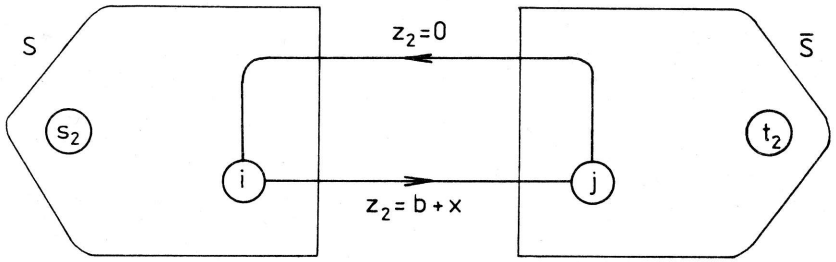
(b) $f(y_1) \geq f(y_2)$. Potom $z_2 = y_2$. Analogicky ako v prípade (a), tu existuje minimový $s_2$—$t_2$ rez $(S, \bar{S})_{G_2}$ s kapacitou $f(z_2)$ (pozri obr. 3). Pre $ij \in (S, \bar{S})_{G_2}$ máme: $x_1 - x_2 = x - \tilde{z}_2 = x - (b+x) = -b$: takto $x_1 \leq 0$ a $x_2 \geq 0$. Preto v $G$ tok $(x_1, x_2)$ nasycuje hrany rezu $[S, \bar{S}]_G$, pričom $x_1$ tečie z $\bar{S}$ do $S$ a $x_2$ z $S$ do $\bar{S}$. Ak je $x_1 = 0$ na všetkých hranách rezu, tak $x_2$ nasycuje rez, a teda je maximálny $s_2$—$t_2$ tok, čo spolu s maximalitou $s_1$—$t_1$ toku $x_1$ dáva požadované tvrdenie.

Preto nech existuje $i \in S$, $j \in \bar{S}$ tak, že $x_1(i, j) < 0$. Potom však $s_1 \in \bar{S}$, $t_1 \in S$ a $[S, \bar{S}]_G$ je rez oddeľujúci $\{s_2, t_1\}$ a $\{t_2, s_1\}$. Keďže tok $(x_1, x_2)$ ho nasycuje, tak je maximálny.

Tým je správnosť algoritmu dokázaná. ■








Obr. 3. K dôkazu správnosti algoritmu (prípad (b)).

Zložitosť algoritmu je zrejme taká, ako pri 1-produktovom toku (pozri 6.B).

V dôkaze algoritmu sme videli, že $f(x_1) = f(x)$, kde $x$ bol maximálny $s_1$—$t_1$ tok. Teda platí [Hu (1963)]:

**Veta 1.** Existuje taký maximálny 2-produktový tok v grafe (a uvedeným algoritmom ho možno nájsť), ktorý súčasne maximalizuje tok jedného predpísaného produktu.

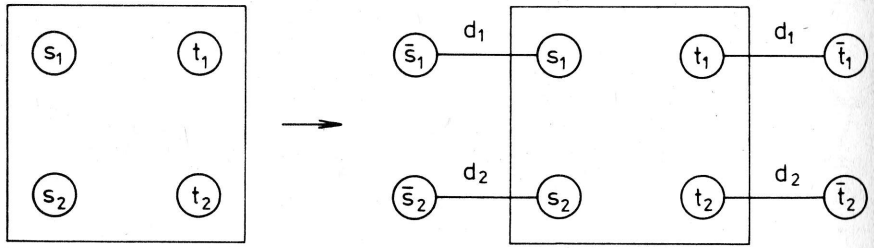
V dôkaze algoritmu sme tiež videli, že zostrojený 2-produktový tok nasycoval hrany istého rezu grafu. Keďže veľkosť žiadneho toku nemôže byť väčšia ako kapacita rezu, tak máme [Hu (1963)]:

**Veta 2.** Maximálna sumárna veľkosť 2-produktového $(s_1, s_2)$—$(t_1, t_2)$ toku v grafe je rovná minimálnej kapacite rezu oddeľujúceho $s_1$ a $t_1$ a súčasne $s_2$ a $t_2$. (Ide o $\{s_1, s_2\}$—$\{t_1, t_2\}$ rezy a $\{s_1, t_2\}$—$\{t_1, s_2\}$ rezy.)

## 2-PRODUKTOVÝ TOK V GRAFE S POŽIADAVKAMI

Túto úlohu možno previesť na úlohu nájsť maximálny 2-produktový tok, tak ako ukazuje obr. 4. Pridáme nové vstupy $\bar{s}_1$, $\bar{s}_2$, výstupy $\bar{t}_1$, $\bar{t}_2$ a rebrá $\bar{s}_1 s_1$, $t_1 \bar{t}_1$ kapacity $d_1$ a $\bar{s}_2 s_2$, $t_2 \bar{t}_2$ kapacity $d_2$, kde $d_1$, $d_2$ sú predpísané požiadavky na produkty.

Zrejme v pôvodnom grafe existuje $(s_1, s_2)$—$(t_1, t_2)$ tok $(x_1, x_2)$ s $f(x_1) \geq d_1$ a $f(x_2) \geq d_2$ práve vtedy, keď v novom grafe existuje $(\bar{s}_1, \bar{s}_2)$—$(\bar{t}_1, \bar{t}_2)$ tok $(\bar{x}_1, \bar{x}_2)$ sumárnej veľkosti $d_1 + d_2$. Takto použitím predchádzajúceho algoritmu vieme riešiť aj 2-produktovú úlohu s požiadavkami.




Obr. 4. Transformácie požiadaviek na maximalizáciu toku.

## CELOČÍSELNÉ TOKY

Ak budú všetky kapacity hrán celé párne čísla, tak toky $z_1$, $z_2$ vieme zaručiť celočíselné a na každej hrane párne. Preto v tomto prípade máme toky $x_1$ a $x_2$ celočíselné [Hu (1963)]. Vo všeobecnosti celočíselnosť kapacít nám vždy zaručuje, že hodnoty tokov na hranách budú celočíselné násobky čísla 1/2. Rothschild a Whinston (1966) dokázali, že ak sú všetky kapacity celočíselné a pri každom vrchole je súčet kapacít okolitých rebier párny, tak existuje taký maximálny 2-produktový tok, ktorý je celočíselný.

Žiaľ, vo všeobecnosti máme [Even, Itai, Shamir (1976)]:

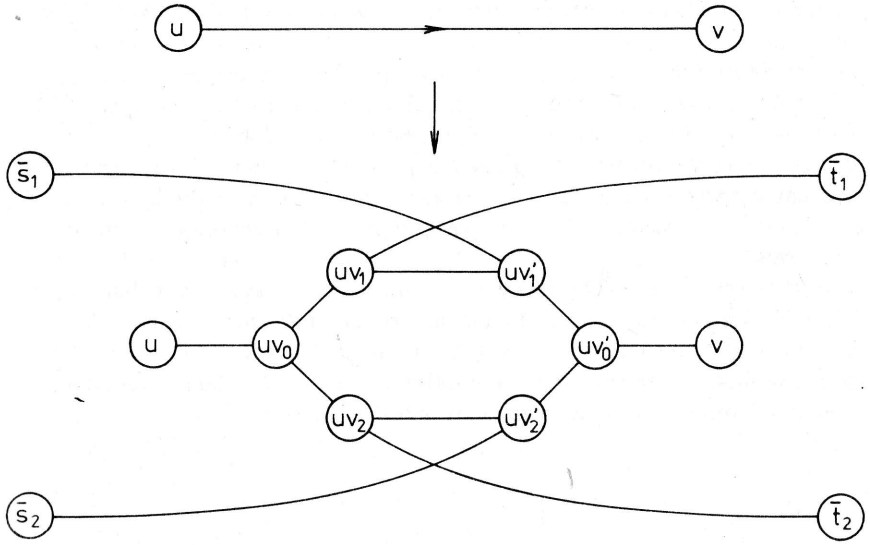
**Veta 3.** Úloha nájsť v grafe celočíselný 2-produktový tok spĺňajúci požiadavky aj úloha nájsť maximálny celočíselný 2-produktový tok sú NP-ťažké, dokonca aj v prípade jednotkových kapacít rebier.

DÔKAZ. Podľa vety 6.F.6 sú takéto úlohy v digrafoch NP-ťažké. Nech je daný digraf $G$ s jednotkovými kapacitami šípov a požiadavkami $d_1$, $d_2$. Najprv pridáme ku $G$ nové vstupy $\bar{s}_1$, $\bar{s}_2$ a výstupy $\bar{t}_1$, $\bar{t}_2$, ktoré spojíme s pôvodnými pomocou šípov $\bar{s}_1 s_1$, $t_1 \bar{t}_1$ kapacity $d_1$ a šípov $\bar{s}_2 s_2$, $t_2 \bar{t}_2$ kapacity $d_2$ (porovnaj obr. 4). Takto sme úlohu nájsť 2-produktový tok s požiadavkami $d_1$, $d_2$ pretransformovali na takúto úlohu v digrafe $\bar{G}$. Každý nový šíp kapacity $d_i$ môžeme nahradiť $d_i$ šípmi jednotkovej kapacity. Môžeme predpokladať, že $d_1$, $d_2 \leq |E(G)|$ (všetky hrany v $G$ sú jednotkovej kapacity), a preto sa rozmer úlohy zväčší len lineárne. Získaný multidigraf $\bar{\bar{G}}$ má len šípy kapacity 1 a každý z nich môžeme nahradiť štruktúrou rebier ako na obr. 5, a tým získať graf $\hat{G}$ s jednotkovými kapacitami rebier.

Ľahko vidieť, že v $\hat{G}$ existuje celočíselný 2-produktový tok spĺňajúci požiadavky $\hat{d}_i := d_i + |E(\bar{\bar{G}})| (i = 1, 2)$ práve vtedy, keď v $\bar{\bar{G}}$ existuje taký tok spĺňajúci požiadavky $d_i$ (rebrá $[\bar{s}_i, uv_i']$, $[uv_i, \bar{t}_i]$ musia byť vždy nasýtené produktom $i$, smer tokov na ostatných rebrách je týmto determinovaný).





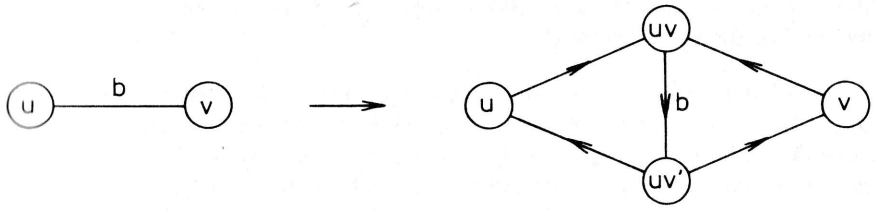



Obr. 5. Nahradenie šípa rebrami.

Posledná podmienka je ekvivalentná s tým, že v $\hat{G}$ existuje celočíselný 2-produktový tok sumárnej veľkosti $\hat{d}_1 + \hat{d}_2$. ■

## TRANSFORMÁCIA GRAFOV NA DIGRAFY

V časti 6.C sme videli, že rebro možno nahradiť dvojicou protišípov, ak ide o jednoproduktový tok. Pre viac produktov takáto substitúcia nevyhovuje. Vždy však možno nahrádzať podľa obr. 6, kde rebro $[u, v]$ kapacity $b$ nahrádzame štruktúrou piatich šípov, z ktorých šíp $(uv, uv')$ má kapacitu $b$ a ostatné kapacitu aspoň $b$.

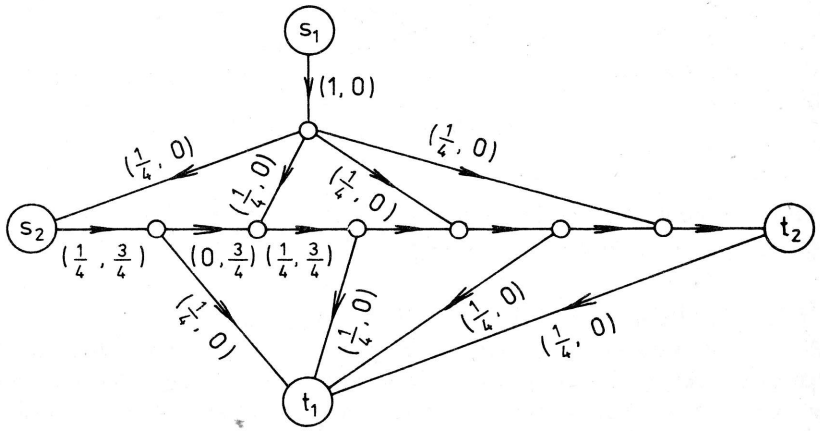



Obr. 6. Nahradenie rebra šípmi.

Každý $k$-produktový tok po rebre $[u, v]$ vieme ľahko simulovať v šípoch a obrátene. Žiaľ, takáto substitúcia nefunguje pre dolné medze. (Ak by sme mali takú transformáciu, tak by sme NP-ťažký problém prípustného toku v grafe s dolnými medzami vedeli sformulovať ako úlohu lineárneho programovania, a teda vedeli riešiť polynomiálne, čo je málo pravdepodobné.)

Uvedená transformácia na digrafy nie je príliš užitočná, lebo doteraz nebol vypracovaný priamy polynomiálny algoritmus ani na riešenie 2-produktovej úlohy v digrafoch. Je to však úloha lineárneho programovania, a teda polynomiálne ju vieme riešiť.

Nájsť algoritmus pre maximálny 2-produktový tok v digrafoch nebude ľahké. K tomuto záveru vedú príklady takého druhu ako na obr. 7, kde hodnoty maximálneho toku sú malé zlomky. (Každý šíp má kapacitu 1.) Iný dôvod je ten, že viaceré úlohy možno na úlohu maximálneho 2-produktového toku polynomiálne pretransformovať, ako uvidíme v nasledujúcej časti.




Obr. 7. Maximálny 2-produktový tok v digrafe.

## 6.F
## Polynomiálne transformácie úlohy lineárneho programovania na tokové úlohy v digrafoch

Tokové úlohy v digrafoch sa ľahko formulujú v tvare úloh lineárneho programovania. Tu zavedieme niektoré ďalšie tokové úlohy a ukážeme, že úlohu lineárneho programovania možno na ne polynomiálne pretransformovať bez ohľadu na existenciu polynomiálneho algoritmu pre úlohu lineárneho programovania. Okrem toho, predložené transformácie sú lineárne, resp. takmer lineárne a tiež





pomocou nich dokážeme o niektorých celočíselných tokových úlohách, že sú NP-ťažké.

### HOMOLOGICKÝ TOK S DOLNÝMI MEDZAMI

Uvažujme digraf so vstupom $s$ a výstupom $t$. Nech každý šíp $ij$ má dolnú medzu (na tok) $a_{ij}$ a kapacitu $b_{ij}$, kde pripúšťame $a_{ij} = 0$ a $b_{ij} = \infty$. Ako vieme (pozri 6.C), tu ľahko nájdeme prípustný tok, avšak úloha je ťažšia, ak sú navyše všetky hrany rozložené do tried tzv. ***homologických hrán*** a žiadame, aby každou hranou z tej istej triedy prechádzal rovnako veľký tok, t. j. $x(e_1) = x(e_2)$, kedykoľvek $e_1$ *a* $e_2$ patria do tej istej triedy, teda ak $e_1$ a $e_2$ sú homologické hrany. Úlohou je nájsť prípustný tok spĺňajúci homologické požiadavky: nájsť prípustný homologický tok.

Táto úloha, zavedená v [Berge, Ghouila–Houri (1962)], je špeciálnou úlohou lineárneho programovania, ale je dostatočne obsažná, ako vidieť z nasledujúceho výsledku [Itai (1978)].

**Veta 1.** Úlohu lineárneho programovania možno polynomiálne pretransformovať na úlohu nájsť prípustný homologický tok.

DÔKAZ. Podľa vety 2.B.2 stačí pretransformovať úlohu: nájsť nezáporné riešenie systému lineárnych rovníc

$$\sum_{j=1}^{n} a_{ij} x_j = b_i, \quad i = 1, 2, \ldots, m, \tag{1}$$

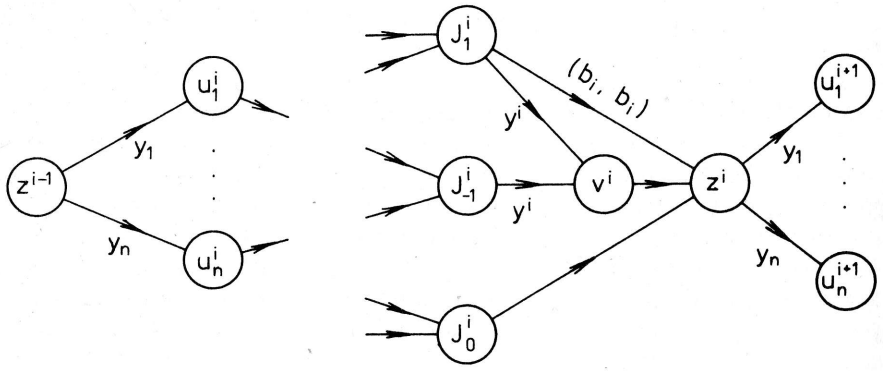
kde všetky $a_{ij} \in \{-1, 0, 1\}$, $b_i \in \{0, 1\}$.

Ak pre $p = -1, 0, 1$ a $i = 1, 2, \ldots, m$ položíme $J_p^i := \{j \mid a_{ij} = p\}$, tak predchádzajúci systém rovníc (1) môžeme ekvivalentne zapísať takto:

$$\sum_{j \in J_1^i} x_j - \sum_{j \in J_{-1}^i} x_j = b_i, \quad i = 1, 2, \ldots, m. \tag{2}$$

Teraz k systému (2) skonštruujeme digraf $G$ pozostávajúci z $m$ častí. Každá časť obsahuje $n + 5$ vrcholov: $u_1^i, u_2^i, \ldots, u_n^i, v^i, z^i, J_{-1}^i, J_0^i, J_1^i$. V závislosti na $i$-tej rovnici systému (2), hrany $i$-tej časti sú tieto (pozri obr. 1): Ak $j \in J_p^i$, potom vedieme šíp z $u_j^i$ do $J_p^i$ ohodnotený dvojicou $(0, \infty)$ (dolná medza je 0, kapacita $\infty$): vrchol $J_1^i$ spojíme do $z^i$ šípom s ohodnotením $(b_i, b_i)$: z $J_1^i$ aj z $J_{-1}^i$ vedieme homologické šípy do $v^i$ s ohodnotením $(0, \infty)$: z $J_0^i$ a z $v^i$ vedieme šípy do $z^i$ s ohodnotením $(0, \infty)$.

Digraf $G$ ešte obsahuje vstupný vrchol $s = z^0$, výstup $t = z^n$ a pre každé $j = 1, 2, \ldots, n$, triedu homologických hrán $(z^0, u_j^1), (z^1, u_j^2), \ldots, , (z^{m-1}, u_j^m)$ s ohodnotením $(0, \infty)$.




Obr. 1. Časť digrafu $G$ z dôkazu vety 1.

Na obr. 1, každý zo symbolov $y_1, \ldots, y_n, y^1, \ldots, y^m$ zodpovedá vždy jednej homologickej triede.

Dokážeme, že systém (2) má nezáporné riešenie práve vtedy, keď $G$ má prípustný homologický tok.

Nech $x \geq 0$ je riešenie systému (2). Potom definujeme tok $y$ v $G$ takto: $y(z^0, u_j^1) = \ldots = y(z^{m-1}, u_j^m) = x_j$. Ďalej kladieme $y(J_1^i, z^i) = b_i$ a

$$y(J_1^i, v^i) = y(J_{-1}^i, v^i) = \sum_{j \in J_{-1}^i} x_j = \sum_{j \in J_1^i} x_j - b_i .$$

Tok $y$ na ostatných hranách je už určený z podmienok kontinuity.

Obrátene, ak $y$ je tok v $G$, tak nech $y_1, \ldots, y_n$ a $y^1, \ldots, y^m$ sú jeho hodnoty na homologických hranách ako na obr. 1. Potom zrejme môžeme položiť $x_j = y_j$ a získame riešenie systému (2), a teda aj systému (1). ■

## SELEKTÍVNY VIACPRODUKTOVÝ TOK S DOLNÝMI MEDZAMI

Nech je daný digraf $G$ a v ňom $2k$ špeciálnych vrcholov $s_1, s_2, \ldots, s_k$ a $t_1, t_2, \ldots, t_k$. Nech každá hrana $e$ má danú dolnú medzu $a(e)$ a kapacitu $b(e)$. $k$-tica 1-produktových tokov $(x_1, x_2, \ldots, x_k)$ sa nazýva (*prípustný*) *k-produktový tok s dolnými medzami*, ak $x_i$ je $s_i$—$t_i$ tok ($s_i$ je vstup a $t_i$ výstup) pre každé $i = 1, 2, \ldots, k$ a pre každú hranu $e \in E(G)$ platí:

$$a(e) \leq x_1(e) + x_2(e) + \ldots + x_k(e) \leq b(e).$$

Tiež hovoríme, že $(x_1, \ldots, x_k)$ je $(s_1, \ldots, s_k)$—$(t_1, \ldots, t_k)$ tok.





Ak je navyše pre každú hranu $e \in E(G)$ daná množina produktov, ktoré môžu hranou prechádzať, tak hovoríme o *selektívnom viacproduktovom toku.*

Poznamenajme, že viacproduktové toky v rôznych modifikáciách majú (podobne ako 1-produktové) dlhú históriu. Medzi prvé práce patrí [Ford, Fulkerson (1958)].

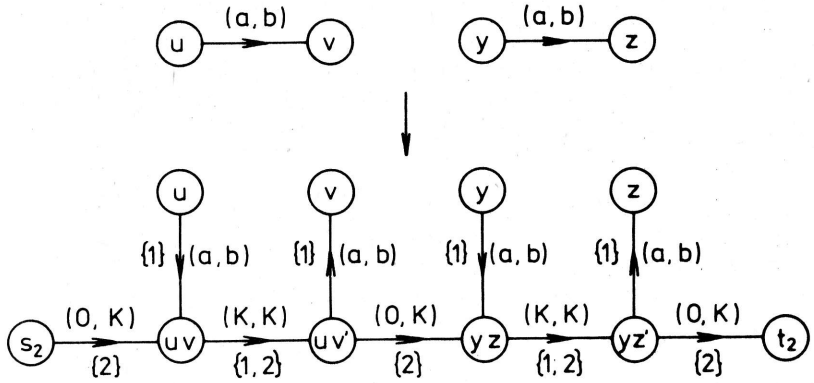
Nasledujúci výsledok dáva Itai (1978).

**Veta 2.** Úlohu nájsť prípustný (1-produktový) homologický tok možno polynomiálne pretransformovať na úlohu nájsť prípustný selektívny 2-produktový tok s dolnými medzami.

DÔKAZ. Môžeme predpokladať, že každá trieda homologických hrán obsahuje nanajvýš dve hrany. (Ak by nejaká trieda $C$ obsahovala hrany $e_1, e_2, \ldots, e_p$ kde $p \geq 3$, tak vsunieme do hrany $e_p$ $p-2$ nových vrcholov, čím ju rozdelíme na $p-1$ hrán $e_1', e_2', \ldots, e_{p-1}'$. Potom namiesto triedy $C$ budeme mať $p-1$ tried homologických hrán: $\{e_1, e_1'\}, \ldots, \{e_{p-1}, e_{p-1}'\}$.)

Dôkaz je založený na myšlienke nahradiť homologické požiadavky zavedením druhého produktu.

Nech $(u, v)$ a $(y, z)$ sú dve homologické hrany a nech $a$ je ich dolná medza a $b$ ich kapacita (zrejme môžeme predpokladať, že homologické hrany majú dolné medze aj kapacity rovnaké). Tieto dve hrany môžeme nahradiť konštrukciou ako na obr. 2, kde $uv$, $uv'$, $yz$ a $yz'$ sú nové vrcholy, $s_2$ a $t_2$ sú nový vstup a výstup pre




Obr. 2. Nahradenie homologických hrán.

druhý produkt (spoločné pre všetky konštrukcie). Prípustné množiny produktov sú vyznačené na každej hrane; $K$ je dostatočne veľká konštanta, ktorú zabezpečíme neskôr.

Je zrejmé, že účinok homologických hrán je zabezpečený.

Ešte nám zostáva zabezpečiť konštantu $K$, ktorá musí byť aspoň taká veľká ako tok na ľubovoľnej hrane. Na druhej strane, jej binárny zápis musí byť dostatočne krátky, aby sme zabezpečili polynomiálnu transformáciu.

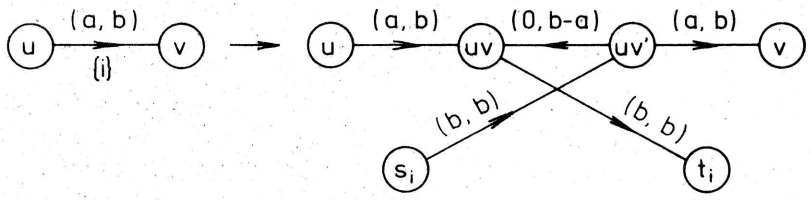
Úlohu nájsť prípustný homologický tok možno sformulovať v tvare lineárnych rovníc a nerovníc, čo možno upraviť na tvar $Ax = b$, $x \geq 0$ (doplnkovými premennými). Pritom všetky koeficienty $a_{ij} \in \{-1, 0, 1\}$ a každé $b_i$ je ohraničené maximálnou konečnou kapacitou (nekonečné kapacity sa do formulácie nedostanú). Z teórie lineárneho programovania je známe, že ak existuje prípustné riešenie, tak existuje aj bázické riešenie. Nech $B$ je podmatica matice $A$ zodpovedajúca nejakému bázickému riešeniu a nech $B_i$ vznikne z $B$ výmenou $i$-tého stĺpca za $b$. Potom podľa Cramerovho pravidla je $x_i = \det(B_i)/\det(B)$, a teda $|x_i| \leq |\det(B_i)|$.

Ak $M$ je maximálna konečná kapacita hrany, tak každá zložka vektora $b$ je ohraničená číslom $M$. Keďže prvky ostatných stĺpcov v $B_i$ sú z množiny $\{-1, 0, 1\}$, tak podľa Hadamarovej nerovnosti máme: $|x_i| \leq Mn^{n/2}$. Takto stačí voliť $K = Mn^{n/2}$ a na binárny zápis tejto konštanty stačí $\log M + \frac{1}{2} n \log n + 1$ bitov, čo zabezpečuje polynomiálnu transformáciu. ■

## 2-PRODUKTOVÝ TOK S DOLNÝMI MEDZAMI

Avšak Itai (1978) ukazuje, že od selektívnosti možno upustiť:

**Veta 3.** Úlohu nájsť prípustný selektívny 2-produktový tok s dolnými medzami možno polynomiálne pretransformovať na úlohu nájsť prípustný 2-produktový tok s dolnými medzami.




Obr. 3. Nahradenie selektívnej hrany.





DÔKAZ. Každú hranu $(u, v)$, ktorá pripúšťa iba tok $i \in \{1, 2\}$, nahradíme tak, ako ukazuje obr. 3. Vrcholy $uv$ a $uv'$ sú nové. Vrcholy $s_i$ a $t_i$ sú pôvodné (vstup a výstup pre $i$-tý produkt) a predpokladáme o nich, že $s_i$ je prameňom a $t_i$ ústím (inak to ľahko možno zariadiť pridaním nových vrcholov $s_i'$ a $t_i'$ a hrán $(s_i', s_i)$ a $(t_i, t_i')$: zrejme $s_i'$ a $t_i'$ už budú také). Z toho vyplýva, že hranami $(s_i, uv')$ a $(uv, t_i)$ môže tiecť len produkt $i$, ktorý ich musí nasycovať. Preto aj hranami $(u, uv)$ a $(uv', v)$ bude tiecť len produkt $i$, a to rovnakej veľkosti. ■

## 2-PRODUKTOVÝ TOK S POŽIADAVKAMI A MAXIMÁLNY 2-PRODUKTOVÝ TOK

Tu upustíme od dolných medzí (t. j. všetky budú nulové), ale budeme žiadať, aby $s_i$—$t_i$ tok $x_i$ ($i$-tého produktu) mal veľkosť $f(x_i) \geq d_i$ ($i = 1, 2,$), kde $d_1$, $d_2$ sú dané čísla (požiadavky).

Iná úloha je taká, kde nemáme predpísané požiadavky, ale treba maximalizovať $f(x_1) + f(x_2)$: vtedy hovoríme o úlohe nájsť maximálny 2-produktový tok.

Aj nasledujúci výsledok dáva Itai (1978).

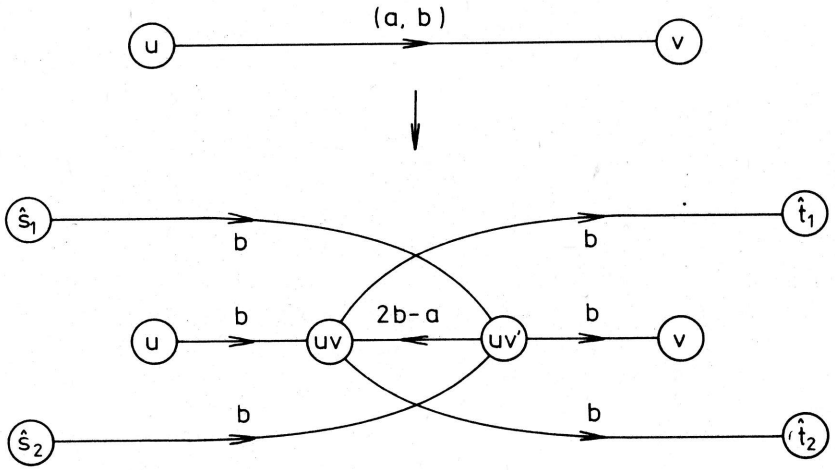
**Veta 4.** (i) Úlohu nájsť prípustný 2-produktový tok s dolnými medzami možno polynomiálne pretransformovať na úlohu nájsť 2-produktový tok s požiadavkami.

(ii) Úlohu nájsť 2-produktový tok s požiadavkami možno polynomiálne pretransformovať na úlohu nájsť maximálny 2-produktový tok.

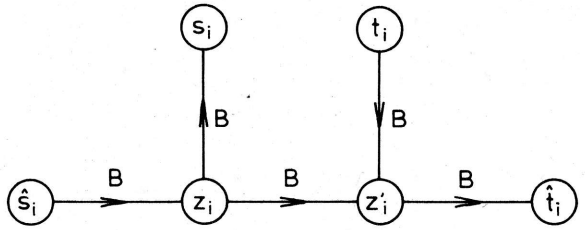
DÔKAZ. (i) K danému digrafu $G$ s dolnými medzami zostrojíme digraf $\hat{G}$ (bez dolných medzí) pridaním nových vstupov $\hat{s}_1$, $\hat{s}_2$ a nových výstupov $\hat{t}_1$, $\hat{t}_2$ a nasledujúcimi úpravami: (1) Každú hranu $(u, v) \in E(G)$ (s dolnou medzou $a$, kapacitou $b$) nahradíme štruktúrou z obr. 4, kde $uv$ a $uv'$ sú nové vrcholy zodpovedajúce hrane $(u, v)$ a každá hrana má už len kapacitu (dolné medze sú nulové). (2) Pôvodné vstupy a výstupy sa stanú obyčajnými vrcholmi a pre $i = 1$, 2 pridáme štruktúru z obr. 5, kde $z_i$, $z_i'$ sú nové vrcholy a $B = \sum_{e \in E(G)} b(e)$ (súčet všetkých kapacít hrán v $G$). (3) Požiadavky na veľkosť jednotlivých tokov kladieme: $d_1 = d_2 = 2B$.

Ak existuje v $G$ tok $(x_1, x_2)$, tak ľahko zostrojíme v $\hat{G}$ tok $(\hat{x}_1, \hat{x}_2)$ nasycujúci všetky hrany incidujúce s niektorými zo vstupov alebo výstupov: $\hat{s}_1$, $\hat{s}_2$, $\hat{t}_1$, $\hat{t}_2$ ($\hat{x}_i(u, uv) := \hat{x}_i(uv', v) := x_i(u, v)$). Potom $f(\hat{x}_i) = 2B$, a teda požiadavky budú splnené. Obrátene, k takémuto toku v $\hat{G}$ nájdeme ľahko tok v $G$. Tým je časť (i) dokázaná.

(ii) Nech je daný digraf $G$ s požiadavkami $d_1$, $d_2$. Utvorme digraf $\bar{G}$ pridaním nových vstupov $\bar{s}_i$, výstupov $\bar{t}_i$ a hrán $(\bar{s}_i, s_i)$, $(t_i, \bar{t}_i)$ kapacity $d_i$ ($i = 1, 2$). Potom




Obr. 4. Odstránenie dolných medzí.




Obr. 5. Ku konštrukcii digrafu $\hat{G}$.

zrejme v $G$ existuje $(s_1, s_2)$—$(t_1, t_2)$ tok $(x_1, x_2)$ s $f(x_1) \geq d_1$ a $f(x_2) \geq d_2$, práve vtedy, keď v $\bar{G}$ existuje maximálny $(\bar{s}_1, \bar{s}_2)$—$(\bar{t}_1, \bar{t}_2)$ tok $(\bar{x}_1, \bar{x}_2)$ s $f(x_1) + f(x_2) = d_1 + d_2$. ■

Každá zo spomínaných tokových úloh v digrafoch je špeciálnou úlohou lineárneho programovania, a teda z viet 1 až 4 máme:

**Veta 5.** Nezávisle od existencie polynomiálneho algoritmu pre úlohu lineárneho programovania sú nasledujúce úlohy s racionálnymi dátami, kde úlohy o tokoch sú v digrafoch, polynomiálne ekvivalentné:

(1) Úloha lineárneho programovania.
(2) Homologický tok s dolnými medzami.





(3) Selektívny 2-produktový tok s dolnými medzami.
(4) 2-produktový tok s dolnými medzami.
(5) 2-produktový tok s požiadavkami.
(6) Maximálny 2-produktový tok.

Poznamenajme, že uvedené konštrukcie v dôkazoch možno rozšíriť na ľubovoľný pevný počet $k$ produktov, a teda vetu 5 formulovať pre $k$-produktové toky ($k \geq 2$). Istá transformácia úlohy lineárneho programovania na úlohu maximálneho mnohoproduktového toku (s premenným $k$) je tiež v [Adeľson–Veľskij et al. (1975)].

Ak namiesto úlohy lineárneho programovania, resp. úlohy nájsť nezáporné riešenie systému rovníc, zoberieme úlohu nájsť také riešenie $x$ systému rovníc $Ax = b$, kde každé $x_i \in \{0, 1\}$, ktorá je NP-ťažká, tak vidíme, že všetky spomínané tokové úlohy sú NP-ťažké, ak žiadame celočíselnosť tokov. Naviac možno požadovať, aby dolné medze, resp. kapacity boli z množiny $\{0, 1\}$ (hranu kapacity $b$ nahradíme $b$ paralelnými cestami dĺžky 2 a kapacity 1). Preto dôkaz nasledujúceho tvrdenia [Itai (1978)] prenechávame čitateľovi (6.Z.13).

**Veta 6.** Nasledujúce celočíselné tokové úlohy v digrafoch sú NP-ťažké, dokonca v prípade, že všetky dolné medze aj kapacity sú z množiny $\{0, 1\}$.

(1) Homologický tok s dolnými medzami.
(2) Selektívny 2-produktový tok s dolnými medzami.
(3) 2-produktový tok s dolnými medzami.
(4) 2-produktový tok s požiadavkami.
(5) Maximálny 2-produktový tok.

Poznamenajme, že tvrdenia (1) a (3) v slabšej forme dáva Sahni (1974). Even, Itai a Shamir (1976) dávajú priamo transformáciu problému splniteľnosti na úlohu (4) a túto transformujú na (5).

## 6.Z
## Cvičenia a poznámky

1. Ukážte príklad, kde algoritmus Forda a Fulkersona nedá ani v limite maximálny $s$—$t$ tok. [Návod: Nech $r := (\sqrt{5}-1)/2$, $b_p := r^p$ $(p = 0, 1, \ldots)$. Potom zrejme $b_{p+2} = b_p - b_{p+1}$, rad $\Sigma\, b_p$ konverguje, a teda má nejaký súčet $B$. Nech $G$ je digraf s vrcholmi $s$, $u_i$, $v_j$, $t$ $(i, j = 1, \ldots, 4)$, šípmi $su_i$, $v_jt$, $u_iv_i$ $(i, j = 1, \ldots, 4)$, a $u_iv_j$, $v_iv_j$, $v_iu_j$ $(i, j = 1, \ldots, 4 : i \neq j)$. Šípy $u_1v_1$, $u_2v_2$, $u_3v_3$ a $u_4v_4$

majú kapacity $b_0$, $b_1$, $b_2$, $b_2$: ostatné majú kapacity rovné $B$. V 0-tom kroku zostrojíme tok veľkosti $b_0$ po ceste $su_1v_1t$, a tým získame na šípoch $u_iv_i$ rezervy 0, $b_1$, $b_2$, $b_2$. V induktívnom $p$-tom kroku ($p \geq 1$) najprv zväčšíme tok o $a_{p+1}$ tak, aby sme z rezerv 0, $b_p$, $b_{p+1}$, $b_{p+1}$ (nie nevyhnutne v tomto poradí) získali rezervy 0, $b_p - b_{p+1}$, 0, $b_{p+1}$ a potom zväčšíme tok o $b_p - b_{p+1}$ a dostaneme rezervy $b_{p+2}$, 0, $b_{p+2}$, $b_{p+1}$. Takto po $p$-tom kroku získame $s$—$t$ tok veľkosti $b_0 + b_1 + \ldots + b_p < B$, ale maximálny tok má veľkosť $4B$. [Ford, Fulkerson (1962)].

2. Tok sa nazýva *acyklický*, ak šípy, na ktorých je kladný, tvoria acyklický digraf. Tok $x$ s veľkosťou $f(x) = 0$ sa nazýva *cirkulácia*. Dokážte, že každý tok možno rozložiť na acyklický tok a cirkuláciu.
3. Ukážte, že algoritmy zo 6.B zaručujú konvergenciu k maximálnemu toku pri ľubovoľných reálnych kapacitách.
4. V danom acyklickom digrafe treba nájsť minimálny počet ciest, z ktorých každé dve sú disjunktné a ktoré spolu obsahujú všetky vrcholy. Sformulujte to ako tokovú úlohu.
5. Dajte tokovú formuláciu nasledujúcich dvoch úloh: (a) Pre daný systém množín $\{S_1, \ldots, S_k\}$ treba nájsť tzv. *systém rozličných reprezentantov* $e_1, \ldots, e_k$ (t. j. $e_i \in S_i$ a $e_i \neq e_j$ pre $i \neq j$). (b) Pre dva systémy $\{S_1, \ldots, S_k\}$ a $\{T_1, \ldots, T_k\}$ treba nájsť spoločný systém rozličných reprezentantov.

   Ukážte, že pre tri systémy je už taký problém NP-ťažký [Návod: Transformujte úlohu 3-rozmerného párenia.]
6. Ukážte, že algoritmus polocyklov a polociest pre najlacnejší maximálny tok (pozri 6.D) možno upraviť pre úlohy, kde cena toku po šípe nie je len lineárna, ale všeobecne konvexná po častiach lineárna funkcia toku.
7. Ukážte, že (všeobecnú) dopravnú úlohu možno pretransformovať na Hitchcockovu dopravnú úlohu. [Návod: Utvoríme bipartitný digraf $H$ s partiami $E(G)$ a $V(G)$, kde požiadavka vo vrchole $ij$ je $b_{ij}$: požiadavka vo vrchole $k$ je $b(k, V_G^+(k)) - a_k$, ak $k$ je vstupom v $G$, $b(k, V_G^+(k)) + b'_k$, ak $k$ je výstupom v $G$ a $b(k, V_G^+(k))$ inak. Pritom šíp $(ij, k)$ v $H$ má cenu $c_{ij}$, *ak* $j = k$, 0 pre $i = k$, a $\infty$ v ostatných prípadoch. Položte $x_G(ij) = x_H(ij, j)$. [Ford a Fulkerson (1962)].]
8. Dokážte, že zmena $c_{ij}$ na $c_{ij} + g_i + h_j$ v Hitchcockovej dopravnej úlohe nemá vplyv na optimalitu riešenia.
9. Vychádzajúc z modelu Hitchcockovej dopravnej úlohy dokážte, že jednožen-stvo je najlepšia forma manželstiev pre spoločnosť. (Tí, ktorým sa to nezdá, podobne dokážu, že je to najhoršia forma.) [Dantzig (1962).] [Návod: Predpokladajte, že každá osoba má požiadavku celkove na jednu osobu iného pohlavia, ktorú možno zložiť aj z príspevkov viacerých ľudí. Zo spolužitia





muža $i$ a ženy $j$ má spoločnosť zisk $c_{ij}$, ak ide o „plný úväzok“ (inak sa berie úmerná časť z $c_{ij}$). Po prípadnom dodaní fiktívnych osôb môžeme predpokladať, že počty mužov a žien sú rovnaké. Vašou úlohou je ukázať, že sumárny maximálny zisk sa dá realizovať aj vtedy, keď každé spolužitie bude „plnoúväzkové“.]

10. Dokážte, že v algoritme pre Hitchcockovu dopravnú úlohu z časti 6.D pri zmene riešenia poklesne hodnota účelovej funkcie o $\delta(u_i + v_j - c_{ij})$.
11. Modifikujte príklad Zadeha z obr. 6.D.3 a ukážte exponenciálnu zložitosť simplexovej metódy zo 6.D pre Hitchcockovu dopravnú úlohu. [Návod: Vstupmi budú $u_1, \ldots, u_k$, $t$ s požiadavkami 1, 3, 5, ..., $2^{k-1} + 2^{k-3}$, $2^k + 2^{k-2}$ a výstupmi $v_1, v_2, \ldots, v_k$, $s$ s požiadavkami 2, 2, 5, ..., $2^{k-1} + 2^{k-3}$, $2^k + 2^{k-2}$. Všetky šípy $su_i$ a $v_jt$ obrátime a pridáme šíp $ts$ ceny $-\infty$: doplníme šípmi ceny $\infty$. Ak začneme z kostry s hranami $su_i$, $v_jt$, $ts$, tak zmeny riešenia budú zodpovedať zmenám toku v pôvodnom príklade. Overte na obr. 6.D.4.]
12. (Toky s násobičmi.) Nech každý šíp $ij$ digrafu má okrem kapacity $b_{ij}$ priradený tzv. *násobič toku* $w_{ij} > 0$. Ak do hrany vstúpi $x_{ij}$ jednotiek, tak vystúpi $x_{ij}w_{ij}$ jednotiek. $s$—$t$ tok $x$ musí spĺňať podmienky: $0 \leq x_{ij} \leq b_{ij}$ a $\sum w_{ji}x_{ji} - \sum x_{ij} = 0$ pre $i \neq s, t$. Čistý odtok z $s$ je $f_s^+(x) := \sum x_{sj} - \sum w_{js}x_{js}$ a čistý prítok do $t$ je $f_t^-(x) := \sum w_{jt}x_{jt} - \sum x_{tj}$. Obvyklou úlohou je *minimalizovať stratu toku*, t. j. $f_s^+(x) - f_t^-(x)$, pričom často býva jedna z hodnôt $f_s^+(x)$, $f_t^-(x)$ predpísaná.

    (a) Navrhnite nejaký postup pre hľadanie optima ([Jewell (1962)], [Onaga (1967)], [Černý (1972)]). Onaga dokazuje analógiu vety 6.D.1.)

    (b) Ukážte na príklade, že celočíselné dáta nezaručujú celočíselný optimálny tok. (Týmto sa zmenšuje význam tokov s násobičmi.)

    (c) Dokážte, že problém nájsť celočíselný tok s maximálnym čistým prítokom do $t$ je NP-ťažký, dokonca pre planárne acyklické digrafy s predpísaným alebo aj nepredpísaným čistým odtokom z $s$. [Návod: Transformujte problém rozdelenia čísel $a_1, \ldots, a_p$ na dve časti so súčtami $d = \sum a_i/2$. Zostrojte digraf so šípmi $si$ kapacity 1 a s násobičom 1, $i\bar{t}$ kapacity 1 a s násobičom $a_i$ a šípom $\bar{i}t$ kapacity $d$ a s násobičom 1 [Sahni (1974)].]
13. Dokážte vetu 6.F.6.

    Poznamenajme, že Fortune et al. (1980) dokázali, že problém nájsť v digrafe pre dané $s_1$, $s_2$, $t_1$, $t_2$ dve disjunktné cesty $s_1$—$t_1$ a $s_2$—$t_2$ je NP-ťažký. Na druhej strane, takýto problém v grafoch je polynomiálne riešiteľný [Shiloach (1980)].
14. Nech $G$ je digraf projektu (pozri 4.A), kde každej činnosti (šípu) $ij$ je priradené číslo $a_{ij}^0$ — *normálne trvanie činnosti*, číslo $b_{ij}$ — *najkratšie trvanie činnosti* a číslo $c_{ij}$ *cena* (náklady potrebné na skrátenie trvania o jednotku). Teda skrátenie činnosti z $a_{ij}^0$ na $t_{ij}$, $b_{ij} \leq t_{ij} \leq a_{ij}^0$, si vyžaduje zvýšenie nákladov

o $c_{ij}(a_{ij}^0 - t_{ij})$. Nech $T_a^0$ je trvanie projektu vypočítané z čísel $a_{ij}^0$ a $T_b$ zasa z čísel $b_{ij}$. Úlohou je zistiť tzv. ***nákladovú krivku projektu,*** t. j. pre každé $T \in [T_b, T_a^0]$ určiť minimálne sumárne náklady $C(T)$ potrebné na skrátenie činností tak, aby celkové trvanie projektu bolo $T$; naviac, žiadame nájsť aj jednotlivé trvania $t_{ij}$. Toto je úloha parametrického lineárneho programovania a nákladová krivka je vždy konvexná po častiach lineárna.

Pre túto špeciálnu úlohu boli vypracované viaceré postupy využívajúce duálnu úlohu a toky v sieťach [Kelley (1961)], [Fulkerson (1961)]. K úlohe o maximálnom toku sa však dostaneme aj jednoduchšie:

Nech $T_a^0$ je zistené trvanie projektu z hodnôt $a_{ij}^0$ (pozri 4.Z.6) a nech $\hat{G} \subseteq G$ pozostáva zo všetkých kritických činností (šípov), t. j. $\hat{G}$ je zjednotením všetkých kritických (najdlhších) $s$—$t$ ciest v $G$. Ak chceme projekt skrátiť o $\delta > 0$, tak musíme všetky kritické cesty skrátiť o $\delta$. Najlacnejšie sa to urobí tak, že nájdeme najlacnejší $s$—$t$ rez v $\hat{G}$, t. j. minimový $s$—$t$ rez $(V_s, V_t)_{\hat{G}}$, pričom kapacita šípa $ij$ je $c_{ij}$ ak $a_{ij}^0 > b_{ij}$ a $\infty$ ak $a_{ij}^0 = b_{ij}$. Potom všetky hrany rezu skrátime o $\delta$, t. j. trvania $a_{ij}^0$ zmeníme na $a_{ij}^1 := a_{ij}^0 - \delta$ a mimo rezu bude $a_{ij}^1 := a_{ij}^0$. Tým sa projekt skráti na dĺžku $T_a^1 = T_a^0 - \delta$, ak $\delta$ volíme dostatočne malé. Číslo $\delta$ treba voliť maximálne možné, ale tak, aby dĺžka žiadnej nekritickej $s$—$t$ cesty v $G$ nepresiahla číslo $T_a^0 - \delta$. Tým zostrojíme lineárny úsek medzi $T_a^0$ a $T_a^1$ a vychádzajúc z hodnôt $a_{ij}^1$ môžeme celý postup opakovať.

Navrhnite postup pre hľadanie čísla $\delta$ a celú metódu zdôvodnite. Tiež uvažujte prípad, kedy namiesto $c_{ij}(a_{ij}^0 - t_{ij})$ je daná konvexná po častiach lineárna nákladová funkcia $c(t_{ij})$ činnosti $ij$.

# 7
# MIERY SÚVISLOSTI

Graf môže byť súvislý alebo nesúvislý. Avšak niektoré súvislé grafy môžu mať „vyššiu" súvislosť, napr. ak sú bez mostov, resp. artikulácií (kap. 3). Podobne je to v digrafoch. V teórii grafov sa zaviedli pojmy umožňujúce merať súvislosť: hranová, resp. vrcholová súvislosť. Medzi dvoma vrcholmi (miestami komunikačnej siete) bude intuitívne veľká súvislosť, ak treba na ich separovanie odstrániť mnoho hrán, resp. vrcholov. Ukazuje sa, že to je ekvivalentné tomu, že sú spojené mnohými disjunktnými cestami. Z takýchto „lokálnych" mier možno vyrobiť „globálne" miery súvislosti (budeme žiadať, aby každé dva vrcholy dostatočne súviseli). Uvedieme, ako tieto súvislosti nájsť. Takýto postup možno nazvať analýzou grafu, resp. digrafu a pri hranách s kapacitami ide vlastne o zistenie maximálneho možného toku medzi každými dvoma miestami komunikačnej siete. Pre grafy riešime aj obrátenú úlohu, tzv. syntézu, kde ide o nájdenie najlacnejšej komunikačnej siete umožňujúcej toky daných veľkostí. Inou mierou súvislosti siete je maximálny počet parciálnych sietí, na ktoré možno danú sieť rozložiť tak, aby v rámci každej siete bola zabezpečená cesta (z jedného miesta do ostatných alebo medzi každými dvoma). Tu ide vlastne o rozklad grafu na hranovo-disjunktné súvislé faktory.

## 7.A
## Hranová a vrcholová súvislosť

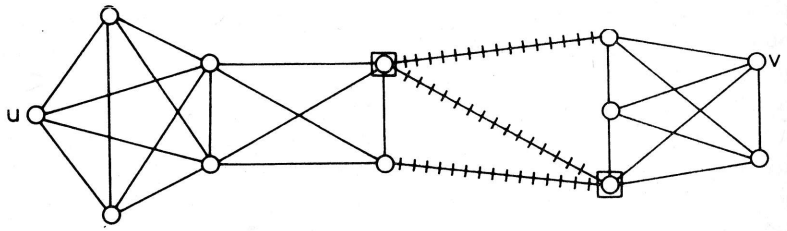
Nech $G$ je pseudomigraf reprezentujúci nejakú komunikačnú sieť. Túto sieť budeme pokladať za tým spoľahlivejšiu, čím viac liniek z nej treba odstrániť, aby sme porušili všetky $u$—$v$ sledy pre nejaké $u$, $v$. Podobne možno hovoriť o počte odstránených vrcholov v súvislosti so sledmi medzi zvyšnými vrcholmi. To vedie k nasledujúcim pojmom.

Nech $u, v \in V(G)$. Hovoríme, že množina hrán $S \subseteq E(G)$ separuje $u$ a $v$ (v tomto poradí), ak v $G - S$ neexistuje $u$—$v$ sled, vtedy sa $S$ nazýva ***hranový*** $u$—$v$ ***separátor.*** Podobne, ak $u, v \in V(G)$ a $S \subseteq V(G) - \{u, v\}$, tak $S$ sa nazýva ***vrcholový*** $u$—$v$ ***separátor,*** ak v $G - S$ neexistuje $u$—$v$ sled. (Ak $uv \in E(G)$, tak vrcholový $u$—$v$ separátor neexistuje.) V oboch prípadoch môžeme klásť $S = \emptyset$, ak už v $G$ neexistuje $u$—$v$ sled.

Minimálny (v zmysle inklúzie) vrcholový (resp. hranový) $u$—$v$ separátor sa nazýva *vrcholový (hranový) $u$—$v$ rez*.

S hranovými rezmi v grafoch a digrafoch sme sa už stretli v častiach 5.B a 6.A, ale inak definovanými. V skutočnosti však každý hranový $u$—$v$ rez pseudomigrafu $G$ má tvar $\{e \in E(G) |\ e$ je hrana $xy$, kde $x \in U$ a $y \in V(G) - U\}$ (za $U$ stačí zvoliť množinu všetkých vrcholov dosiahnuteľných z $u$ v $G - S$). Množinu všetkých hrán rezu označujeme symbolom $(U,\ V(G) - U)_G$; ak ide o *rebrový rez* (pozostávajúci výhradne z rebier), tak uprednostňujeme zápis $[U,\ V(G) - U]_G$.

Hranový $u$—$v$ rez $S$, ktorý má minimálny počet prvkov, nazývame *minimový*; potom hovoríme, že *hranová súvislosť* medzi $u$ a $v$ je $\lambda(u, v) = |S|$. Analogicky definujeme *vrcholovú súvislosť* $\varkappa(u, v)$ medzi $u$ a $v$ pre $uv \notin E(G)$. Obr. 1




Obr. 1. Minimové rezy, $\lambda(u, v) = 3$, $\varkappa(u, v) = 2$.

ilustruje tieto pojmy v grafe. Budeme hovoriť, že $u$—$v$ cesty $P_1, P_2, \ldots, P_k$ sú *vnútorne disjunktné*, ak žiadne dve z nich nemajú spoločný vnútorný vrchol a hovoríme, že sú *hranovo-disjunktné*, ak žiadne dve z nich nemajú spoločnú hranu.

Nasledujúci výsledok je známy pod názvom Mengerova veta. Menger (1927) formuloval svoj výsledok v tvare vrcholovej verzie v grafoch a ďalšie variácie vznikli neskôr.

**Veta 1** (Menger). Nech $G$ je pseudomigraf a $u$, $v$ nech sú jeho dva vrcholy. Potom platí:

(a) mohutnosť $\lambda(u, v)$ minimového hranového $u$—$v$ rezu je rovná maximálnemu počtu hranovo-disjunktných $u$—$v$ ciest,

(b) ak neexistuje hrana $uv$, tak mohutnosť $\varkappa(u, v)$ minimového vrcholového $u$—$v$ rezu je rovná maximálnemu počtu vnútorne disjunktných $u$—$v$ ciest.

DÔKAZ. Mengerova veta nám pripomína vetu (6.A.1) o minimovom reze a maximálnom toku. Túto vetu aj využijeme na dôkaz [Dantzig, Fulkerson (1956)].

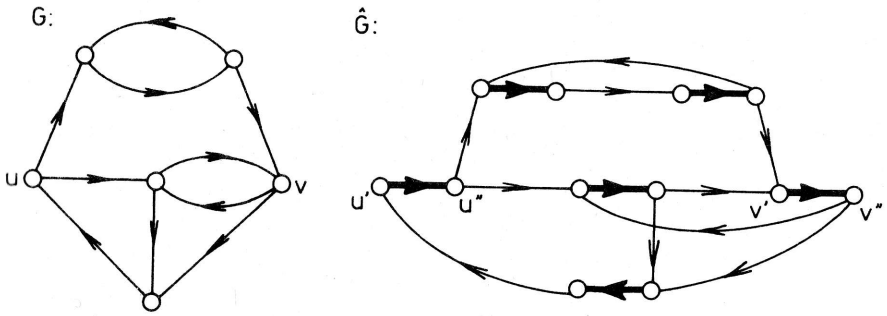




(a) Ak $G$ je pseudodigraf, tak stačí dať každej hrane kapacitu 1 a hľadať maximálny $u$—$v$ tok $x$ (slučky možno vynechať a namiesto násobných hrán uvažovať jednu so sumárnou kapacitou). Veľkosť toku $x$ je rovná kapacite minimového $u$—$v$ rezu, t. j. číslu $\lambda(u, v)$. Pri celočíselných kapacitách môžeme predpokladať, že $x$ je celočíselný tok. Ak sa vrátime k pôvodným hranám, tak na každej hrane bude tok 0 alebo 1. Nech $H \subseteq G$ pozostáva z tých hrán $e \in E(G)$, kde $x(e) = 1$. Z podmienok kontinuity toku máme: $\deg_H^+(i) = \deg_H^-(i)$ pre každý vrchol $i \neq u, v$. Takto každý nepredĺžiteľný ťah $T$ začínajúci vo vrchole $u$ končí vo $v$. V $H - E(T)$ zasa môžeme hľadať taký ťah atď. Teda v $H$ existuje $\lambda(u, v)$ hranovo-disjunktných $u$—$v$ ťahov. Keďže z každého ťahu možno vybrať cestu, tvrdenie je dokázané, lebo zrejme viac ako $\lambda(u, v)$ hranovo-disjunktných $u$—$v$ ciest v $G$ nemôže byť.

Ak $G$ má aj rebrá, tak každé rebro nahradíme dvojicou protišípov a použijeme predchádzajúci výsledok pre vzniknutý pseudodigraf $G'$.

(b) Aj tu tak isto možno rebrá nahradiť šípmi, a preto nech $G$ je pseudodigraf. Každý vrchol $i$ rozložíme na dva vrcholy $i'$ a $i''$, pridáme šíp $i'i''$, šípy prichádzajúce do $i$ necháme prichádzať do $i'$ a odchádzajúce z $i$ necháme odchádzať z $i''$ (pozri obr. 2). Všetkým šípom vzniknutého pseudodigrafu $\hat{G}$ predpíšeme kapacity 1.




Obr. 2. Digraf $\hat{G}$ na zisťovanie vrcholovej súvislosti v $G$.

Maximálny celočíselný $u''$—$v'$ tok v $\hat{G}$ nám „ukáže" $\varkappa(u, v)$ vnútorne disjunktných $u$—$v$ ciest v $G$. Keďže viac ich zrejme neexistuje, dôkaz je hotový. ■

*Hranová* (resp. *vrcholová*) *súvislosť* $\lambda(G)$ (resp. $\varkappa(G)$) pseudomigrafu $G$ je definovaná ako minimálny počet hrán (resp. vrcholov), po odstránení ktorých získame pseudomigraf bez silnej súvislosti alebo jednovrcholový. Teda ak $G$ nie je silne súvislý, tak $\lambda(G) = \varkappa(G) = 0$. Formálne:

$$\lambda(G) = \begin{cases} 0, & ak \ |V(G)| = 1; \\ \min \{\lambda(u, v) | u, v \in V(G), u \neq v\} & \text{inak.} \end{cases}$$

$$\varkappa(G) = \begin{cases} |V(G)| - 1, & \text{ak pre každé dva vrcholy } u, v \text{ existuje hrana } uv; \\ \min \{\varkappa(u, v) | u, v \in V(G), u \neq v, \text{neexistuje hrana } uv\} & \text{inak.} \end{cases}$$

Pseudomigraf $G$ sa nazýva *hranovo* (resp. *vrcholovo*) *k-súvislý*, ak $\lambda(G) \geq k$ (resp. $\varkappa(G) \geq k$).

V nasledujúcej vete sa obmedzujeme na grafy a digrafy, ale čitateľ by si mal premyslieť zovšeobecnenie na pseudomigrafy.

Nech $\delta(G)$, $\delta^+(G)$, resp. $\delta^-(G)$ označujú minimálne hodnoty stupňa, odchádzajúceho, resp. prichádzajúceho stupňa v $G$. Potom platí nasledujúce tvrdenie [Whitney (1932)].

**Veta 2.**

(a) Pre ľubovoľný graf $G$ je

$$\varkappa(G) \leq \lambda(G) \leq \delta(G).$$

(b) Pre ľubovoľný digraf $G$ je

$$\varkappa(G) \leq \lambda(G) \leq \min \{\delta^+(G), \delta^-(G)\}.$$

(c) Ľubovolné dva vrcholy $u$, $v$ grafu alebo digrafu $G$ sú spojené aspoň $\varkappa(G)$ vnútorne disjunktnými a aspoň $\lambda(G)$ hranovo-disjunktnými $u$—$v$ cestami.

DÔKAZ. V (a) aj (b) sú nerovnosti so stupňami zrejmé, a preto dokazujeme len vzťah $\varkappa(G) \leq \lambda(G)$. Nech $(U, \bar{U})_G$ je hranový rez v $G$ mohutnosti $\lambda(G)$ a nech $uv$ je jedna jeho hrana. Ukážeme, že po odstránení vhodných $\lambda(G)$ vrcholov bude $G$ bez silnej súvislosti alebo triviálny. Pre každú hranu $xy \in (U, \bar{U})_G - \{uv\}$ vynechajme ten z vrcholov $x$, $y$, ktorý je rôzny od $u$ aj $v$. Takto vynecháme nanajvýš $\lambda(G) - 1$ vrcholov. Ak v $U$ zostal len vrchol $u$, tak ešte odstránime vrchol $v$, inak odstránime vrchol $u$.

(c) Pre hranovú súvislosť $\lambda(G)$ to vyplýva ihneď z Mengerovej vety. Pre vrcholovú súvislosť $\varkappa(G)$ to máme zaručené len pre nesusedné vrcholy $u$, $v$. Pre kompletné grafy a digrafy je to tiež pravda, a preto nech $uv \in E(G)$ a $G$ nie je kompletný; potom $\varkappa(G) \leq n - 2$. Ak tvrdenie neplatí, tak maximálny počet $p$ vnútorne disjunktných $u$—$v$ ciest v $G$—$uv$ je nanajvýš $\varkappa(G) - 2$ a podľa Mengerovej vety existuje vrcholový rez $S$, $|S| = p$, separujúci $u$ a $v$. Nech $U$ je množina vrcholov dosiahnuteľných z $u$ v $(G - uv) - S$ a $\bar{U} = V(G) - U$. Ak existuje $x \in U - \{u\}$, tak $\{u\} \cup S$ zrejme separuje $x$ a $v$ v $G$ a ak existuje $y \in \bar{U} - \{v\}$, tak $S \cup \{v\}$ separuje $u$ a $y$. Keďže $|S| = p \leq \varkappa(G) - 2 \leq n - 4$, tak $x$





alebo $y$ existuje. Avšak rez $\{u\} \cup S$, resp. $S \cup \{v\}$ má nanajvýš $\varkappa(G) - 1$ prvkov, čo je nemožné. ■

Ďalšie vlastnosti hranových a vrcholových súvislostí možno nájsť v knihách z teórie grafov. Z novších úspechov pozri napr. Mader (1979). Dobrý prehľad je v [Mader (1979a)]. Fiedler (1973) zaviedol tzv. *algebraickú súvislosť* grafu $G$ ako druhé najmenšie vlastné číslo matice $A = (a_{ij})$, kde $a_{ij} = -1$, ak $ij \in E_G$, $a_{ii} = \deg_G(i)$ a $a_{ij} = 0$ inak. O význame tohto parametra pojednáva [Fiedler (1975)].

Z výpočtového hľadiska je vhodné sa obmedziť na grafy a digrafy, a preto tak ďalej aj robíme.

## HĽADANIE SÚVISLOSTÍ V GRAFOCH A DIGRAFOCH

Z dôkazu vety 1 vidíme, že na nájdenie čísla $\lambda(u, v)$ alebo $\varkappa(u, v)$ stačí vyriešiť jednu tokovú úlohu vo vhodnom digrafe s jednotkovými kapacitami. Ak $G$ je graf alebo digraf, maximálny tok je nanajvýš $n - 1$, a preto nanajvýš toľko zväčšovaní toku treba vo Fordovom a Fulkersonovom algoritme. Jedno zväčšenie toku možno urobiť v čase $O(m)$ (na nájdenie rezervnej polocesty použijeme algoritmus Moora), a tak celkove vystačíme s $O(mn)$ operáciami.

Trochu lepšie algoritmy vyvinuli Even a Tarjan (1975). Pre $\lambda(u, v)$ dávajú $O(mn^{2/3})$ algoritmus a pre $\varkappa(u, v)$ $O(mn^{1/2})$ algoritmus.

Čísla $\lambda(G)$, resp. $\varkappa(G)$ možno zistiť vyriešením množstva tokových úloh. (Pre $\lambda(G)$ je to $n(n-1)$ úloh, ak $G$ je digraf a $n(n-1)/2$ úloh ak $G$ je graf. Pre $\varkappa(G)$ hľadáme maximálny $u$—$v$ tok len pre $uv \notin E(G)$.) Avšak stačí riešiť omnoho menej tokových úloh.

Na určenie $\lambda(G)$ stačí nájsť minimový hranový rez $(U, \bar{U})_G$; jeho mohutnosť je $\lambda(G)$. K tomu stačí zistiť $(2n - 1)$ minimových rezov; minimový 1—$i$ rez a minimový $i$—1 rez pre $i = 2, 3, \ldots, n$, ak $G$ je digraf s vrcholmi $1, 2, \ldots, n$. (Ak $G$ je graf, tak $\lambda(1, i) = \lambda(i, 1)$, a preto stačí riešiť polovičný počet tokových úloh.) Zdôvodnenie je v tom, že alebo $1 \in U$ alebo $1 \in \bar{U}$, takže napr. v prvom prípade rez $(U, \bar{U})_G$ je 1—$i$ rezom pre nejaké $i$. Fordov–Fulkersonov algoritmus tu poskytuje $O(mn^2)$ algoritmus pre $\lambda(G)$. Avšak Even a Tarjan (1975) uvádzajú $O(mn^{5/3})$ algoritmus.

Pre vrcholovú súvislosť $\varkappa(G)$ nevieme takto zredukovať počet tokových úloh (lebo ak by vrchol 1 patril do minimového vrcholového rezu, tak by sme dostali falošný výsledok). Teda riešiac $O(n^2)$ tokových úloh, dostávame $O(mn^3)$ algoritmus. Even a Tarjan (1975) tu dávajú $O(m^2n^{1/2})$ algoritmus.

Pripomeňme, že zistiť, či $\varkappa(G) \geq 1$ alebo 2 možno v čase $O(m + n)$ (pozri 3.A, B), a tak je to aj pre zistenie 3-súvislosti [Hopcroft, Tarjan (1973a)].

Konečne Even (1975) dáva $O(kn^3)$ algoritmus na zistenie, či $\varkappa(G) \geq k$, ak $k < n^{1/2}$.

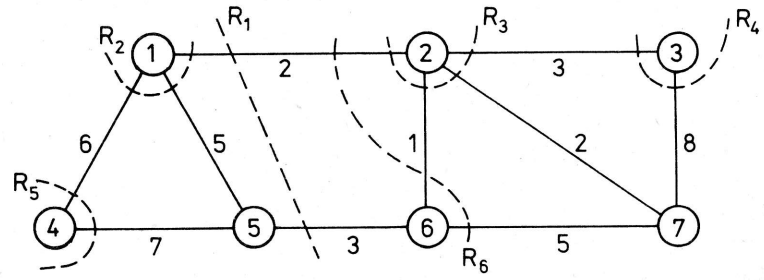
## 7.B
## Hranové súvislosti medzi každými dvoma vrcholmi grafu

V predchádzajúcej časti sme videli, že na určenie hranovej súvislosti $\lambda(G)$ grafu $G$ stačí riešiť $n-1$ úloh o maximálnom toku. Gomory a Hu (1961) ukázali, že toľko úloh stačí riešiť aj na zistenie všetkých hranových súvislostí $\lambda(i, j)$ pre $i, j \in V(G)$. Toto ukázali všeobecne pri reálnych kapacitách rebier $b_{ij} \geq 0$. Potom namiesto $\lambda(i, j)$ sa uvažuje veľkosť $f_{ij}$ maximálneho $i$—$j$ toku. Zisťovanie čísel $f_{ij}$ sa často nazýva *analýza grafu*. Obrátenie tejto úlohy je tzv. *syntéza grafu*, kde ide o navrhnutie grafu a kapacít jeho rebier tak, aby umožňoval toky predpísaných veľkostí, čo má význam pri navrhovaní komunikačných sietí. Najprv sa venujeme analýze a potom syntéze.

### ANALÝZA GRAFU

Gomory a Hu (1961) analyzujú graf $G$ postupným riešením tokových úloh a kontrahovaním vrcholov. Postupne zostrojujú strom $T$ s kapacitami rebier, ktorý má tie isté veľkosti $f_{ij}$ ako pôvodný graf $G$. Hrany stromu $T$ zodpovedajú $n-1$ „oddeleným" hranovým rezom grafu $G$.

Hovoríme, že rez $R = [A, B]_G$ *pretína* množinu vrcholov $M \subset V_G$, ak $A \cap M \neq \emptyset$ aj $B \cap M \neq \emptyset$. Tiež budeme hovoriť, že dva rezy $R_1 = [A_1, B_1]_G$ a $R_2 = [A_2, B_2]_G$ sa *pretínajú*, ak všetky štyri množiny $A_1 \cap A_2$, $A_1 \cap B_2$, $B_1 \cap A_2$ a $B_1 \cap B_2$ sú neprázdne. (V diagrame grafu zobrazíme rez $R_i$ čiarou oddeľujúcou




Obr. 1. Graf s kapacitami rebier a s vyznačenými minimovými rezmi.





$A_i$ a $B_i$, $i = 1, 2$. Ak sa rezy $R_1$ a $R_2$ nepretínajú, tak ich možno zobraziť nepretínajúcimi sa čiarami.)

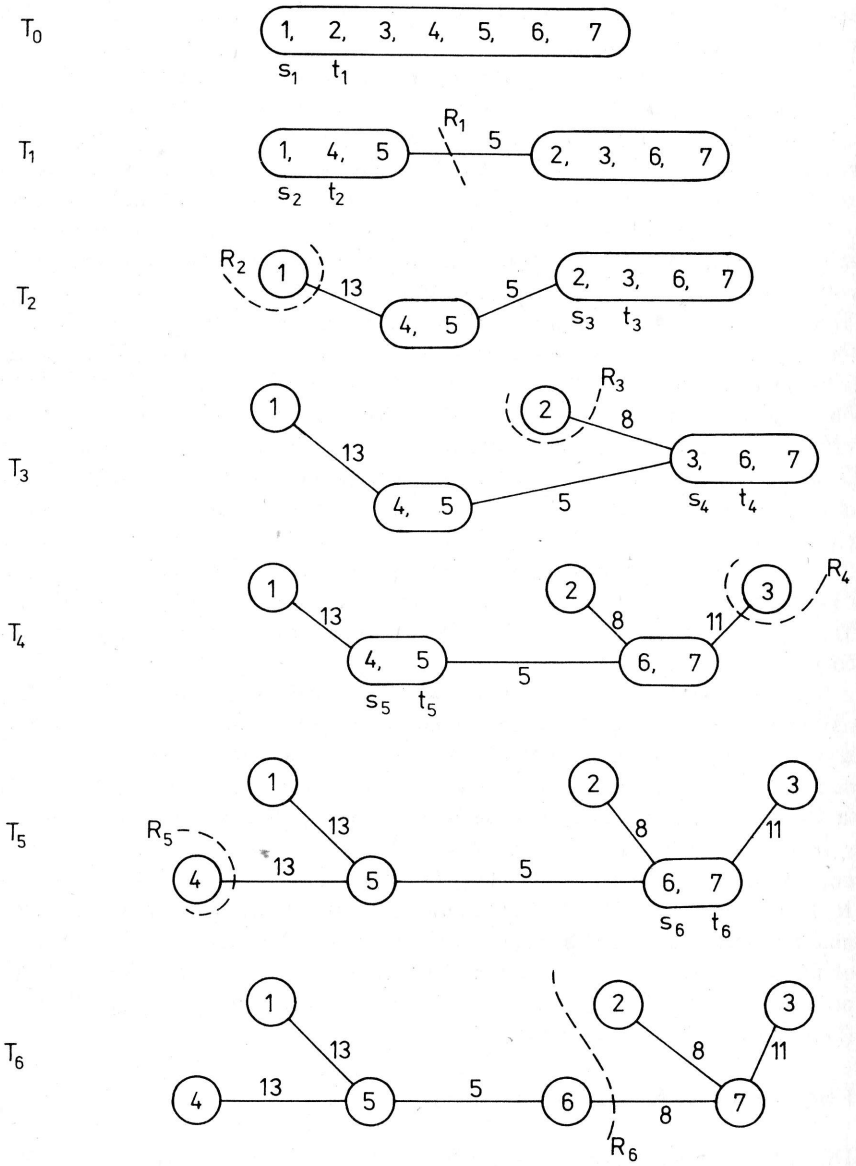
Analýzu grafu najprv ilustrujeme na príklade a potom postup zdôvodníme.

Nech $G$ je graf z obr. 1 s vyznačenými kapacitami rebier (prerušované čiary a symboly $R_i$ označujú rezy, ktoré získame neskôr). Tento graf je reprezentovaný stromom $T_0$ s jedným vrcholom, ktorý je vlastne „obálkou" pre $V(G)$ (pozri obr. 2). Potom zvolíme dva vrcholy $s_1$, $t_1$ z obálky a nájdeme nejaký minimový $s_1$—$t_1$ rez $R_1 = [U_1, \bar{U}_1]_G$, čím získame z $T_0$ strom $T_1$ s dvoma vrcholmi-obálkami: jedna z nich pozostáva z $U_1$ a druhá z $\bar{U}_1$. Obálky spojíme hranou s kapacitou rovnou kapacite $b(R_1)$ rezu $R_1$ (pozri obr. 2, kde $s_1 = 1$, $t_1 = 2$, $U_1 = \{1, 4, 5\}$, $b(R_1) = 5$, rez $R_1$ tvoria hrany, ktoré pretína prerušovaná čiara označená symbolom $R_1$). V strome $T_1$ zvolíme ľubovoľne dva vrcholy $s_2$, $t_2$ z tej istej obálky $C$ a nájdeme minimový $s_2$—$t_2$ rez $R_2$ v grafe $G$, avšak taký, ktorý nepretína druhú obálku stromu $T_1$. To sa zaručí tak, že hľadáme $s_2$—$t_2$ rez $R_2$ v grafe $\hat{G}$, ktorý vznikne kontrakciou množiny vrcholov $V_1 = \{2, 3, 6, 7\}$ (druhá obálka) do jediného vrcholu (prípadné vzniknuté násobné hrany môžeme nahradiť jednoduchými, ak im dáme sumárne kapacity). Neskôr ukážeme, že kapacita rezu $R_2$ sa tým nezmení. Keďže v reze $R_2 = [U_2, \bar{U}_2]_G$ je obálka $V_1$ v $\bar{U}_2$, tak v novom strome $T_2$ bude spojená do obálky $\{4, 5\}$ (hranou kapacity 5 ako v strome $T_1$). Kapacita $b(R_2)$ rezu $R_2$ sa vyznačí na hrane spájajúcej dve časti (nové obálky) starej obálky. Takto postupujeme ďalej aj vo všeobecnosti, až získame strom $T_{n-1}$ s $n$ obálkami, čo je výsledok postupu.

Všeobecný krok možno dobre ilustrovať na prechode od $T_5$ k $T_6$: V strome $T_{k-1}$ nájdeme obálku $C$ s aspoň dvoma vrcholmi, z ktorých zvolíme $s_k$ a $t_k$ (u nás $s_6 = 6$, $t_6 = 7$). V lese $T_{k-1} - C$ je niekoľko komponentov, ktorých vrcholové množiny nech sú $V_1, V_2, \ldots, V_q$ (u nás $q = 3$: $V_1 = \{1, 4, 5\}$, $V_2 = \{2\}$, $V_3 = \{3\}$). V grafe $G$ kontrahujeme každú množinu $V_p$ na jediný vrchol $v_p$, čím získame (multi)graf $\hat{G}$, v ktorom nájdeme minimový $s_k$—$t_k$ rez $\hat{R}_k = [\hat{U}_k, \bar{\hat{U}}_k]_{\hat{G}}$. Tento rez zodpovedá nejakému $s_k$—$t_k$ rezu $R_k = [U_k, \bar{U}_k]_G$ tej istej kapacity (neskôr dokážeme, že aj $R_k$ je minimový rez). Strom $T_k$ vznikne z $T_{k-1}$ takto: Namiesto obálky $C$ budeme mať dve obálky $C \cap U_k$ a $C \cap \bar{U}_k$, ktoré spojíme hranou kapacity $b(R_k)$. Ostatné obálky budú ako v $T_{k-1}$; pritom hrana, ktorá viedla z nejakej množiny $V_p$ do $C$, povedie teraz do $C \cap U_k$, ak $V_p \subset U_k$, resp. povedie do $C \cap \bar{U}_k$, ak $V_p \subset \bar{U}_k$ (kapacita hrany sa nemení). ●

**Lema 1(a).** Rez $R_k$ je minimový $s_k$—$t_k$ rez v grafe $G$.

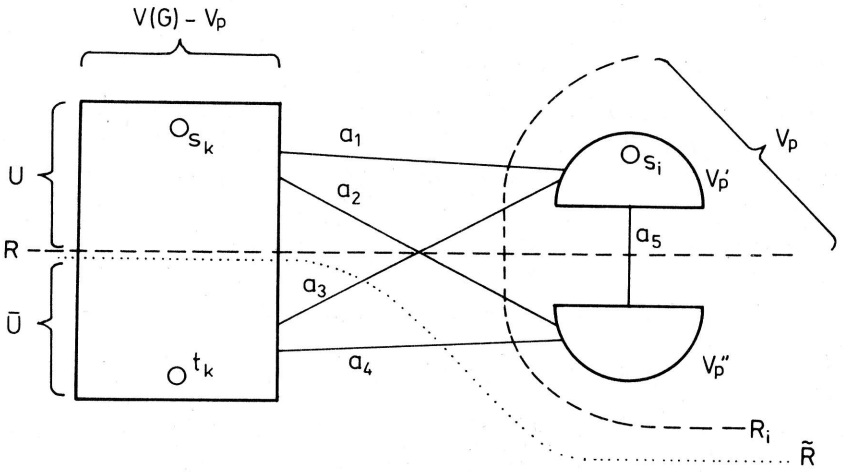
DÔKAZ. Stačí dokázať, že z ľubovoľného minimového $s_k$—$t_k$ rezu $R$ možno vyrobiť taký minimový $s_k$—$t_k$ rez, ktorý nepretína žiadnu množinu $V_p$. Taký rez možno




Obr. 2. Generovanie ekvivalentného stromu $T_6$ ku grafu z obr. 1.








Obr. 3. Nahradenie rezu $R$ rezom $\tilde{R}$.

utvoriť opakovaním konštrukcie rezu $\tilde{R}$ z obr. 3. Tu $s_k$—$t_k$ rez $R$ pretína množinu $V_p$, a tým ju rozdeľuje na dve časti $V_p'$ a $V_p''$. Množina $V_p$ súvisela s obálkou $C$ cez nejakú hranu, ktorá zodpovedá nejakému rezu $R_i$ v $G$, ktorým sme v minulosti oddelili $V_p$ od $V(G) - V_p$. Preto vo $V_p$ leží práve jeden z vrcholov $s_i$, $t_i$. V grafe majú oba symetrickú funkciu, a preto môžeme predpokladať, že $s_i \in V_p$ a $t_i \in V(G) - V_p$. Symetria obr. 3 nám dovoľuje predpokladať, že $s_i \in V_p'$ (ak $s_i \in V_p''$, tak $\tilde{R}$ bude „nad" množinou $V_p$).

Nech sumárna kapacita rebier spájajúcich: $U \cap (V(G) - V_p)$ a $V_p'$ je $a_1$, $U \cap (V(G) - V_p)$ a $V_p''$ je $a_2$, $\bar{U} \cap (V(G) - V_p)$ a $V_p'$ je $a_3$, $\bar{U} \cap (V(G) - V_p)$ a $V_p''$ je $a_4$, $V_p'$ a $V_p''$ je $a_5$ (pozri obr. 3). Potom z toho, že rez $R_i$ je minimový máme:

$$a_1 + a_2 + a_3 + a_4 \leq a_1 + a_3 + a_5,$$

odkiaľ získame:

$$a_3 + a_4 \leq a_3 + a_5 - a_2 \leq a_2 + a_3 + a_5.$$

Preto $\tilde{R} = [U \cup V_p'', \bar{U} - V_p'']_G$ bude tiež minimovým $s_k$—$t_k$ rezom v $G$ (vzhľadom na množinu $V(G) - V_p$ sú $R$ a $\tilde{R}$ rovnaké).

Rez $\tilde{R}$ pretína o jednu množinu $V_j$ menej ako rez $R$, a preto opakovaním tohto postupu možno získať minimový $s_k$—$t_k$ rez nepretínajúci žiadnu z množín $V_j$. Keďže rez $R_k$ je minimový spomedzi $s_k$—$t_k$ rezov takejto vlastnosti, tak je minimový vôbec. ■

**Lema 1(b).** Pre ľubovoľné vrcholy $v_1, v_2, \ldots, v_p$ grafu $G$ platí:

$$f_{v_1v_p} \geq \min\{f_{v_1v_2}, f_{v_2v_3}, \ldots, f_{v_{p-1}v_p}\}.$$

DÔKAZ. Najprv nech $p=3$. Podľa Fordovej–Fulkersonovej vety existuje $v_1$—$v_3$ rez $[U, \bar{U}]_G$ kapacity $b := f_{v_1v_3}$. Ak $v_2 \in U$, tak $f_{v_2v_3} \leq b$ a ak $v_2 \in \bar{U}$, tak $f_{v_1v_2} \leq b$. Preto $f_{v_1v_3} \geq \min\{f_{v_1v_2}, f_{v_2v_3}\}$. Požadovaná nerovnosť ihneď vyplýva indukciou. ■

**Veta 1.** Veľkosť $f_{ij}$ maximálneho $i$—$j$ toku v grafe $G$ je rovná veľkosti $f'_{ij}$ maximálneho $i$—$j$ toku v strome $T_{n-1}$, t. j.

$$f_{ij} = f'_{ij} = \min\{b(e) | e \text{ je hrana (jedinej) } i\text{—}j \text{ cesty v } T_{n-1}\},$$

kde $b(e)$ označuje kapacitu hrany $e$ v strome $T_{n-1}$.

DÔKAZ. Nech $f'_{ij} = b(e)$. Hrana $e$ v strome $T_{n-1}$ zodpovedá nejakému rezu $R_k = [U_k, \bar{U}_k]_G$ s $i \in U_k$ a $j \in \bar{U}_k$. (Žiadne dva z rezov $R_p$ sa nepretínajú, a preto hrana $e$ bude zodpovedať rezu $R_k$ v $T_k$ aj v $T_{n-1}$.) Preto $f_{ij} \leq b(R_k) = b(e) = f'_{ij}$.

Obrátenú nerovnosť dokážeme tým, že ukážeme pre každé $k$:

(∗) Ak hrana $e$ spája v $T_k$ dve obálky $C$, $D$, existujú vrcholy $u \in C$ a $v \in D$ tak, že platí: $f_{uv} = b(e)$.

Z platnosti (∗) pre $T_{n-1}$ potom vyplýva, že $f_{ij} \geq f'_{ij}$ pre každé 2 vrcholy $i, j \in V(G)$. Skutočne: Ak $ij$ je hranou v $T_{n-1}$, tak podľa (∗) je $f_{ij} = f'_{ij}$. Inak nech $v_1v_2\ldots v_p$ je (jediná) $i$—$j$ cesta v $T_{n-1}$. Potom tvrdenie vyplýva z lemy 1(b).

Preto nám zostáva dokázať (∗). Tvrdenie zrejme platí pre $T_1$ a pre indukciu nech platí pre $T_{k-1}$. Pre hranu $e$ v $T_k$ spájajúcu dve obálky $C$ a $D$ sú tri možnosti:

(i) $C$ a $D$ boli obálkami v $T_{k-1}$: potom požadované vrcholy $u$ a $v$ zo stromu $T_{k-1}$ ponecháme.

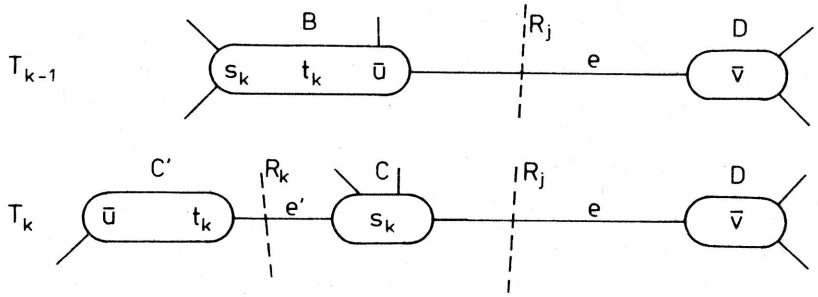
(ii) $C$ a $D$ vznikli rozdelením jednej obálky v $T_{k-1}$: potom bolo $s_k \in C$ a $t_k \in D$ (resp. obrátene), a preto môžeme položiť $u = s_k$ a $v = t_k$.

(iii) Jedna z obálok, povedzme $C$, vznikla rozdelením nejakej obálky $B$ v $T_{k-1}$ na $C'$ a $C$, ale $D$ bolo obálkou v $T_{k-1}$. Teda v $T_{k-1}$ hrana $e$ spájala $B$ a $D$: nech $\bar{u} \in B$, $\bar{v} \in D$ boli také, že $f_{\bar{u}\bar{v}} = b(e)$. Ak $\bar{u} \in C$, tak zrejme môžeme položiť $u = \bar{u}$, $v = \bar{v}$. Takto zostáva prípad, keď $\bar{u} \in C'$. Môžeme predpokladať, že $t_k \in C'$ a $s_k \in C$ (pozri obr. 4). Dokážeme, že možno zvoliť $u = s_k$ a $v = \bar{v}$.

Pretože rezy $R_k$ a $R_j$ sa nepretínajú, tak kapacity rezov $R_k$, $R_j$ sa nezmenia, ak všetky vrcholy v $C'$ pospájame hranami kapacity $\infty$. V takomto grafe $\tilde{G}$ môžeme písať: $\tilde{f}_{s_k\bar{u}} = b(e') = f_{s_k\bar{u}}$, $\tilde{f}_{t_k\bar{u}} = \infty$, $\tilde{f}_{\bar{u}\bar{v}} = b(e)$, a teda podľa lemy 1(b):








Obr. 4. K dôkazu vety 1.

$$\tilde{f}_{s_k\bar{u}} \geq \min\{\tilde{f}_{s_kt_k}, \tilde{f}_{t_k\bar{u}}, \tilde{f}_{\bar{u}\bar{v}}\} = \min\{b(e'), b(e)\}.$$

Pretože rez $R_k$ je minimový $t_k$—$s_k$ rez v $G$, tak existuje taký minimový $s_k$—$\bar{v}$ rez $R$ v $G$, ktorý nepretína $C'$. (Nech $R$ je ľubovoľný minimový $s_k$—$\bar{v}$ rez v $G$. Ak $R$ pretína $C'$, tak rovnako ako v dôkaze lemy 1(a), kde sme z rezu $R$ zostrojili rez $\tilde{R}$ pomocou rezu $R_i$ (pozri obr. 3), možno teraz pomocou rezu $R_k$ utvoriť z rezu $R$ minimový $s_k$—$\bar{v}$ rez $R'$ nepretínajúci $C'$.) Preto $\tilde{f}_{s_k\bar{v}} = f_{s_k\bar{v}}$. Takto z uvedeného máme:

$$f_{s_k\bar{v}} \geq \min\{b(e'), b(e)\}.$$

Avšak $b(e') \geq b(e)$, lebo $f_{\bar{u}\bar{v}} = b(e)$ a vrcholy $\bar{u}$ a $\bar{v}$ separuje rez $R_k$ kapacity $b(e')$. Takto $f_{s_k\bar{v}} \geq b(e)$, ale pretože rez $R_j$ kapacity $b(e)$ separuje $s_k$ a $\bar{v}$, tak $f_{s_k\bar{v}} = b(e)$. Preto nám vrcholy $u = s_k$ a $v = \bar{v}$ vyhovujú. ■

## SYNTÉZA GRAFU

Zaujímavejšia úloha ako analýza sa zdá byť *syntéza grafu:* Navrhnúť graf a kapacity hrán tak, aby umožňoval toky predpísaných veľkostí medzi každými dvoma vrcholmi. Aplikácie takejto úlohy pri konštruovaní komunikačných sietí sú zrejmé, ak pridáme požiadavku, aby sieť bola najlacnejšia. To vedie k úlohe:

Daná je symetrická matica $(r_{ij})$ typu $(n, n)$ požiadaviek na $i$—$j$ toky $(r_{ii} = 0)$ a symetrická matica $(c_{ij})$ cien za jednotkové kapacity. Úlohou je navrhnúť kapacity $b_{ij} \geq 0$ hrán kompletného grafu $G$ tak, aby pre každé dva vrcholy $i, j$ v $G$ existoval $i$—$j$ tok veľkosti aspoň $r_{ij}$, t. j. $f_{ij} \geq r_{ij}$ a pritom aby

$$\sum_{i,j \in G} c_{ij}b_{ij}$$

bolo minimálne.

Navrhnuté postupy na riešenie tejto úlohy (napr. [Gomory, Hu (1962)]), nie sú dosť efektívne. Avšak v špeciálnom prípade, keď všetky ceny $c_{ij}$ sú rovnaké, úlohu

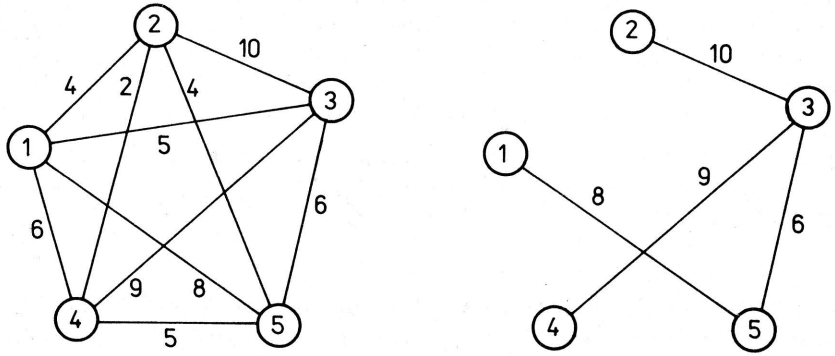
vieme dobre riešiť. Vtedy môžeme predpokladať, že všetky $c_{ij} = 1$, a teda účelom je minimalizovať súčet kapacít $b_{ij}$. Nasledujúci algoritmus [Gomory, Hu (1961)] navrhne kapacity $b_{ij}$ tak, že v nasledujúcich vzťahoch budú rovnosti.

(*) $$\sum_j b_{ij} \geq \max_j \{r_{ij}\} \quad (i = 1, 2, \ldots, n).$$

Keďže každé prípustné riešenie musí spĺňať (*), tak je zrejmé, že získame optimálne riešenie.

Významné je pozorovanie, ktoré sme získali pri analýze grafu, že namiesto požiadaviek pre kompletný graf $K_n$ stačí uspokojiť požiadavky na nejakej (ľubovoľne zvolenej) maximovej kostre $T$ (t. j. najlacnejšia kostra pri ohodnoteniach $-r_{ij}$). Maximum z požiadaviek pri každom vrchole $i$ je v $K_n$ a v $T$ rovnaké (inak by kostra $T$ nebola maximová).

Ďalší postup ilustrujeme na príklade z obr. 5, kde je kompletný graf $K_5$ s vyznačenými požiadavkami $r_{ij}$ a maximová kostra $T$. Túto kostru $T$ potom rozložíme na „rovnomerné" stromy, t. j. stromy, v ktorých má každá hrana



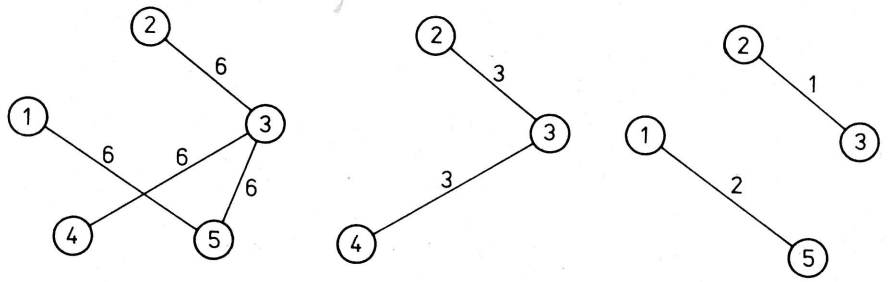

Obr. 5. Tokové požiadavky a maximová kostra $T$.

rovnakú požiadavku (obr. 6). Prvý z nich bude kostra s požiadavkou rovnou minimálnej požiadavke v $T$ (u nás 6). Po odčítaní tejto hodnoty nenulové hrany tvoria les, ktorého každý netriviálny strom týmto spôsobom rozkladáme na rovnomerné stromy atď. Takto získaný rozklad je zrejme jednoznačný.

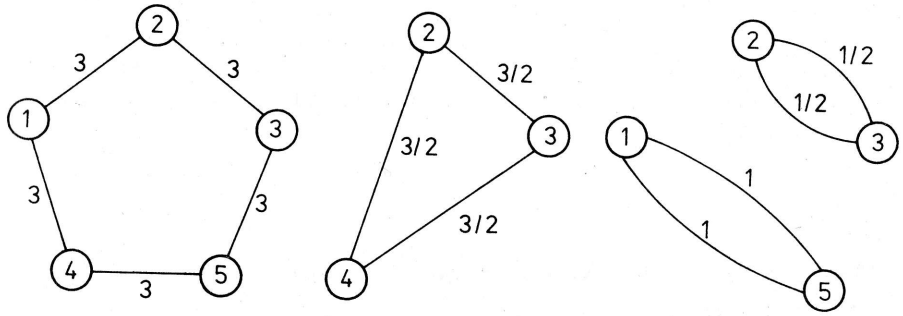
Ďalej každý rovnomerný strom nahradíme cyklom prechádzajúcim všetkými vrcholmi stromu (v ľubovoľnom poradí) a na každú jeho hranu dáme ohodnotenie rovné polovičnej rovnomernej požiadavke príslušného stromu (pozri obr. 7). (Výnimku by mohli tvoriť stromy s jednou hranou, ako uvidíme z nasledujúceho kroku, kde každý 2-cyklus zmeníme na pôvodnú hranu.)








Obr. 6. Rozklad kostry $T$ na rovnomerné stromy.



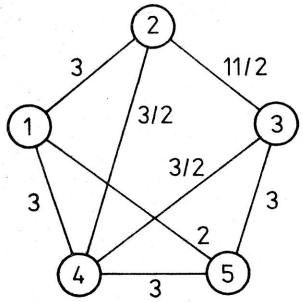

Obr. 7. Nahradenie rovnomerných stromov cyklami.

Konečným krokom je superpozícia všetkých cyklov do jedného grafu $G$, kde každá hrana $ij$ dostane ohodnotenie $b_{ij}$ rovné súčtu ohodnotení hrany $ij$ vo všetkých cykloch (pozri obr. 8). ●

**Veta 2.** Graf $G$ s kapacitami $b_{ij}$ získaný vyššie uvedeným postupom je optimálnym riešením úlohy syntézy.

DÔKAZ. Z postupu vidíme, že ak zabezpečíme tokové požiadavky každého rovnomerného stromu zvlášť, tak aj pre kostru $T$ (súčet maximálnych $i$—$j$ tokov v jednotlivých rovnomerných stromoch je zrejme $i$—$j$ tokom v $T$). Avšak v zodpovedajúcich cykloch možno viesť $i$—$j$ tok tej istej veľkosti, a teda požadovaný sumárny tok možno viesť aj v $G$.

Optimalita vyplýva z toho, že vo vzťahoch (*) budú rovnosti. ■




Obr. 8. Finálny graf $G$ získaný superpozíciou cyklov z obr. 7.

Poznamenajme, že ak sú všetky požiadavky $r_{ij}$ celé čísla, tak uvedený algoritmus zabezpečí, že všetky $b_{ij}$ budú celočíselné násobky čísla 1/2. Problém nájsť celočíselné optimum zostáva otvorený, avšak problém s (nerovnakými) cenami $c_{ij}$ je tu NP-ťažký (7.Z.7).

## 7.C
## Maximálny počet disjunktných zdrojových kostier

Istou mierou súvislosti nejakého vrcholu $v_0$ s ostatnými môže byť počet (hranovo) disjunktných zdrojových kostier so zdrojom $v_0$.

Nech $G$ je multidigraf a $v_0$ nejaký jeho vrchol. Ak $S \subset V(G)$ a $v_0 \in S$, tak množinu hrán $(S, V(G)-S)_G$ označíme $E^+_G(S)$ a nazývame $v_0$-rez. Píšeme: $h_G(S) = |E^+_G(S)|$.

Nasledujúcu vetu objavil Edmonds (1973), avšak jeho dôkaz nedával potrebný algoritmus. Tu sledujeme dôkaz algoritmickej povahy, ktorý dal Lovász (1976).

**Veta 1.** Maximálny počet hranovo-disjunktných kostier so zdrojom $v_0$ je rovný minimálnemu počtu hrán vo $v_0$-reze.

DÔKAZ. Jedným smerom je tvrdenie zrejmé. Preto iba dokazujeme, že ak $h_G(S) \geq k$ pre každé $S \subset V(G)$, $v_0 \in S$, tak v $G$ existuje $k$ hranovo-disjunktných kostier so zdrojom $v_0$. Postupujeme indukciou podľa $k$.

Nech $Q = (U, F) \subset G$ je strom so zdrojom $v_0$ a taký, že

(*) $\qquad h_{G-F}(S) \geq k-1$, pre každé $S \subset V(G)$, kde $v_0 \in S$.





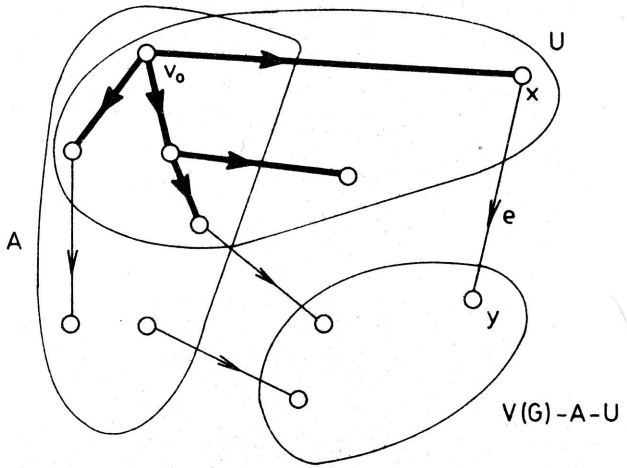
Ak $U = V(G)$, tak $Q$ je kostra v $G-F$, čo spolu s $k-1$ kostrami v $G-F$ (existujúcimi podľa indukčného predpokladu) je $k$ kostier. Preto nech $U \subset V(G)$. Ukážeme, že existuje šíp $e \in E^+_G(U)$ tak, že nový strom $Q+e$ ešte stále spĺňa (*). Takto postupne získame z $Q$ kostru.

Najprv zvoľme ľubovoľnú hranu $e \in E^+_G(U)$. Zrejme $Q+e$ bude strom so zdrojom $v_0$. Preto kľúčovou zostáva podmienka (*). Ak pre každé $S \subset V(G)$, kde $v_0 \in S$, je $h_{G-F-e}(S) \geq k-1$, tak máme požadovanú hranu $e$ nájdenú.

V opačnom prípade pre nejaké $A \subset V(G)$ s $v_0 \in A$ platí: $h_{G-F-e}(A) = k-2$. Keďže hrana $e$ odchádza aj z množiny $U$ aj z $A$, tak odchádza aj z $A \cup U$, t. j., $e \in E^+_{G-F}(A \cup U)$, takže $A \cup U \subset V(G)$ a zrejme $h_{G-F}(A) = k-1$. Budeme predpokladať, že $A$ je maximálna (v zmysle inklúzie) množina spĺňajúca podmienku

(**) $\qquad v_0 \in A, \quad A \cup U \subset V(G)$ a $h_{G-F}(A) = k-1$.

Situáciu znázorňuje obr. 1. Keďže $h_{G-F}(A \cup U) = h_G(A \cup U) \geq k$, tak $A \cup U \neq A$, $U \not\subseteq A$ a existuje šíp $e \in E^+_{G-F}(A \cup U) - E^+_{G-F}(A)$: nech $e = xy$. Zrejme $x \in U - A$ a $y \in V(G) - A - U$. Dokážeme, že strom $Q' = (U \cup \{y\}, F \cup \{e\})$ so zdrojom $v_0$ spĺňa (*).




Obr. 1. K dôkazu vety 1.

Ak $e \notin E^+_G(S)$, tak potom $h_{G-F-e}(S) = h_{G-F}(S) \geq k-1$. Preto nech $e \in E^+_G(S)$. Potom $x \in S$, $y \in V(G) - S$.

Vo všeobecnosti platí nerovnosť $h(S \cup A) + h(S \cap A) \leq h(S) + h(A)$, ktorú použijeme. U nás máme $h_{G-F}(A) = k-1$, $h_{G-F}(S \cap A) \geq k-1$ (indukčný

predpoklad). Keďže $x \in S$, $x \in U - A$ a $y \in V(G) - S$, $y \in V(G) - U - A$, tak $S \cup A \neq A$ a $S \cup A \subset V(G)$. Pretože $A$ je maximálna množina spĺňajúca (**), tak $h_{G-S}(S \cup A) \geq k$. Potom dosadením do spomínanej všeobecnej nerovnosti získame $h_{G-F}(S) \geq k$. Preto $h_{G-F-e}(S) \geq k-1$, čo sme chceli dokázať. ■

Na základe vety 1 a jej dôkazu možno postaviť polynomiálny algoritmus pre nájdenie maximálneho počtu disjunktných zdrojových kostier. Prvé také algoritmy dávajú nezávisle Tarjan (1974) a Lovász (1976). Tu uvedieme Lovászov prístup, ktorý vychádza z predchádzajúceho dôkazu.

Algoritmus vyžaduje často zisťovať, či hodnota minimového $v_0$-rezu $r(\hat{G}) := \min\{h_{\hat{G}}(S) | S \subset V(\hat{G}), v_0 \in S\}$ v danom faktore $\hat{G} \subset G$ je aspoň $k-1$. To možno zistiť vyriešením úlohy o existencii $v_0$—$v$ toku v $\hat{G}$ veľkosti aspoň $k-1$ (pri jednotkových kapacitách hrán) pre každé $v \in V(\hat{G}) - \{v_0\}$. Fordov–Fulkersonov algoritmus tu vystačí s $O(km)$ operáciami na jednu úlohu, teda zistiť, či $r(\hat{G}) \geq k-1$ možno v čase $O(knm)$.

**Algoritmus**

KROK 0: Nájdeme $k := r(G)$, $G_k := G$.

KROK 1: $F_k := \emptyset$, $P_k := E(G_k)$, $U_k := \{v_0\}$. Ak $k = 0$, tak STOP: $F_1, F_2, \ldots, F_{r(G)}$ sú disjunktné kostry so zdrojom $v_0$ a je ich maximálny možný počet.

KROK 2: Ak $|F_k| = n-1$, tak $G_{k-1} := G_k - F_k$, $k := k-1$ a ideme na krok 1. Inak ideme na krok 3.

KROK 3: Zvolíme ľubovoľne $e = xy \in P_k \cap E^+_{G_k}(U_k)$, $P_k := P_k - \{e\}$. Ak $r(G_k - F_k - e) \geq k-1$, tak $F_k := F_k \cup \{e\}$, $U_k := U_k \cup \{y\}$ a ideme na krok 2.
Inak $P_k := P_k - \{e\}$ a ideme na krok 3. ●

Význam množiny $P_k$ spočíva v tom, že v kroku 3 testujeme každú hranu $e$ nanajvýš raz. Totiž ak zistíme pre strom $(U_k, F_k)$, že $r(G_k - F_k - e) \leq k-2$, tak zrejme pre väčší strom $(U'_k, F'_k)$ bude tiež $r(G_k - F'_k - e) \leq k-2$, a teda je to zbytočné overovať. Teda hranu $e$ môžeme z množiny prípustných hrán $P_k$ vynechať. Takto pre každé $k$, krok 3 prebehne nanajvýš $|E(G_k)| \leq m$ razy. Dominujúcou zložkou v kroku 3 je výpočet hodnoty minimového $v_0$-rezu: $O(knm)$ operácií, ako sme už uvádzali.

Potom na nájdenie jednej zdrojovej kostry stačí $O(km^2n)$ operácií a na nájdenie $k$ zdrojových kostier stačí $O(k^2m^2n)$ operácií.

Tarjan (1974) uvádza $O(k^2m^2)$ algoritmus a v práci [Tarjan (1976)] je pre $k=2$ takmer lineárny algoritmus.





## 7.D
## Najobjemnejší systém $k$ kostier

Hranová súvislosť $\lambda(u, v)$ dvoch vrcholov $u$ a $v$ je podľa Mengerovej vety rovná maximálnemu počtu hranovo disjunktných $u$—$v$ ciest. V časti 7.C sme skúmali „súvislosť" jedného vrcholu $v_0$ ku všetkým ostatným a hľadali maximálny počet (hranovo) disjunktných kostier so zdrojom $v_0$. Ešte lepšiu mieru „súvislosti" predstavuje maximálny počet silne súvislých faktorov, na ktoré možno daný graf (alebo digraf) rozložiť, lebo zahŕňa všetky vrcholy rovnako. Pre digrafy nebola táto otázka vyriešená, ale pre grafy dáme definitívnu odpoveď.

Úlohy takéhoto druhu vznikajú pri rozkladoch komunikačných sietí medzi viacerých užívateľov. (Napr. cestnú sieť chceme rozložiť medzi $k=3$ užívateľov: bicykle, osobné autá a nákladnú dopravu tak, aby sa okrem križovatiek nestretávali, pričom majú všetky základné body v každej parciálnej sieti súvisieť. Podobne možno rozkladať telegrafnú alebo inú sieť.) Takto sa dostávame k úlohe:

(a) Nájsť $k$ disjunktných kostier multigrafu $G$.

Tutte (1961) dokázal, že multigraf $G$ má $k$ disjunktných kostier práve vtedy, ak pre každé $r$ a každý rozklad $V_1, V_2, \ldots, V_r$ množiny $V(G)$ je počet hrán spájajúcich dve rôzne triedy aspoň $k(r-1)$. Iný dôkaz dal Nash–Williams (1961). Oba dôkazy poskytujú algoritmy na riešenie úlohy (a), hoci samotný výsledok priamo nevieme využiť. Príbuzná je úloha:

(b) Nájsť $k$ kostier multigrafu $G$, ktoré v zjednotení obsahujú všetky hrany.

Nash–Williams (1964) ukázal, že takých $k$ kostier existuje práve vtedy, ak $k \geq m(H)/(n(H)-1)$, pre každý netriviálny indukovaný podgraf $H \subseteq G$. Edmonds (1965a) dáva dobrý algoritmus pre úlohu (b) (v reči matroidov). Úloha (a) a úloha (b) sú zrejme špeciálne prípady nasledujúcej:

(c) Nájsť najobjemnejší systém $k$ kostier multigrafu $G$, t. j. nájsť takých $k$ kostier $T_1, T_2, \ldots, T_k$, že $|E(T_1) \cup E(T_2) \cup \ldots \cup E(T_k)|$, tzv. *objem*, je maximálny.

Pre $k=2$ má úloha (c) tiež význam pri analýze elektrických obvodov a jej riešenie dávajú Kishi a Kajitani (1969). Kameda (1976) zovšeobecnil ich úvahy a tu sledujeme jeho prácu.

Ak $S \subseteq E(G)$, tak označíme $\bar{S} := E(G) - S$. Nech $\{T_1, T_2, \ldots, T_k\}$ je nejaký systém $k$ kostier multigrafu $G$. Nech $C_i(e)$ (pre $e \in E(G)$) označuje množinu hrán (jediného) cyklu v $T_i + e$, ak $e \notin E(T_i)$; inak kladieme $C_i(e) = \emptyset$.

Predstavme si, že kostry $T_x$ a $T_1$ majú spoločnú hranu $e_1$ a nech $e_2$ je taká hrana, že $e_1 \in C_1(e_2)$ a $e_2 \in \overline{\cup E(T_i)}$. Potom zrejme zmenením kostry $T_1$ na susednú kostru $T_1 - e_1 + e_2$ získame objemnejší systém $k$ kostier. Ak by $e_2 \in \cup E(T_i)$, ale napr.

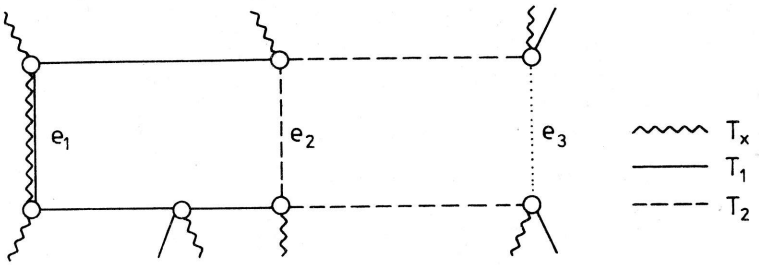
$e_2 \in C_2(e_3)$ kde $e_3 \in \overline{\cup E(T_i)}$, tak zmeniac $T_1$ na $T_1 - e_1 + e_2$ a $T_2$ na $T_2 - e_2 + e_3$ získame opäť objemnejší systém (pozri obr. 1), lebo $e_3$ pribudne do $\cup E(T_i)$. V týchto úvahách zrejme možno pokračovať. To vedie k zavedeniu funkcií $Z(S)$ a $Z^*(S)$ pre $S \subseteq E(G)$:

$$Z(S) = \bigcup_{e \in S} \bigcup_{i=1}^{k} C_i(e),$$

$$Z^*(S) = Z(S) \cup Z^2(S) \cup Z^3(S) \cup \ldots,$$

kde $Z^2(S) = Z(Z(S))$ atď.

V našom príklade $e_1 \in Z^2(\{e_3\})$.




Obr. 1. Zväčšenie objemu zmenou kostier na susedné.

Všimnime si, že $Z^*(S)$ možno vypočítať v polynomiálnom počte krokov. Totiž platí:

**Veta 1.** Pre každé $S \subseteq E(G)$ je

$$Z^*(S) = Z(S) \cup Z^2(S) \cup \ldots \cup Z^{kn}(S).$$

DÔKAZ. Ak označíme $S(i) := Z(S) \cup \ldots \cup Z^i(S)$, tak z definície $Z$ vidíme, že ak $S(i) = S(i+1)$ pre nejaké $i$, tak potom $S(i) = S(i+j)$ pre každé $j \geq 1$, a teda $S(i) = Z^*(S)$. Avšak $S(i) - S \subseteq \cup E(T_j)$ pre ľubovoľné $i$ a $|\cup E(T_j)| \leq kn$. Preto $S(kn) = S(kn+1) = Z^*(S)$. ■

Uvedené pozorovanie nás vedie k nasledujúcemu výsledku, kde pod *duplicitnou hranou* systému kostier rozumieme hranu ležiacu v aspoň dvoch kostrách.

**Veta 2.** Systém $k$ kostier $T_1, T_2, \ldots, T_k$ multigrafu $G$ je najobjemnejší práve vtedy, keď každá duplicitná hrana je mimo množiny $Z^*(\overline{\cup E(T_i)})$.

DÔKAZ. ($\Rightarrow$) Ak existuje duplicitná hrana $e_1 \in Z^*(\overline{\cup E(T_i)})$, tak existuje postupnosť hrán $e_1, e_2, \ldots, e_{r+1}$ a indexov $i_1, \ldots, i_r$ tak, že $e_p \in C_{i_p}(e_{p+1})$ pre $p = 1, 2, \ldots, r$





(porovnaj obr. 1). Po zmene kostry $T_{i_p}$ na kostru $T_{i_p} + e_{p+1} - e_p$ pre $p = 1, 2, \ldots, r$ (ostatné kostry zostanú nezmenené) získame objemnejší systém kostier, lebo hrana $e_{r+1}$ pribudne a žiadna nevypadne ($e_1$ zostane v aspoň jednej kostre, lebo bola duplicitná). Získaný spor dokazuje implikáciu.

($\Leftarrow$) Nech $\hat{G}$ je faktor multigrafu $G$ s $E(\hat{G}) = Z^*(\overline{\cup E(T_i)})$. Množinu $E(\hat{G})$ rozložíme na hrany ležiace v $\cup E(T_i)$ a v $\overline{\cup E(T_i)}$. Najprv však dokážeme, že pre každý komponent $H$ v $\hat{G}$ platí: $H \cap T_1, H \cap T_2, \ldots, H \cap T_k$ sú hranovo disjunktné kostry v $H$, kde $H \cap T_j := (V(H), E(H) \cap E(T_j))$. K tomu stačí ukázať, že ľubovoľné dva vrcholy $x$ a $y$ v $H \cap T_j$ súvisia. Keďže $H$ je súvislý, tak v ňom existuje nejaká $x$—$y$ cesta $P$. Tvrdíme, že pre každú hranu $e_r = x_r y_r$ cesty $P$ existuje $x_r$—$y_r$ cesta $Q_r$, ktorej všetky hrany ležia v $E(T_j) \cap Z^*(\overline{\cup E(T_i)})$. Ak $e_r \in E(T_j)$, tak je to zrejmé z definície $\hat{G}$. Inak za $Q_r$ stačí zobrať jedinú $x_r$—$y_r$ cestu v $T_j + e_r$ (potom $Q_r \subseteq Z(\overline{\cup E(T_i)})$). Tým je ukázané, že $x$ a $y$ súvisia v $H \cap T_j$, a teda $H \cap T_j$ je kostra v $H$. Disjunktnosť kostier $H \cap T_1, \ldots, H \cap T_k$ vyplýva z predpokladu, že duplicitné hrany systému $T_1, \ldots, T_k$ sú mimo $E(\hat{G})$.

Teraz množinu $E(\hat{G})$ rozložíme:

$$E(\hat{G}) = [E(\hat{G}) \cap (\overline{\cup E(T_i)})] \cup [E(\hat{G}) \cap (\cup E(T_i))] =$$

$$= [\overline{\cup E(T_i)}] \cup [\cup (E(\hat{G}) \cap E(T_i))],$$

lebo $\overline{\cup E(T_i)} \subseteq E(\hat{G})$. Nech $p$ je počet komponentov v $\hat{G}$, a nech $\hat{m} = |E(\hat{G})|$. Z posledného vzťahu potom máme: $\hat{m} = |\overline{\cup E(T_i)}| + k(n-p)$, lebo les s $p$ komponentmi má $n-p$ hrán a tieto lesy sú podľa uvedeného hranovo disjunktné.

Nech teraz $T_1', T_2', \ldots, T_k'$ je ľubovoľný iný systém $k$ kostier v $G$. Potom máme:

$$E(\hat{G}) = [E(\hat{G}) \cap (\overline{\cup E(T_i')})] \cup [E(\hat{G}) \cap (\cup E(T_i'))]$$

a odtiaľ

$$\hat{m} = |E(\hat{G})| \leq |\overline{\cup E(T_i')}| + k(n-p).$$

Porovnanie tejto nerovnosti s uvedenou rovnosťou dáva: $|\overline{\cup E(T_i)}| \leq |\overline{\cup E(T_i')}|$, a teda $|\cup E(T_i)| \geq |\cup E(T_i')|$, t. j. systém $T_1, \ldots, T_k$ je najobjemnejší. ■

Kameda navrhuje nasledujúci algoritmus pre najobjemnejší systém $k$ kostier, ktorý je založený na postupnom zväčšovaní systému $k$ kostier ako v prvej časti dôkazu vety 2.

**Algoritmus**

KROK 1: Nájdeme nejaký systém $k$ kostier $T_1, T_2, \ldots, T_k$ v $G$ (napr. nájdeme nejakú kostru $T_1$ a položíme $T_i := T_1$ pre všetky $i$).

KROK 2 (*overovanie optimality systému* podľa vety 2):

2.1: $M := \overline{\cup E(T_i)}$.

2.2: Ak $M = \emptyset$, tak STOP: systém $T_1, \ldots, T_k$ je najobjemnejší. Inak zvolíme ľubovoľne hranu $e \in M$, $M := M - \{e\}$, $S := \{e\}$, $R := \{e\}$.

2.3: $S := Z(S) - R$.

Ak $S$ obsahuje duplicitnú hranu systému $T_1, \ldots, T_k$, tak tento systém zmeníme na objemnejší (tak ako v prvej časti dôkazu vety 2), zahŕňajúci hranu $e$: označíme ho opäť ako $T_1, \ldots, T_k$ a ideme na krok 2.

Ak $S \not\subseteq R$, tak $R := R \cup S$ a ideme na krok 2.3. Inak ideme na krok 2.2. ●

Správnosť algoritmu prenechávame overiť čitateľovi (7.Z.9). Kameda (1976) uvádza, že vhodná implementácia dáva $O(k^2mn)$ algoritmus. Tarjan (1976) tvrdí, že má $O(k^2n^2)$ algoritmus.

## 7.Z
## Cvičenia a poznámky

1. Dokážte, že pravidelný bipartitný graf stupňa $k$ nemá hranový rez nepárnej mohutnosti menšej ako $k$.
2. Dokážte, že v grafe priemeru 2 je hranová súvislosť rovná minimálnemu stupňu vrcholu [Plesník (1975)]. Ukážte, že pre všetky multigrafy to neplatí.
3. Dokážte, že pre ľubovoľný graf $G$ platí: $m \leq (n-1)(n-2)/2 + \varkappa(G)$ [Gliviak (1975)].
4. Nech $G$ je hranovo $k$-súvislý multidigraf (t. j. $\lambda(G) \geq k$). Dokážte, že pre ľubovoľné dva vrcholy $s, t \in V(G)$ a každé $p$, $0 \leq p \leq k$, existuje $p$ $s$—$t$ ciest a $k-p$ $t$—$s$ ciest, ktoré sú všetky hranovo disjunktné. [Návod: Pridajte ku $G$ nový vrchol $v_0$, $p$ násobných šípov $v_0s$, $k-p$ násobných šípov $v_0t$ a ukážte, že vzniknutý multidigraf má $k$ disjunktných kostier so zdrojom $v_0$ [Even (1979)].]
5. (a) Ukážte, že maximálny počet vnútorne disjunktných $s$—$t$ ciest dĺžky nanajvýš $L$ môže byť aj menší, ako je minimálny počet vrcholov, ktorých





vynechanie z grafu poruší každú $s$—$t$ cestu dĺžky nepresahujúcej $L$ [Lovász et al. (1978)]. Teda vo všeobecnosti analógia Mengerovej vety neplatí, ale pre $L = d(s, t)$ platí.

(b) Ukážte, že pre hranovo disjunktné cesty je to podobné.

Itai et al. (1982) ukázali, že problém nájsť maximálny počet vnútorne (resp. hranovo) disjunktných $s$—$t$ ciest dĺžky nanajvýš $L$ je NP-ťažký v grafoch aj digrafoch už pre $L \geq 5$.

6. Dokážte, že ak má graf $G$ všetky stupne párne, tak má takú orientáciu $H$, že pre každé dva vrcholy $u, v$ platí: $\lambda_H(u, v) = \lambda_H(v, u) = \frac{1}{2}\lambda_G(u, v)$. [Návod: Súhlasne zorientujte eulerovský ťah v $G$ (časť 9.A).]

   Zosilnenie pre ľubovoľný multimigraf dáva Nash-Williams (1960).
7. Dokážte, že problém syntézy grafu (7.B) je NP-ťažký, ak ceny $c_{ij}$ nie sú rovnaké a žiadame nájsť celočíselné optimum (t. j. celočíselné $b_{ij}$), dokonca pre prípad, že všetky $r_{ij} = 2$. [Návod: Transformujte problém hamiltonovského cyklu.]
8. Nech je daný digraf $G$ s $n$ vrcholmi a $n$-tica $(k_1, k_2, \ldots, k_n)$ čísel. Navrhnite algoritmus pre nájdenie $k_1 + \ldots + k_n$ hranovo disjunktných kostier, kde $k_i$ z nich má zdroj vo vrchole $v_i$ $(i = 1, \ldots, n)$. [Návod: Ku $G$ pridáme nové vrcholy $v_0, u_1, \ldots, u_k, w_1, \ldots, w_k$ a šípy $v_0u_j$, $u_jw_t$, $u_jw_s$ pre všetky $j, t, s: j \neq t$ a šípy $w_1v_1, \ldots, w_{k_1}v_1$: $w_{k_1+1}v_2, \ldots, w_{k_1+k_2}v_2$: …: $w_{k_1+} \ldots + k_{n-1}+1v_n, \ldots, w_{k_1+\ldots k_n}v_n$. Potom hľadáme v novom digrafe kostry so zdrojom $v_0$.]
9. Dokážte správnosť algoritmu z časti 7.D.
10. Kundu (1974) dokázal, že pre maximálny počet $t(G)$ disjunktných kostier grafu $G$ platí: $(\lambda(G)-1)/2 \leq t(G) \leq \lambda(G)$. Ukážte nejaké triedy grafov, kde sa nerovnosti dosahujú.
11. Problém rozkladu grafu na súvislé faktory možno zosilniť a žiadať, aby faktory mali predpísané priemery. Dokážte, že ak možno rozložiť nejaký kompletný graf na faktory s danými priemermi, tak aj každý väčší kompletný graf možno rozložiť na faktory s takými priemermi [Bosák et al. (1968)].

    Skúmali sa aj iné variácie úlohy: predpísané polomery [Palumbíny, Znám (1973)], predpísané priemery aj polomery [Niepel (1980)], rozklady kompletného bipartitného grafu [Tomová (1977)] a i. Ak však vo všeobecnosti upustíme od kompletnosti grafov, tieto úlohy sú NP-ťažké [Plesník (1982)].

    Ako špeciálny druh rozkladu digrafu možno chápať aj zorientovanie daného grafu $G$ na digraf $H$ s minimálnym priemerom, resp. polomerom. Táto úloha je tiež NP-ťažká [Chvátal, Thomassen (1978)]. Nájdite taký graf $G$ priemeru 2, pre ktorý je $\mathrm{diam}(H) = 6$ (najmenší taký má $n = 7$, $m = 9$).

# 8
# PÁRENIA

Pod *párením* v grafe $G$ budeme rozumieť takú podmnožinu $X$ hranovej množiny, že každý vrchol z $G$ inciduje nanajvýš s jednou hranou z $X$. (Niekedy tiež hovoríme, že $X$ je *množina nezávislých hrán*.) Pre dané párenie $X$ v grafe $G$ sa každá hrana z $X$ nazýva *páriaca* a každá hrana z $E(G)-X$ *nepáriaca*. Ak vrchol inciduje s hranou párenia, tak sa nazýva *párený* alebo tiež *pokrytý* vrchol; inak sa vrchol nazýva *voľný* alebo tiež *nepárený (nepokrytý)*. Ak je každý vrchol párený, tak párenie $X$ sa nazýva *úplné* a vytvára 1-*faktor (lineárny faktor)* v $G$. Tento pojem sa zovšeobecňuje na faktory s ohraničeniami na stupne. Pre aplikácie má význam priradiť hranám ceny a uvažovať cenu párenia. Špeciálnou skupinou úloh o pároniach sú úlohy v bipartitných grafoch, ktoré však možno pretransformovať na tokové úlohy.

Okrem priamych úloh o pároniach (priraďovanie pracovníkov k strojom, utváranie priateľských dvojíc a pod.) existujú aplikácie nepriame (najkratšia cesta v grafe (4.D), čínsky poštár (9.B) a i.).

Budeme uvažovať dve úlohy (druhá je zovšeobecnením prvej):

Problém najpočetnejšieho párenia: V danom grafe treba nájsť párenie maximálnej mohutnosti.

Problém najcennejšieho párenia: V danom grafe, kde každá hrana $ij$ má cenu $c_{ij}$ (reálne číslo), treba nájsť párenie s maximálnou sumárnou cenou.

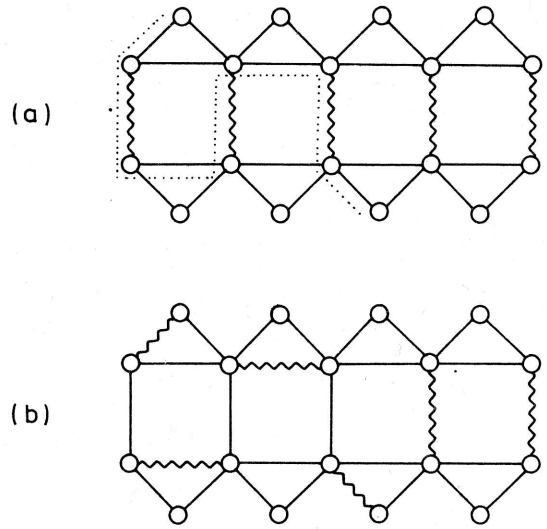
Najprv sa venujeme len problému najpočetnejšieho párenia.

## 8.A
## Alternujúce cesty

Veľmi ľahko možno nájsť tzv. *maximálne párenie*, t. j. párenie, ku ktorému už nemožno pridať žiadnu ďalšiu hranu. Pravda, mohutnosť maximálneho párenia sa môže značne líšiť od mohutnosti najpočetnejšieho párenia. Vhodným nástrojom na prerábanie párení je pojem *alternujúcej cesty*, t. j. cesty, v ktorej sa striedajú páriace a nepáriace hrany. Alternujúca cesta, ktorej oba kraje sú voľné, nazýva sa *zväčšujúca*. Na obr. 1(a) sú páriace hrany vlnovkové a bodkovane je vyznačená zväčšujúca cesta. Pomocou takejto cesty možno dané párenie zmeniť na párenie,







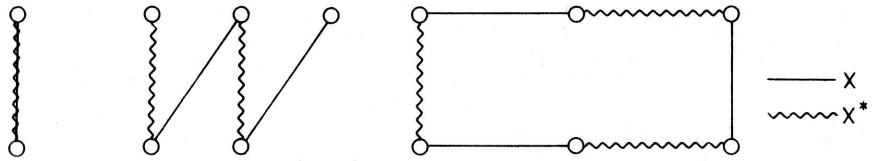

Obr. 1. Zmena párenia pomocou zväčšujúcej cesty.

ktoré má o jednu hranu viac (ako ilustruje obr. 1(b)), jednoducho vymenením páriacich a nepáriacich hrán cesty.

S myšlienkou zväčšujúcej cesty sa stretávame už u Petersena (1891), Egerváryho (1931) a i., ale plnú silu tohto pojmu uvádza až Berge (1957):

**Veta 1.** Párenie $X$ je najpočetnejšie $\Leftrightarrow$ neexistuje zväčšujúca cesta pre $X$.

DÔKAZ. ($\Rightarrow$) Týmto smerom je tvrdenie zrejmé. ($\Leftarrow$) Nech $X^*$ je nejaké najpočetnejšie párenie. Dokážeme, že $|X| = |X^*|$ a dôkaz bude hotový. Uvažujme podgraf $H = G\langle X \cup X^* \rangle$. Zrejme $H$ má len vrcholy stupňa 1 a 2, a preto každý jeho komponent má niektorý z troch tvarov z obr. 2, t. j.: (1) Jediná




Obr. 2. Komponenty z dôkazu vety 1.

hrana: keďže neexistuje zväčšujúca cesta pre $X$ ani pre $X^*$ (podľa prvej časti tvrdenia), tak táto hrana patrí do $X \cap X^*$. (2) Alternujúca cesta párnej dĺžky (alternujúca cesta nepárnej dĺžky by bola zväčšujúca alebo pre $X$ alebo pre $X^*$). (3) Alternujúci cyklus (teda párnej dĺžky). Takto každý komponent obsahuje rovnaký počet hrán z $X$ ako z $X^*$. Preto $|X| = |X^*|$. ■

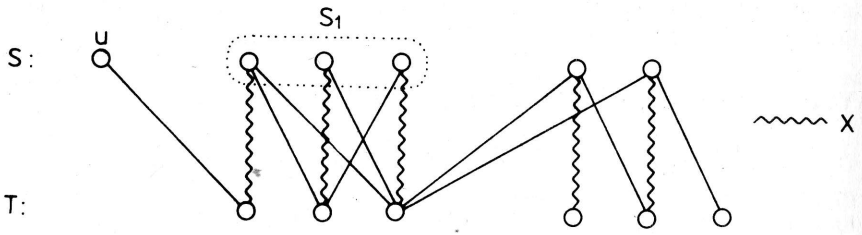
Užitočnosť zväčšujúcich ciest demonštrujeme ešte na nasledujúcej vete, ktorú objavili König (1931) a Hall (1935) a možno ju interpretovať takto: Zo systému množín $S_1, S_2, \ldots, S_p$ možno vybrať *p rozličných reprezentantov* (t. j. prvky $e_1 \in S_1, \ldots, e_p \in S_p$ tak, že $e_i \neq e_j$ pre $i \neq j$) práve vtedy, keď pre každé $k$ zjednotenie ľubovoľných $k$ množín obsahuje aspoň $k$ prvkov. Nech

$$V_G(A) := \bigcup_{a \in A} V_G(a) \text{ pre } A \subseteq V(G).$$

**Veta 2.** (König–Hall). V bipartitnom grafe (aj multigrafe) $G = (S, T; E)$ existuje párenie pokrývajúce $S \Leftrightarrow$ pre každé $A \subseteq S$ platí: $|A| \leq |V_G(A)|$.

DÔKAZ. ($\Rightarrow$) Hrany párenia zabezpečujú požadovanú nerovnosť.

($\Leftarrow$) Nech $X$ je najpočetnejšie párenie v $G$ a nech existuje vrchol $u \in S$ nepokrytý s $X$ (pozri obr. 3). Nech $S_1 \subseteq S$ je množina vrcholov dosiahnuteľných z $u$ po alternujúcich cestách nenulovej dĺžky. Zrejme $V_{G\langle X\rangle}(S_1) = V_G(S_1 \cup \{u\})$ (podľa definície $S_1$). Potom však pre $A = S_1 \cup \{u\}$ máme spor s predpokladom vety. ■




Obr. 3. K dôkazu Königovej-Hallovej vety.

Königova–Hallova veta má široké použitie pri odvodzovaní rôznych tvrdení (pozri cvičenia). Už sme ju použili napr. v 5.C. Gallai (1963) pomocou nej dokazuje aj nasledujúci pekný výsledok [Tutte (1947)], ktorý tu nedokazujeme.

**Veta 3** (Tutte). Graf (aj pseudograf) $G$ má 1-faktor $\Leftrightarrow$ pre každú podmnožinu $P \subseteq V(G)$ je počet nepárnych komponentov grafu $G - P$ nanajvýš $|P|$. (Komponent nazývame nepárny, ak má nepárny počet vrcholov.)





## 8.B
## Hľadanie najpočetnejšieho párenia

Zo všeobecného problému najprv vyčleníme úlohu v bipartitných grafoch, ktorá je riešiteľná pomocou tokov.

### PÁRENIA V BIPARTITNÝCH GRAFOCH

Táto úloha sa ľahko zvládne pomocou tokov: Daný bipartitný graf $G = (S, T; E)$ doplníme dvoma vrcholmi $s$ a $t$, pridáme šíp $su$ pre každé $u \in S$, šíp $vt$ pre každé $v \in T$ a pôvodné rebrá zorientujeme smerom z $S$ do $T$. Potom každému šípu vzniknutého digrafu $\hat{G}$ predpíšeme kapacitu 1. Je zrejmé, že maximálny celočíselný $s$—$t$ tok v $\hat{G}$ určuje nejaké najpočetnejšie párenie v $G$ (tvoria ho nasýtené hrany $uv$, kde $u \in S$, $v \in T$) a naopak.

To je jedna z mnohých aplikácií tokov, ktoré možno nájsť v [Ford, Fulkerson (1962)]. Nasledujúce tvrdenie sa zvyčajne pripisuje Königovi (1931) a Egerváry-mu (1931), hoci sa našiel aj skorší prameň [Frobenius (1912)] s príbuzným výsledkom v reči matíc. Pripomeňme, že vrchol pokrýva hranu a obrátene, ak sú incidentné.

**Veta 1** (König–Egerváry). V bipartitnom grafe je maximálny počet nezávislých hrán rovný minimálnemu počtu vrcholov pokrývajúcich (spolu) všetky hrany.

DÔKAZ. Uvážme uvedený digraf $\hat{G}$ a aplikujme Mengerovu vetu 7.A.1(b). Zrejme vrcholový $s$—$t$ rez v $\hat{G}$ je vrcholovým pokrytím hrán v $G$ a obrátene. Vnútorne disjunktné $s$—$t$ cesty v $\hat{G}$ zodpovedajú nezávislým hranám v $G$ (cesta $su_iv_jt$ korešponduje s hranou $u_iv_j$). ■

Pripomeňme, že pomocou tokov vieme nájsť aj takéto minimové pokrytie. Ak však upustíme od bipartitnosti, tak problém je NP-ťažký (veta 2.D.7).

### EDMONDSOVA METÓDA

Ak graf $G$ nie je bipartitný, tak metódy tokov zlyhávajú. Možno sa však oprieť o Bergeovu vetu a dané párenie postupne meniť pomocou zväčšujúcich alternujúcich ciest, až získame najpočetnejšie párenie. Týmto je však úloha len zdanlivo vyriešená, lebo sme nedali metódu na hľadanie zväčšujúcich ciest. Edmonds (1965) bol prvý, ktorý túto prekážku zdolal. Dolu uvádzame základné myšlienky jeho metódy pre hľadanie najpočetnejšieho párenia.

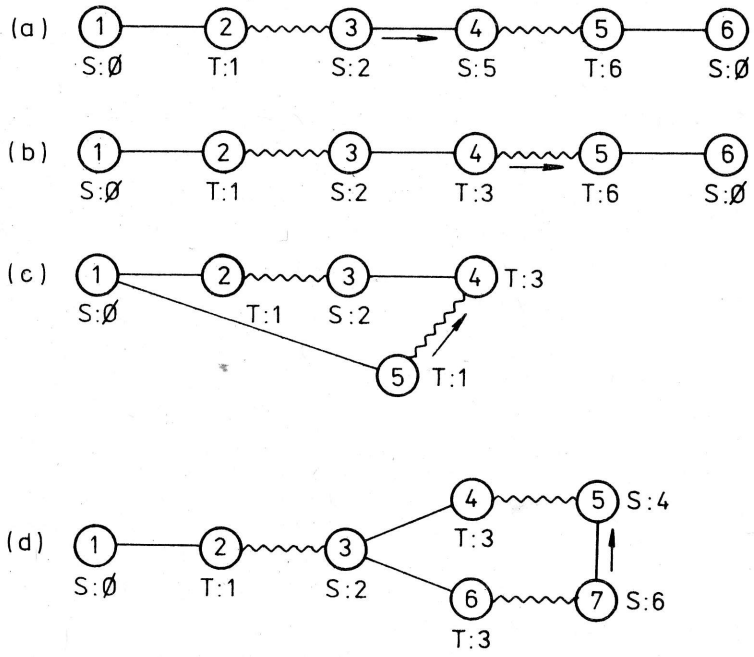
Predpokladajme, že máme nájdené nejaké párenie $X$ v grafe $G$. Ak neexistujú voľné vrcholy, tak $X$ je úplné párenie a niet čo robiť. V opačnom prípade

značkovaním vrcholov nájdeme zväčšujúcu cestu a párenie $X$ cez ňu zväčšíme, alebo zistíme, že zväčšujúca cesta neexistuje, čo podľa Bergeovej vety znamená, že $X$ je už najpočetnejšie párenie. Ak sa uskutočnilo zväčšenie, tak postup opakujeme pre nové párenie.

*Značkovanie* robíme nasledovne: Na začiatku dostanú všetky voľné vrcholy značku „$S$:Ø". Vo všeobecnosti, nech už niektoré vrcholy majú značky. Každý označkovaný vrchol $i$ má značku tvaru „$S: k$", resp. „$T: k$", čo značí, že existuje alternujúca $u$—$i$ cesta začínajúca v nejakom voľnom vrchole $u$ a jej predposledný vrchol je $k$. (Stručne hovoríme, že $i$ má značku $S$, resp. $T$, lebo tieto prvé zložky sú podstatné.) Vrchol $u$ sa nazýva koreň pre $i$. Označkovaný, ale nepreskúmaný vrchol $i$ *preskúmame* takto:

(1) Ak $i$ má značku $S$, tak každý neoznačkovaný vrchol $j$, taký, že $ij \in E(G) - X$, dostáva značku „$T: i$",

(2) ak $i$ má značku $T$, $ij \in X$ a vrchol $j$ je neoznačkovaný, tak $j$ dostáva značku „$S: i$".




Obr. 1. Možnosti vyskytujúce sa pri značkovaní.



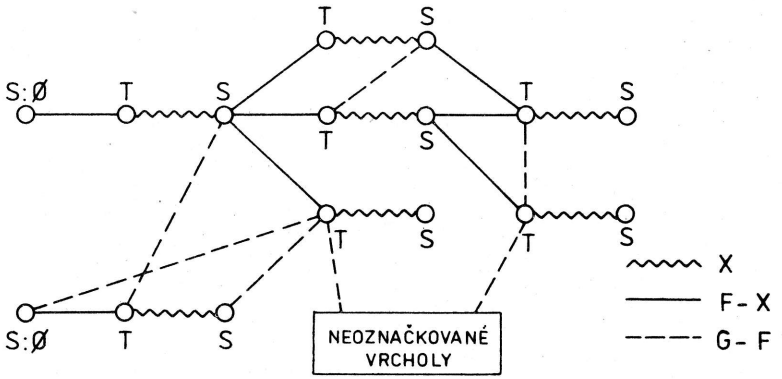


Isté problémy vznikajú, ak vrchol $j$ už značku má. Všetky takéto možnosti ukazuje obr. 1, kde hrany párenia sú vlnovkové a šípka vyznačuje „kolíznu" hranu $ij$. V týchto prípadoch $i$ aj $j$ majú rovnaké značky ($S$ alebo $T$), a vtedy vyhľadáme ich *korene* (t. j. voľné vrcholy $u_i$ a $u_j$, z ktorých vychádza značkovanie pre $i$, resp. $j$), pomocou druhých zložiek vrcholov postupujúc spätne od $i$ a $j$. Tým nájdeme alternujúcu $u_i$—$i$ cestu $P_i$ a alternujúcu $u_j$—$j$ cestu $P_j$. Ak $u_i \neq u_j$, tak zrejme $P_i i j P_j^{-1}$ je zväčšujúca $u_i$—$u_j$ cesta (na obr. 1(a) je $i=3$, $j=4$, $u_i=1$, $u_j=6$).

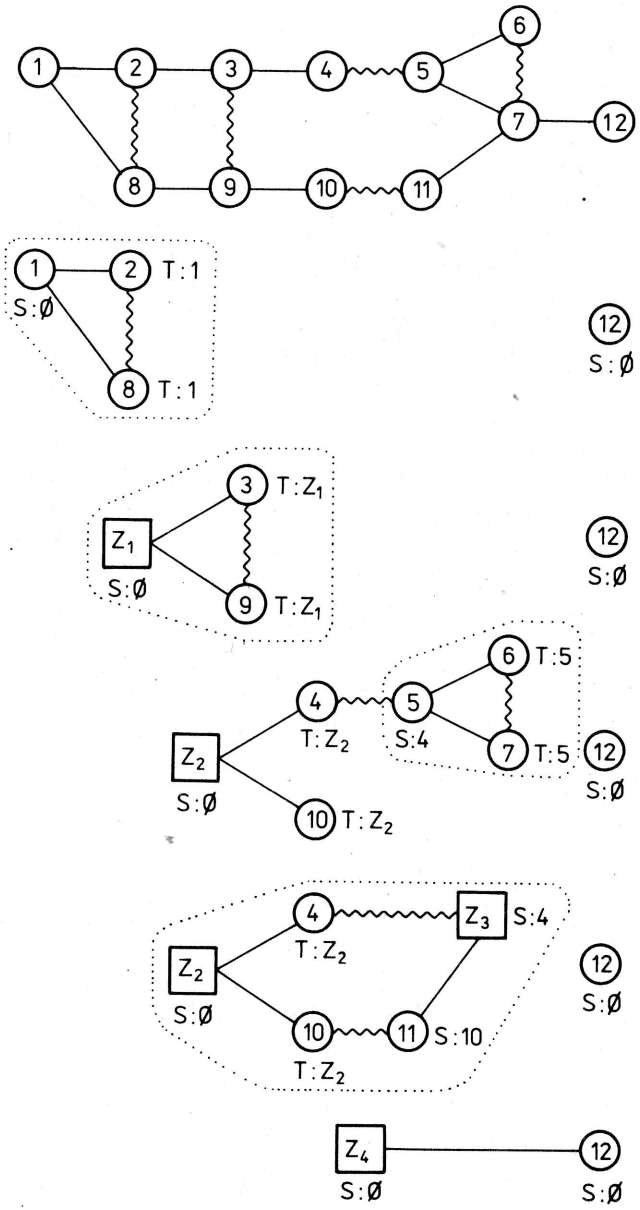
Ak $u_i = u_j =: u$ (ako na obr. 1(c) a (d)), tak $P_i i j P_j^{-1}$ vytvára istý podgraf, ktorý sa skladá z nejakej alternujúcej $u$—$v$ cesty $P$ a alternujúceho $v$—$v$ cyklu $Z$ nepárnej dĺžky (na obr. 1(d) máme $u=1$, $v=3$), pričom obe hrany cyklu incidujúce s $v$ sú nepáriace. Cyklus $Z$ sa nazýva *kvet*, cesta $P$ *stopka* (kvetu $Z$) a $v$ je *báza* (kvetu $Z$). Poznamenajme, že všetky alternujúce cesty pred skúmaním hrany $ij$ tvorili les, a preto $P$ a $Z$ sú určené jednoznačne. Hoci $P$ môže mať nulovú dĺžku (obr. 1(c)), $Z$ má vždy dĺžku aspoň 3.

Kvet $Z$ stiahneme do jediného vrcholu, ktorý budeme nazývať *pseudovrchol* (a na obrázkoch vyznačovať ako štvorček). Dáme mu značku, ktorú mala báza kvetu $Z$, pričom vzniknutý pseudovrchol považujeme za nepreskúmaný (totiž vrcholy $i$, resp. $j$ boli také.). Potom pokračujeme v značkovaní v získanom grafe $G'$ pri korešpondujúcom párení $X'$. Po čase opäť môžeme objaviť kvet v $G'$. Po jeho stiahnutí pokračujeme v značkovaní vo vzniknutom grafe $G''$ pri párení $X''$ atď. (Hovoríme o grafoch, lebo násobné hrany, resp. slučky nemajú význam, a preto ich netreba uvažovať.)

Keďže každá kontrakcia zmenšuje počet vrcholov aspoň o 2, tak v nejakom $G^{(k)}$ buď (i) objavíme zväčšujúcu cestu, alebo (ii) značkovanie skončí preskúmaním




Obr. 2. Maďarský les $F$ v grafe $G$ pri párení $X$.







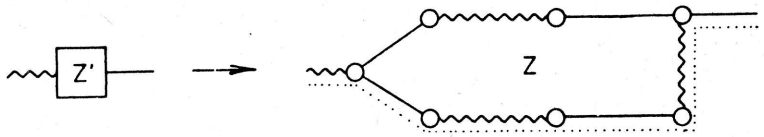

všetkých označkovaných vrcholov a bez objavenia zväčšujúcej cesty. V prípade (i) možno párenie $X^{(k)}$ a (ako ukážeme) aj $X$ zväčšiť. V prípade (ii) zjednotenie všetkých alternujúcich ciest, cez ktoré sme označkovali jednotlivé vrcholy, tvorí tzv. *maďarský les*. (V takomto lese pre každú hranu grafu platí: ak jeden jej kraj má značku $S$, tak druhý patrí tiež do lesa a má značku $T$; pozri obr.2.) Z postupu značkovania je zrejmé, že v prípade (ii) v $G^{(k)}$ žiadna zväčšujúca cesta neexistuje, a potom (ako ukážeme) ani v $G$, takže $X$ je najpočetnejšie párenie.

Na obr. 3 je príklad grafu $G$, párenia $X$ a postupných kontrakcií objavovaných kvetov. Na záver sa získala zväčšujúca cesta $(Z_4, 12)$. Rozvinutím pseudovrcholov postupne z takejto cesty získavame cesty: $(Z_2, 4, Z_3, 12)$, $(Z_1, 9, 3, 4, 5, 6, 7, 12)$ a (1, 2, 8, 9, 3, 4, 5, 6, 7, 12), čo je už zväčšujúca cesta v pôvodnom grafe.

Použitie kontrakcií v procese hľadania zväčšujúcich ciest oprávňuje nasledujúca veta.

**Veta 2.** Nech $Z$ je kvet objavený v grafe $G$ pri značkovaní vzhľadom na párenie $X$. Ďalej nech po kontrakcii kvetu $Z$ na vrchol $Z'$ vznikne z $G$ graf $G'$ a z $X$ párenie $X'$. Potom v $G'$ existuje zväčšujúca cesta pre $X' \Leftrightarrow$ v $G$ existuje zväčšujúca cesta pre $X$.

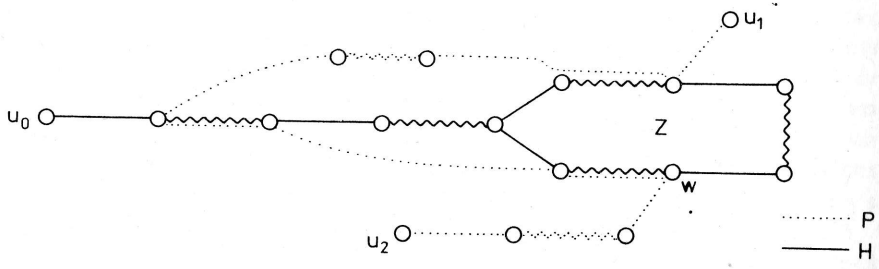
DÔKAZ. ($\Rightarrow$) Ku každej zväčšujúcej ceste v $G'$ obsahujúcej vrchol $Z'$ vieme ľahko nájsť zväčšujúcu cestu $P$ v $G$ tak, ako ilustruje obr. 4.




Obr. 4. K dôkazu prvej implikácie z vety 2.

($\Leftarrow$) Nech $P$ je zväčšujúca $u_1$—$u_2$ cesta v $G$ a nech kvet $Z$ má koreň $u_0$. Ďalej, nech $H$ je graf pozostávajúci z kvetu $Z$ a jeho stopky. Ak $P$ a $H$ sú disjunktné, tak zrejme $P$ existuje aj v $G'$; preto nech $P \cap H \neq \emptyset$. Budeme predpokladať, že $u_2 \neq u_0$. Nech $w$ je posledný vrchol cesty $P$ ležiaci v $H$. Potom ľahko nájdeme alternujúcu $u_0$—$w$—$u_2$ cestu $Q$ (ako na obr. 5) takú, že úsek $u_0$—$w$ leží v $H$ a úsek $w$—$u_2$ je mimo $H$. Pri kontrakcii kvetu $Z$ sa zrejme $Q$ zmení na zväčšujúcu cestu $Q'$ v $G'$. ■

◀ Obr. 3. Priebeh značkovania a kontrakcie kvetov.




Obr. 5. K dôkazu druhej implikácie z vety 2.

Tým máme algoritmus pre najpočetnejšie párenie zdôvodnený a môžeme ho zhrnúť takto:

**Algoritmus**

KROK 0: Nájdeme nejaké (napr. prázdne alebo maximálne) párenie $X$ v grafe $G$. Žiadny vrchol nie je označkovaný.

KROK 1 (*značkovanie*):

1(a): Každému voľnému vrcholu priradíme značku „$S:\emptyset$". Všetky vrcholy sú nepreskúmané.

1(b): Ak sú všetky označkované vrcholy preskúmané, ideme na krok 3. Inak zvolíme ľubovoľný nepreskúmaný vrchol $i$. Ak $i$ má značku $S$, ideme na 1(c): ak má $T$, ideme na 1(d).

1(c) (*prieskum vrcholu i so značkou S*):
Pre každú nepáriacu hranu $ij$ urobíme:
Ak vrchol $j$ má značku $S$, tak podľa druhých zložiek značiek nájdeme korene pre $i$ a $j$. Ak majú rôzne korene, tak ideme na krok 2. Ak majú spoločný koreň, tak sme objavili kvet, ktorý stiahneme na jediný vrchol so značkou bázy kvetu: považujeme ho za nepreskúmaný a ideme na 1(b).
Ak vrchol $j$ má značku $T$, tak ju ponecháme a ak nemá žiadnu značku, tak mu dáme „$T:i$".
Tým je vrchol $i$ preskúmaný a ideme na 1(b).

1 (d) (*prieskum vrcholu i so značkou T*):
Nájdeme (jedinú) páriacu hranu $ij$.
Ak $j$ má značku $T$, tak podľa druhých zložiek značiek nájdeme korene pre $i$ a $j$. Ak majú rôzne korene, ideme na krok 2. Ak majú spoločný koreň, tak sme objavili kvet, ktorý stiahneme na





jediný vrchol so značkou bázy kvetu: považujeme ho za nepreskúmaný a ideme na 1(b).
Ak vrchol $j$ nemá žiadnu značku, tak mu dáme „$S:i$" a ideme na 1(b).

KROK 2 (*zväčšovanie párenia*):
Pomocou nájdenej zväčšujúcej cesty v stiahnutom grafe $G'$ nájdeme zväčšujúcu cestu v pôvodnom grafe $G$ a párenie $X$ zväčšíme. Všetky značky zotrieme a ideme na krok 1.

KROK 3: Žiadna zväčšujúca cesta neexistuje, a teda párenie je najpočetnejšie: STOP. ●

Odhadnime zložitosť uvedeného algoritmu. Počet zväčšovaní je nanajvýš $O(n)$. Každé zväčšovanie vyžaduje značkovanie, pri ktorom prezrieme nanajvýš $O(n)$ vrcholov (lebo pseudovrcholov nie je viac ako $O(n)$) a $O(n)$ operácií stačí na nájdenie koreňov a vlastnú zmenu párenia. Na prezretie jedného vrcholu stačí $O(n)$ operácií, ak nepočítame kontrakciu a rozvíjanie, na ktoré však stačí $O(n^2)$ operácií. Teda celkove máme $O(n^4)$ algoritmus.

Ak $G$ je bipartitný graf, tak tam kvety neexistujú, a teda stačí $O(n^3)$ operácií. Avšak aj vo všeobecnosti možno dosiahnuť $O(n^3)$ algoritmus, ak sa vyhneme kontrakcii kvetu vhodným označením jeho vrcholov [Lawler (1976)]. Inú implementáciu tejto zložitosti dáva aj Gabow (1976). Hopcroft a Karp (1973) dávajú $O(n^{5/2})$ algoritmus pre bipartitné grafy (presnejšie: $O((m+n)n^{1/2})$ algoritmus) a rozvinutím ich myšlienok, Even a Kariv (1975) dosiahli takúto nízku zložitosť aj vo všeobecnom prípade.

## 8.C
## Hľadanie najcennejšieho párenia

Edmonds (1965b) navrhol dobrý algoritmus aj pre problém najcennejšieho párenia. Zovšeobecnil myšlienku Kuhna (1955), ktorý použil lineárne programovanie (a vypracoval tzv. *maďarskú metódu* pre *priraďovací problém*, t. j. najcennejšie párenie v bipartitnom grafe).

Poznamenajme, že ak z párenia vynecháme hrany s cenou $c_{ij} \leq 0$, tak sa jeho celková cena nezmenší, a preto v ďalšom môžeme predpokladať, že v grafe $G$ sú všetky ceny $c_{ij} > 0$.

Úlohe najprv priradíme istú úlohu lineárneho programovania. Nech pre každú hranu $ij$ je $x_{ij}$ premenná, pričom chceme, aby $x_{ij} = 1$, ak hrana $ij$ je vybraná do

párenia a $x_{ij} = 0$ inak. Ak $c_{ij}$ je cena hrany $ij$ a $B$ je vrcholovo-hranová incidenčná matica uvažovaného grafu $G$, tak úlohu môžeme formulovať takto:

maximalizovať $c^T x$

pri ohraničeniach $Bx \leq 1$,
$x_{ij} \in \{0, 1\}$.

Posledné ohraničenie však robí úlohu diskrétnou a nevieme ju dobre riešiť. Edmonds preto upustil od diskrétnosti, ale pribral ďalšie ohraničenia, ktoré musí každé párenie spĺňať. Nech $Q_k \subseteq V(G)$ má nepárny počet prvkov (stručne: $Q_k$ je nepárna podmnožina), teda $|Q_k| = 2q_k + 1$, potom z ľubovoľného párenia nanajvýš $q_k$ hrán patrí do $G(Q_k)$. Nech $Q_1, Q_2, \ldots, Q_p$ sú všetky nepárne podmnožiny množiny $V(G)$ (zrejme $p = 2^{n-1}$) a nech $Q = (q_{ij})$ je matica, ktorej riadky zodpovedajú týmto nepárnym podmnožinám a stĺpce hranám $e_1, e_2, \ldots, e_m$ grafu $G$, pričom $q_{ks} = 1$ práve vtedy, keď $e_s \in G(Q_k)$. Potom spomínanú podmienku môžeme formálne zapísať takto: $Qx \leq q$, kde $q = (q_1, q_2, \ldots, q_p)^T$. Takto získavame úlohu:

maximalizovať $c^T x$

pri ohraničeniach $Bx \leq 1$,
$Qx \leq q$,
$x \geq 0$.

Pravda, na prvý pohľad sme si pôvodnú formuláciu pokazili, lebo sme nezaručili, že riešenie $x$ tejto úlohy bude mať celočíselné zložky (0 a 1). Avšak aj keby sme to zaručili, alebo nežiadali, úlohu riešiť priamo by sme sotva mohli, keďže už jej zápis je rozsiahly (matica $Q$ má $m2^{n-1}$ prvkov). Nasledujúca metóda takýto explicitný zápis nevyžaduje a zaručuje celočíselné riešenie.

*Duálna úloha* k uvedenej *primárnej úlohe* lineárneho programovania je:

minimalizovať $\sum_{i=1}^{n} u_i + \sum_{k=1}^{p} q_k z_k$

pri ohraničeniach $B^T u + Q^T z \geq c$,
$u \geq 0$,
$z \geq 0$,

kde $u$ a $z$ sú stĺpcové vektory duálnych premenných ($u_i$ zodpovedá vrcholu $i$ a $z_k$ nepárnej podmnožine $Q_k$).

Z teórie duality v lineárnom programovaní sú známe tzv. *podmienky ortogonality*, ktoré sú nutnými a postačujúcimi podmienkami na to, aby prípustné riešenia primárnej a duálnej úlohy boli optimálne. Pre našu dvojicu úloh tieto podmienky sú takéto:





(1) $x_{ij} > 0 \Rightarrow u_i + u_j + \sum_{\substack{k=1 \\ ij \in G(Q_k)}}^{p} z_k = c_{ij}$, $(ij \in E(G))$:

(2) $u_i > 0 \Rightarrow \sum_{j \in V(i)} x_{ij} = 1$, $(i \in V(G))$:

(3) $z_k > 0 \Rightarrow Q_k x = q_k$, $(k = 1, 2, \ldots, p)$,

kde $Q_k$ je $k$-tý riadok matice $Q$ (a zodpovedá podmnožine označenej tým istým symbolom $Q_k$). (Poznamenajme, že v ďalšom používame len postačiteľnosť týchto podmienok a táto ľahko vyplýva z tzv. *slabej vety o dualite*: pozri časť 2.B.)

Myšlienka Edmondsovej metódy pre hľadanie najcennejšieho párenia je takáto: Nájdeme také prípustné riešenia primárnej a duálnej úlohy, ktoré spĺňajú podmienky (1) a (3), ale nemusia spĺňať (2). Potom postupne prerábame tieto riešenia na také, ktoré spĺňajú viac a viac podmienok (2), až dospejeme k takým prípustným riešeniam, ktoré budú spĺňať všetky podmienky (1), (2) a (3), a teda už to budú optimálne riešenia.

Na začiatku zrejme stačí voliť:

$x_{ij} = 0$ pre všetky $ij \in E(G)$,

$u_i = \frac{1}{2} \max_{ij \in E(G)} \{c_{ij}\}$ pre všetky $i \in V(G)$,

$z_k = 0$ pre všetky $k$.

Teda v $G$ zvolíme párenie $X = \emptyset$ a vieme, že každé párenie spĺňa primárne ohraničenia. Keďže $u_i$ sú dostatočne veľké, aj duálne ohraničenia sú splnené. Triviálne sú splnené (1) a (3).

Vo všeobecnosti, ak máme párenie $X$ (t. j. prípustné riešenie primárnej úlohy) a prípustné riešenie duálnej úlohy spĺňajúce (1) a (3), tak alebo je splnené aj (2), alebo existuje vrchol $i$, pre ktorý (2) neplatí. V prvom prípade máme optimum. V druhom prípade existuje voľný vrchol a pokúsime sa „zlepšiť" riešenia. Vo všeobecnosti zlepšovanie riešení sa robí pomocou alternujúceho lesa, ktorý získame značkovaním (ako pri hľadaní najpočetnejšieho párenia) v tzv. *grafe rovnosti* $G'$, ktorý je získaný z $G$ kontrakciami všetkých množín $Q_k$, pre ktoré $z_k > 0$ a vynechaním všetkých hrán $ij$, kde $u_i + u_j + \Sigma z_k > c_{ij}$.

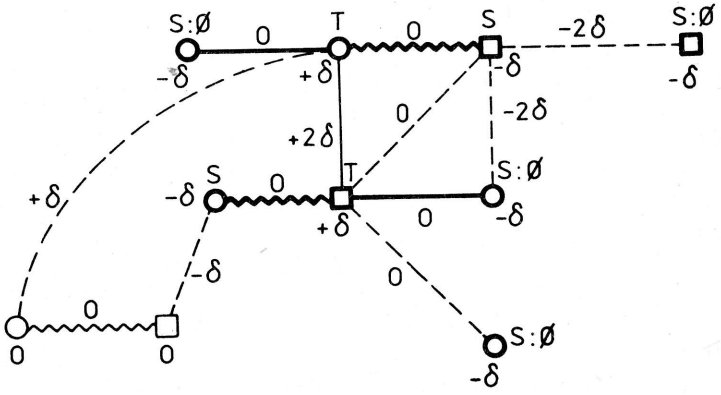
Pri značkovaní sa nájdené kvety stiahnu do pseudovrcholov, ktoré sa môžu dostať ešte do väčších pseudovrcholov atď. Tiež sa budú niektoré pseudovrcholy rozťahovať na kvety. Teda graf $G'$, na začiatku utvorený z $G$, sa bude týmto meniť, ale vždy bude označený symbolom $G'$. Taký pseudovrchol, ktorý neleží v inom pseudovrchole, sa nazýva *vonkajší*. Poznamenajme, že v skutočnosti nemusíme

kvety stiahnuť, ale stačí ich dať do „obálok", ktoré pri značkovaní považujeme za pseudovrcholy. Vtedy bude $G'$ multigraf, avšak pre potreby značkovania stačí uvažovať graf. Z výpočtového hľadiska je dôležité, že len také nepárne množiny vrcholov $Q_k$ budú mať $z_k > 0$, ktoré predstavujú pseudovrcholy (kvety). Preto stačí registrovať len málo hodnôt $z_k$ (ostatné sú nuly).

Majme teda $G'$ a korešpondujúce párenie. Začíname značkovanie voľných vrcholov. Ak objavíme kvet $k$, tak ho stiahneme na pseudovrchol, ktorý dostane značku $S$, položíme $z_k = 0$ a pokračujeme v značkovaní vo vzniknutom multigrafe (ktorý tiež označíme $G'$). Značkovanie sa zrejme ukončí alebo nájdením zväčšujúcej cesty alebo nájdením maďarského lesa. Rozoberieme obe možnosti.

V prvom prípade pomocou nájdenej zväčšujúcej cesty zväčšíme okamžité párenie v $G'$ o jednu hranu, všetky značky zotrieme a v $G'$ začneme znova značkovanie (podľa nového párenia).

Všimnime si, ako sa týmto zmenila situácia pre graf $G$. Zväčšujúcej ceste v $G'$ zrejme zodpovedá nejaká zväčšujúca $s$—$t$ cesta v $G$ a novému páreniu v $G'$ nové párenie v $G$, ktoré však teraz nekonštruujeme (ale až na konci celého algoritmu). Z ďalšieho vyplynie, že minimálna hodnota premennej $u_i$ je nenulová a že sa nadobúda aj na všetkých voľných vrcholoch grafu $G$. Preto $u_s = u_t > 0$ a keďže $u_s$ a $u_t$ boli voľné, tak podmienku (2) nespĺňali. Avšak pri novom párení už ju budú spĺňať. Takto sa počet vrcholov nespĺňajúcich (2) zmenší o 2. Počet páriacich hrán v pseudovrchole sa zmenou párenia nemení (zostáva ich tam $(|Q_k| - 1)/2$), a preto sa podmienka (3) neporuší. Splnená zostane aj (1), lebo zväčšujúca cesta v $G$ pozostáva len z hrán, kde $u_i + u_j + \Sigma z_k = c_{ij}$ (na hranách v $G'$ je to zrejmé a hrany vo vnútri pseudovrcholov patrili do nejakého predchádzajúceho $G'$; vidíme, že




Obr. 1. Účinok zmien duálnych premenných.





splnenie rovnosti si zachovávajú). Duálne premenné $u_i$, $z_k$ sa pri zmene párenia nemenia.

Teraz rozoberieme druhú možnosť, t. j. situáciu, keď už v značkovaní nemožno pokračovať, lebo vytvorený les $F$ je maďarský. V tomto prípade pozmeníme duálne premenné, čo bude istý pokrok. Určíme číslo $\delta > 0$ (z nižšie uvedených podmienok) a premenné $u_i$ a $z_k$ zmeníme nasledovne: Pre každý vrchol $i \in V(G)$ so značkou $S$ alebo obsiahnutý vo vonkajšom pseudovrchole so značkou $S$, zmeníme $u_i$ na $u_i - \delta$. Pre každý vrchol $i \in V(G)$ so značkou $T$, resp. obsiahnutý vo vonkajšom pseudovrchole so značkou $T$, zmeníme $u_i$ na $u_i + \delta$. Pre každý vonkajší pseudovrchol $k$ (presnejšie: pre množinu $Q_k$ vrcholov v ňom) zmeníme $z_k$ na $z_k + 2\delta$, ak má značku $S$ a na $z_k - 2\delta$, ak má značku $T$. Hodnoty $u_i$ a $z_k$ sa pre neoznačkované vrcholy a pseudovrcholy a vnútorné pseudovrcholy nemenia. Obr. 1 ukazuje zmeny $u_i$ a výrazu $u_i + u_j + \Sigma z_k$ na hranách $ij$. (Páriace hrany sú vlnovkové, pseudovrcholy sú štvorcové, prvky lesa $F$ sú vyznačené hrubo a čiarkované sú hrany z $G$ nepatriace do $G'$ ani do pseudovrcholov.) Na hranách $ij \in E(G)$, ktorých oba kraje ležia v nejakom pseudovrchole, sa hodnota výrazu $u_i + u_j + \Sigma z_k$ nezmení, lebo tam sa zmeny kompenzujú.

Z požiadavky duálnej prípustnosti máme nasledujúce obmedzenia na $\delta$.

(4) $u_i - \delta \geq 0$, ak $i$ má značku $S$, alebo leží vo vonkajšom pseudovrchole so značkou $S$.

(5) $z_k - 2\delta \geq 0$, ak $k$ je vonkajší pseudovrchol so značkou $T$.

Pokiaľ ide o výraz $u_i + u_j + \Sigma z_k$, tak duálnu prípustnosť môžu ovplyvniť len zmeny, pri ktorých sa výraz zmenšuje. Keďže na hranách neležiacich v pseudovrcholoch je vždy $\Sigma z_k = 0$, tak získame ďalšie obmedzenia na $\delta$:

(6) $(u_i - \delta) + (u_j - \delta) \geq c_{ij}$, ak každý z krajov hrany $ij$ alebo má značku $S$, alebo leží vo vonkajšom pseudovrchole so značkou $S$, ale oba vrcholy neležia v tom istom pseudovrchole.

(7) $(u_i - \delta) + u_j \geq c_{ij}$, ak je $ij$ taká hrana, že $i$ alebo obsahujúci ho vonkajší pseudovrchol má značku $S$ a ani $j$ ani obsahujúci ho vonkajší pseudovrchol nemajú žiadnu značku.

Tieto štyri obmedzenia spolu s $\delta \geq 0$ zaručujú duálnu prípustnosť. Keďže pre žiadnu páriacu hranu $ij$ sa výraz $u_i + u_j + \Sigma z_k$ nemení, tak (1) zostáva dodržaná. Hodnota $z_k$ sa mohla zväčšiť iba pre množinu $Q_k$ v pseudovrchole. Ten však vždy obsahuje $q_k = (|Q_k| - 1)/2$ páriacich hrán, a teda ani (3) sa neporušila. Hodnota $u_i$ sa zväčšuje len pri vrcholoch so značkou $T$, resp. ležiacich v pseudovrcholoch so značkou $T$. Pretože skúmame prípad, kedy neexistuje zväčšujúca cesta, tak pri každom takom vrchole, resp. pseudovrchole musí byť páriaca hrana. Teda každý

vrchol $i$ spĺňajúci (2) bude spĺňať (2) aj po zmene duálnych premenných. Aby sme dosiahli „kvalitatívne zlepšenie", tak volíme $\delta$ maximálne možné spĺňajúce (4) až (7).

Preskúmame jednotlivé prípady určujúce takéto $\delta$ a ukážeme, že vždy je $\delta > 0$. Predovšetkým si všimnime, že na začiatku boli všetky $u_i$ zvolené rovnaké a pri každom zmenšení $u_i$ majú všetky voľné vrcholy, resp. pseudovrcholy, značku $S$. Preto minimálna hodnota $u_i$ je (aj) vo všetkých voľných vrcholoch a je nenulová (inak sú všetky podmienky (2) splnené). Teda ak je $\delta$ určené podľa (4), tak $\delta > 0$ a po zmene budú všetky $u_i$ vo voľných vrcholoch nulové, a tým (2) splnená, takže máme nájdené optimálne riešenia oboch úloh.

Ak je $\delta$ určené podľa (5), tak nejaké $z_k$ sa stane nulové a príslušný pseudovrchol so značkou $T$ môžeme roztiahnuť. Ak sa dohodneme, že po každej zmene duálnych premenných každý vonkajší pseudovrchol $k$ so $z_k = 0$ a značkou $T$ roztiahneme a tiež po každom zväčšovaní párenia roztiahneme všetky pseudovrcholy so $z_k = 0$, tak aj v (5) vždy získame $\delta > 0$. Totiž pri poslednom značkovaní môžu vzniknúť len pseudovrcholy so značkou $S$ (a $z_k = 0$), ale (5) sa týka pseudovrcholov so značkou $T$.

Ak je $\delta$ určené podľa (6), tak po zmene bude hrana $ij$ figurovať v $G'$, a preto bude možné existujúci maďarský les $F$ rozšíriť o túto hranu (získame alebo zväčšujúcu cestu alebo nový kvet). Teraz tam však hrana $ij$ nepatrí, lebo les $F$ je maximálny, a preto je $u_i + u_j > c_{ij}$. Teda aj tu získame $\delta > 0$.

Konečne, ak je $\delta$ určené podľa (7), tak k lesu $F$ opäť bude možné pridať aspoň hranu $ij$. Podobne ako v predchádzajúcom prípade, aj tu získame $\delta > 0$.

Tým sme kompletne preskúmali aj prípad, keď sme značkovaním získali maďarský les. Ako sme videli, sú takéto možnosti: (i) našli sme optimum (najcennejšie párenie), (ii) po zmene premenných $u_i$ a $z_k$ alebo nájdeme zväčšujúcu cestu, alebo roztiahneme pseudovrchol so značkou $T$, alebo utvoríme pseudovrchol so značkou $S$, alebo k lesu $F$ pridáme ďalšiu hranu, a tým novú značku $T$.

V celom algoritme sa optimum môže nájsť iba raz a zväčšujúca cesta nanajvýš $\lfloor n/2 \rfloor$ razy. Preto si všimnime len operácie medzi dvoma po sebe idúcimi zväčšovaniami párenia (resp. pred prvým a po poslednom). Pretože vtedy sa značky $S$ a $T$ nevymieňajú a vytvárame len vrcholy so značkou $S$, roztiahnutie pseudovrcholu so značkou $T$ sa môže uskutočniť nanajvýš $\lfloor n/2 \rfloor$-krát. Pseudovrcholy so značkou $S$ nerozťahujeme, a preto utvorenie pseudovrcholu so značkou $S$ sa môže uskutočniť nanajvýš $\lfloor n/2 \rfloor$-krát. Konečne, pridať k alternujúcemu lesu $F$ novú hranu, a teda novú značku $T$, možno nanajvýš $(n-1)$-krát. Teda celkove medzi dvoma zväčšovaniami párenia robíme zmenu duálnych premenných nanajvýš $O(n)$-krát. Každú zmenu duálnych premenných možno zrejme urobiť v čase $O(n^2)$. Z operácií medzi dvoma zväčšovaniami dominuje stiahnutie kvetov





a roztiahnutie pseudovrcholov — $O(n^3)$ operácií (jedno stiahnutie, resp. roztiahnutie možno urobiť v čase $O(n^2)$). Takto na celý algoritmus vystačíme s $O(n^4)$ operáciami. Avšak starostlivou implementáciou možno získať $O(n^3)$ algoritmus. Implementácie s časmi $O(n^4)$ a $O(n^3)$ uvádza napr. Lawler (1976).

Uvedený zdôvodnený postup môžeme zhrnúť takto:

**Algoritmus**

KROK 0: Začíname s párením $X := \emptyset$ a duálnymi premennými $u_i := \frac{1}{2} \max c_{rs}$ a $z_k := 0$. Utvoríme graf rovnosti $G'$.

KROK 1 (*značkovanie*):

1(a): Každému voľnému vrcholu alebo pseudovrcholu priradíme značku „$S$:$\emptyset$". Tieto vrcholy tvoria les $F$.

1(b): Les $F$ rozširujeme značkovaním v $G'$. Ak nájdeme kvet $k$, tak ho stiahneme na pseudovrchol so značkou jeho bázy (má značku $S$), položíme $z_k = 0$ a so vzniknutým $G'$ a príslušným lesom $F$ ideme na 1(b). Ak nájdeme zväčšujúcu cestu, ideme na krok 2. Ak utvoríme maďarský les, ideme na krok 3.

KROK 2 (*zväčšovanie párenia*): Pomocou nájdenej zväčšujúcej cesty zväčšíme párenie v $G'$. Zotrieme všetky značky, roztiahneme všetky pseudovrcholy so $z_k = 0$ a s takýmto $G'$ ideme na krok 1.

KROK 3 (*zmena duálnych premenných*): Z podmienok (4) až (7) určíme maximálne $\delta > 0$ a urobíme zmeny duálnych premenných. Ak $\delta$ dáva v (4) rovnosť, tak ideme na krok 4. Ak $\delta$ dáva rovnosť v (5), tak roztiahneme tie vonkajšie pseudovrcholy značky $T$, ktoré majú $z_k = 0$ a s takýmto $G'$ a príslušným $F$ ideme na 1(b). Ak $\delta$ dáva rovnosť v (6) alebo (7), tak ku $G'$ pridáme tie hrany $ij$, ktoré dávali v (6) alebo (7) rovnosť a s takým $G'$ a s nezmeneným $F$ ideme na 1(b).

KROK 4 (*záver*): Roztiahnutím všetkých pseudovrcholov v $G'$ (v obrátenom poradí ich vzniku) získame najcennejšie párenie v $G$; STOP. ●

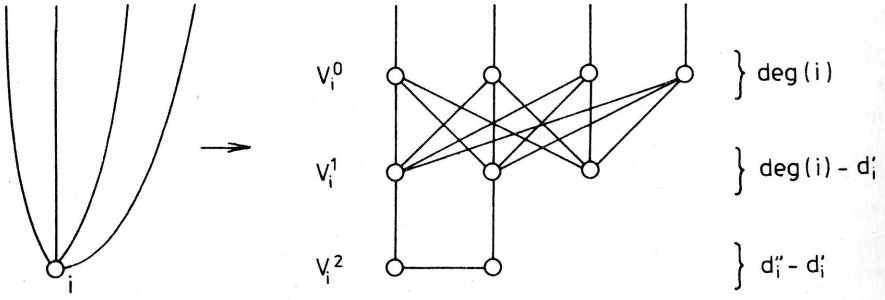
## 8.D
## Faktory s predpísanými stupňami

Nájdením najpočetnejšieho párenia vyriešime aj otázku, či graf má úplné párenie, t. j. lineárny faktor. Od tohto problému sa dostávame k úlohe nájsť kvadratický alebo iný pravidelný faktor, resp. všeobecne k úlohe nájsť faktor, ktorý má v každom vrchole $i$ predpísaný stupeň $d_i$. Tutte (1954) pretransformoval túto úlohu na úlohu nájsť lineárny faktor. Vychádzajúc z tejto myšlienky možno zvládnuť trochu všeobecnejšie úlohy.

### DOLNÉ A HORNÉ MEDZE NA STUPNE

Nech je daný graf (alebo aj pseudograf) $G$ a pre každý jeho vrchol $i$ nech je daná dvojica celých čísel $d'_i$, $d''_i$ $(0 \leq d'_i \leq d''_i \leq \deg(i))$. Úlohou je nájsť taký faktor $F$ grafu $G$, že pre každý vrchol $i$ platí: $d'_i \leq \deg_F(i) \leq d''_i$.

Ukážeme polynomiálnu transformáciu tejto úlohy na úlohu nájsť lineárny faktor. Najprv každý vrchol $i \in V(G)$ nahradíme štruktúrou z obr. 1, kde $|V_i^0|$




Obr. 1. Transformácia stupňových medzí.

$= \deg_G(i)$, $|V_i^1| = \deg_G(i) - d'_i$, $|V_i^2| = d''_i - d'_i$, každý vrchol z $V_i^0$ je spojený s každým vrcholom z $V_i^1$ a medzi $V_i^1$ a $V_i^2$ je práve $d''_i - d'_i$ hrán, ktoré sú nezávislé, ale inak ľubovoľné. Ďalej medzi vrcholmi množiny $V_i^2$ je práve $\lfloor |V_i^2|/2 \rfloor$ hrán, ktoré sú nezávislé, ale inak ľubovoľné. Okrem toho, pre každé $i$, kde $|V_i^2|$ je nepárne, vyznačíme ten (jediný) vrchol, ktorý nie je susedný so žiadnym vrcholom z $V_i^2$. Nech takto vyznačené vrcholy tvoria množinu $\bar{V}^2$. Ak je $|\bar{V}^2|$ nepárne, tak pridáme ešte jeden nový vrchol $w$ a získame množinu $U^2 := \bar{V}^2 \cup \{w\}$. Ak $|\bar{V}^2|$ je párne, tak $U^2 := \bar{V}^2$. Konečne, vrcholy množiny $U^2$ pospájame do kompletného grafu.





Takto sme ku grafu $G$ priradili nový graf, ktorý označíme symbolom $\hat{G}$. Prenechávame čitateľovi overiť (8.Z.6), že platí:

**Veta 1.** Graf $G$ má faktor $F$ taký, že $d'_i \leq \deg_F(i) \leq d''_i$ pre každé $i \Leftrightarrow$ *graf* $\hat{G}$ má lineárny faktor.

DÔKAZ vety 1 poskytne aj polynomiálny algoritmus pre $F$.

Teda ak stupeň každého vrcholu faktora môže prebiehať interval, tak pre úlohu máme polynomiálny algoritmus. Žiaľ, vo všeobecnosti ak je pre každý vrchol predpísaná prípustná množina stupňov, už to nie je tak [Plesník, Wawruch (1982)]:

**Veta 2.** Problém nájsť faktor grafu so stupňami z predpísaných množín je NP-ťažký dokonca v planárnych kubických a bipartitných grafoch.

Istou analógiou úlohy o najlacnejšom maximálnom toku je nasledujúca úloha.

### ÚLOHA S CENAMI A KAPACITAMI HRÁN A S VRCHOLOVÝMI POŽIADAVKAMI

Daný je graf $G$. Pre každý jeho vrchol $i$ je predpísaná celočíselná požiadavka $d_i > 0$ a každá hrana $ij$ má celočíselnú kapacitu $b_{ij} > 0$ a reálnu cenu $c_{ij}$. Úlohou je nájsť celé čísla $x_{ij}$ $(ij \in E(G))$

minimalizujúce

$$\sum_{ij \in E(G)} c_{ij} x_{ij}$$

pri ohraničeniach

(1)
$$\sum_{j \in V(i)} x_{ij} = d_i \quad \text{pre} \quad i \in V(G),$$

(2)
$$0 \leq x_{ij} \leq b_{ij} \quad \text{pre} \quad ij \in E(G).$$

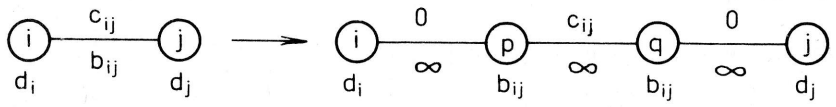
Takéto úlohy skúmali Edmonds (1967a), Edmonds a Johnson (1970) a Edmonds (1975). Tu uvedieme len niekoľko poznatkov.

Všimnime si, že ak všetky $b_{ij} = 1$, tak hrany $ij$ s $x_{ij} = 1$ určujú faktor grafu, ktorý má vo vrchole $i$ stupeň $d_i$.

Hoci uvedená úloha sa zdá byť dosť všeobecná, predsa ju možno transformovať na úlohu najcennejšieho párenia, čo teraz urobíme v troch krokoch (i) až (iii).

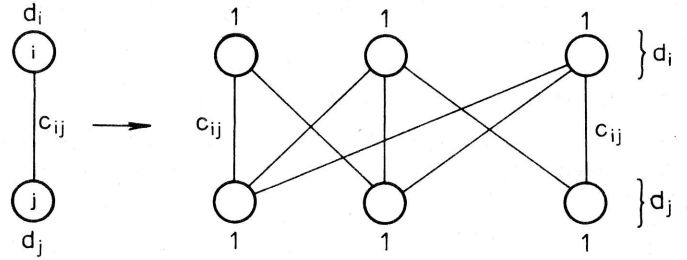
(i) Najprv vylúčime kapacity hrán. Každú hranu $ij$ nahradíme cestou $ipqj$, kde $p$ a $q$ sú nové vrcholy priradené hrane $ij$. Požiadavky vrcholov $i$ a $j$ sa nemenia,

$\hat{d}_p := b_{ij} =: \hat{d}_q$ a (ceny) $\hat{c}_{ip} := \hat{c}_{qj} := 0$, $\hat{c}_{pq} := c_{ij}$ (pozri obr. 2). Transformácia je zrejme správna: Vo vzniknutom grafe $\hat{G}$ bude $\hat{x}_{ip} = \hat{x}_{qj} = x_{ij}$ a $\hat{x}_{pq} = b_{ij} - x_{ij}$, ak $x_{ij}$ je v $G$ a obrátene. Takto získame úlohu maximalizovať $\Sigma \hat{c}_{pq}\hat{x}_{pq} = \Sigma c_{ij}b_{ij} - \Sigma c_{ij}x_{ij}$.




Obr. 2. Vylúčenie kapacít hrán.

(ii) Teraz transformujeme túto maximalizačnú úlohu bez hranových kapacít (všetky sú ∞) na úlohu nájsť najcennejší lineárny faktor. K tomu každý vrchol $i$ s požiadavkou $d_i$ nahradíme $d_i$ vrcholmi s jednotkovými požiadavkami a každú hranu $ij$ nahradíme $d_id_j$ hranami s cenami $c_{ij}$ (obr. 3).




Obr. 3. Transformácia na jednotkové požiadavky.

(iii) Konečne, úlohu nájsť najcennejší lineárny faktor transformujeme na úlohu nájsť najcennejšie párenie. Na tento účel každú cenu $c_{ij}$ zväčšíme o rovnaké dostatočne veľké číslo $M$. Tieto nové ceny $c_{ij} := c_{ij} + M$ zabezpečia, že najcennejšie párenie bude úplné (ak úplné existuje).

Poznamenajme, že kroky (i) a (iii) predstavujú polynomiálne transformácie, ale krok (ii) nie, a preto je vhodný iba pre malé hodnoty $d_i$ (a $b_{ij}$, ak uvažujeme pôvodný problém). Toto je splnené napr. pri hľadaní najkratšej cesty v grafe (pozri 4.D).

Rozvinutím myšlienok použitých pri hľadaní najcennejšieho párenia (8.C) sa však podarilo zostrojiť polynomiálny algoritmus aj pre úlohu s cenami a kapacitami hrán a ľubovoľnými vrcholovými požiadavkami a tiež pre niektoré jej modifikácie (pozri [Edmonds, Johnson (1970)] a [Edmonds (1975)].





## 8.Z
## Cvičenia a poznámky

1. Ukážte, že v nenulovom grafe pomer mohutností najpočetnejšieho a maximálneho párenia nepresahuje 2; uveďte extremálne príklady. [Návod: Dôkaz vety 8.A.1.]
2. Dokážte Königovu–Hallovu vetu 8.A.2 pomocou tokov.
3. (a) Dokážte, že pre ľubovoľné párenie $X$ v grafe $G$ existuje najpočetnejšie párenie $X^*$ pokrývajúce všetky vrcholy pokryté párením $X$. Špeciálne, pre ľubovoľný vrchol $w$ s $\deg(w)>0$ existuje najpočetnejšie párenie pokrývajúce $w$ [Lawler (1976)]. [Návod: Všimnite si účinok zmeny párenia cez alternujúci ťah na pokrytie vrcholov.]
   (b) Nech $G$ je bipartitný graf s partiami $S$ a $T$. Dokážte, že pre ľubovoľné dve párenia $X_1$ a $X_2$ v $G$ existuje také párenie $X \subseteq X_1 \cup X_2$, ktoré pokrýva všetky vrcholy v $S$ pokryté s $X_1$ a všetky vrcholy v $T$ pokryté s $X_2$ [Mendelsohn, Dulmage (1958)].
4. Dokážte, že: (a) Pravidelný bipartitný multigraf stupňa $k$ možno rozložiť na $k$ 1-faktorov [König (1936)].
   (b) Pravidelný pseudograf stupňa $2k$ možno rozložiť na $k$ 2-faktorov [Petersen (1891)]. [Návod: Každý komponent má eulerovský ťah, ktorého hrany súhlasne zorientujeme: získaný pseudodigraf reprezentujeme bipartitným multigrafom a použijeme (a).]
5. Nech $F$ je maďarský les grafu $G$ pri párení $X$ (pozri 8.B). Dokážte, že ak $X_1$ je najpočetnejšie párenie v $G - V(F)$, tak $X_1 \cup (X \cap E(F))$ je najpočetnejšie párenie v $G$. Ako to možno využiť v algoritme pre najpočetnejšie párenie?
6. Dokážte vetu 8.D.1.
7. (a) Uvažujte zovšeobecnenie úlohy s cenami z 8.D, kde namiesto (2) žiadame, aby $b'_{ij} \leq x_{ij} \leq b''_{ij}$ ($b'_{ij}$ a $b''_{ij}$ sú predpísané celé nezáporné čísla). Redukujte toto zovšeobecnenie na základnú úlohu.
   (b) V inom zovšeobecnení namiesto (1) žiadame: $d'_i \leq \sum_{j \in V(i)} x_{ij} \leq d''_i$, kde $d'_i$ a $d''_i$ sú dané celé nezáporné čísla. Redukujte takúto úlohu na základnú. [Návod: Pri každom vrchole $i$ dáme namiesto dvojice $d'_i$, $d''_i$ len požiadavku $d''_i$ a pridáme dva nové vrcholy $i_1$, $i_2$ s požiadavkami $r_i := d''_i - d'_i$. Ďalej pridáme hrany $ii_1$, $ii_2$ a $i_1i_2$ s príslušnými kapacitami: $\lfloor r_i/2 \rfloor$, $\lfloor r_i/2 \rfloor$ a $r_i$. Potom pridáme štyri vrcholy $w_1$, $w_2$, $w_3$ a $w_4$ (spoločné pre všetky $i$), z ktorých prvé tri majú požiadavky $n$ ($n$ je počet pôvodných vrcholov) a $w_4$ má 0 alebo 1. Konečne pridáme hrany $w_1w_2$, $w_1w_3$, $w_2w_3$, $w_3w_4$ kapacít $\lfloor n/2 \rfloor$, $\lceil n/2 \rceil$, $n$, 1 a pre každé pôvodné $i$ hranu $i_1w_1$ kapacity 1. Ceny všetkých nových hrán sú nulové.

Získaný graf označme $G_0$, ak $w_4$ má požiadavku 0 a $G_1$, ak $w_4$ má 1. Ukážte, že stačí riešiť dve základné úlohy (jednu pre $G_0$ a jednu pre $G_1$).]

8. Navrhnite algoritmus pre nájdenie takého párenia, v ktorom minimálna cena hrany je maximálna.
9. Nech $p$ je mohutnosť najpočetnejšieho párenia v grafe $G$ bez nulových vrcholov a nech $q$ je minimálny počet hrán pokrývajúcich všetky vrcholy v $G$ (hrana pokrýva svoje kraje). Dokážte, že $p+q=n(G)$ [Gallai (1959)] a navrhnite algoritmus pre nájdenie minimového hranového pokrytia vrcholov grafu [Norman, Rabin (1959)].
   Poznamenajme, že v grafe s cenami hrán má zmysel hľadať tzv. *najlacnejšie hranové pokrytie vrcholov* grafu (minimalizujeme súčet cien pokrývajúcich hrán). Algoritmus pre nájdenie takéhoto pokrytia vypracoval L. J. White (dizertácia 1967) a uvádza ho napr. Minieka (1978): využívajú sa podobné myšlienky ako v časti 8.C pri hľadaní najcennejšieho párenia. Najlacnejšie hranové pokrytie možno uvažovať aj ako špeciálny prípad úlohy z cvičenia 7(b) ($d'_i=1$, $d''_i=n-1$). Úlohu najlacnejšieho hranového pokrytia však možno pretransformovať na úlohu najlacnejšieho párenia aj v čase $O(n^2)$: v tomto čase možno urobiť aj obrátenú transformáciu [Plesník (1982a)].
10. V časti 4.D sme redukovali úlohu nájsť najkratšiu $s$—$t$ cestu v grafe na úlohu o faktore za predpokladu, že neexistujú cykly zápornej dĺžky. Uveďte, ako to zistíme.
11. Ukážte, že nezávisle od existencie polynomiálneho algoritmu pre najcennejšie párenie sú úlohy najcennejšieho párenia a najcennejšieho úplného párenia polynomiálne ekvivalentné. Ďalej ukážte, že stačí uvažovať len úlohy s kladnými cenami.
12. Dokážte nasledujúce zosilnenie tvrdenia z cvičenia 1 pre grafy s kladnými cenami hrán: Pažravá metóda (pozri 2.E) poskytuje $\frac{1}{2}$-aproximáciu najcennejšieho párenia, ale nezaručuje $\varrho$-aproximáciu pre $\varrho>\frac{1}{2}$.
13. Dokážte, že úloha nájsť kubický podgraf grafu je NP-ťažká [V. Chvátal — pozri v [Garey, Johnson (1979)]]. [Návod: Transformujte problém presného pokrytia 3-množinami.]
    Úloha je NP-ťažká dokonca aj pre planárne bipartitné grafy s maximálnym stupňom 4 [Plesník [1981b)].
14. Dokážte, že množina všetkých párených vrcholov v ľubovoľne zvolenom maximálnom párení predstavuje 2-aproximáciu minimového vrcholového pokrytia hrán grafu [Garey, Johnson (1979)].



# 9
# POCHÔDZKY

Úlohy skúmané v tejto kapitole vychádzajú z klasických úloh nájsť eulerovský ťah, resp. hamiltonovský cyklus. V prvom prípade ide o uzavretý ťah obsahujúci každú hranu grafu a v druhom o cyklus obsahujúci všetky vrcholy grafu. Tieto úlohy sa prirodzeným spôsobom prenášajú do digrafov, resp. migrafov. Významnejšie sú však zovšeobecnenia týchto úloh, kde sú hranám priradené dĺžky a úlohou je nájsť najkratšiu „pochôdzku". Takto vzniká úloha obchodného cestujúceho alebo úloha čínskeho poštára. Hoci zdanlivo ide o podobné úlohy, hľadať eulerovské ťahy je ľahké, zatiaľ čo úloha nájsť hamiltonovský cyklus patrí k veľmi obťažným úlohám. Toto platí aj v ohodnotených grafoch, hoci za istých predpokladov vieme nájsť aspoň aproximáciu optima.

## 9.A
## Eulerovské ťahy

Taký uzavretý ťah pseudomigrafu, ktorý obsahuje všetky hrany a vrcholy, nazýva sa *eulerovský ťah*. Ak má pseudomigraf takýto ťah, tak sa nazýva *eulerovský*. Tento pojem budeme skúmať iba v grafoch, digrafoch a migrafoch, ale všetky výsledky a dôkazy zostávajú platné aj pri násobných hranách a slučkách. V súvislosti s tzv. problémom 7 mostov dal Euler (1736) úplnú charakterizáciu takýchto grafov:

**Veta 1.** Graf $G$ je eulerovský $\Leftrightarrow$ (i) je súvislý a (ii) každý jeho vrchol má párny stupeň.

DÔKAZ. ($\Rightarrow$) Eulerovský ťah zabezpečuje súvislosť a počet príchodov do nejakého vrcholu je rovný počtu odchodov z tohto vrcholu.

($\Leftarrow$) Začínajúc z ľubovoľného vrcholu $v_0 \in E(G)$ cestujme stále po neprejdených hranách. Takéto cestovanie zrejme skončí vo vrchole $v_0$, lebo platí (ii), a teda sme utvorili nejaký $v_0$—$v_0$ ťah $T$. Ak $E(T)=E(G)$, tak $T$ je eulerovský ťah. Inak utvoríme dlhší $v_0$—$v_0$ ťah takto: Zvolíme netriviálny komponent $C$ v $G-E(T)$. Podľa (i), $T$ a $C$ majú nejaký spoločný vrchol $w$, a tak, ako sme utvorili $T$ v $G$, tak v $C$ možno utvoriť nejaký netriviálny $w$—$w$ ťah $Q$. Ak do ťahu $T$ vsunieme ťah $Q$,

získame $v_0$—$v_0$ ťah $T'$ dlhší ako $T$. Takéto predlžovanie nás zrejme privedie k eulerovskému ťahu. ■

## LABYRINTOVÝ ALGORITMUS PRE EULEROVSKÝ ŤAH

Dôkaz vety 1 je zároveň algoritmus na zostrojenie eulerovského ťahu. Edmonds a Johnson (1973) navrhli efektívnejší postup na báze Tarryho algoritmu (3.A) pre prieskum labyrintu, a preto nasledujúci postup nazvali labyrintový algoritmus. Jeho zložitosť je $O(m+n)$.

V danom grafe $G$ zvolíme ľubovoľne vrchol $v_0$. Začínajúc vo $v_0$ vytvárame sled dotiaľ, pokiaľ to dovoľujú nižšie uvedené pravidlá (P1) a (P2). Pritom si zaznamenávame (ako pri Tarryho algoritme) smer prechodu po hranách a pre každý vrchol $v \neq v_0$ aj hranu prvého príchodu do $v$. Okrem toho si zaznamenávame hrany v tom poradí, ako nimi prechádzame druhýkrát, t. j. spätne. Táto postupnosť hrán sa nazýva *spätná postupnosť*.

(P1) V jednom smere môžeme hranou prechádzať iba raz.

(P2) Pri odchode z nejakého vrcholu $u$ uprednostňujeme hrany v tomto poradí:
- (a) hrany neprejdené ani raz:
- (b) hrany prejdené iba v smere do $u$, okrem hrany prvého príchodu:
- (c) hrana prvého príchodu. ●

Uvedený postup je špeciálnym prípadom Tarryho algoritmu, a preto, ak už nemožno pokračovať, tak každá hrana komponentu obsahujúceho $v_0$ je prejdená každým smerom práve raz. Nech $T$ označuje príslušnú spätnú postupnosť.

**Veta 2.** Pre získanú spätnú postupnosť platí: $T$ je eulerovský ťah grafu $G \Leftrightarrow G$ je eulerovský.

DÔKAZ. ($\Rightarrow$) Zrejmé. ($\Leftarrow$) $T$ obsahuje každú hranu práve raz, a preto stačí dokázať, že $T$ je sled. Predovšetkým vidíme (stupne vrcholov sú párne), že prvá hrana v $T$ odchádza z $v_0$. Pre indukciu predpokladajme, že prvých $k \geq 1$ hrán v $T$ tvorí sled $v_0v_1 \ldots v_k$ (vo vrcholovom vyjadrení). Ukážeme, že aj ďalšia hrana $xy$ v $T$ odchádza z $v_k$. Nech $x \neq v_k$. Potom sme z $v_k$ prechádzali do $x$ len po hranách neprejdených ani v jednom smere; preto $x \neq v_i$ aj pre každé $i = 0, 1, \ldots, k-1$ (prednosť hrán). Potom však $xy$ nemôže existovať, lebo $x$ má párny stupeň. ■

## EULEROVSKÉ DIGRAFY A MIGRAFY

V prípade digrafov máme nasledujúcu analógiu vety 1: jej dôkaz je podobný a prenechávame ho čitateľovi (9.Z.2(a)).

**Veta 3.** Digraf $G$ je eulerovský $\Leftrightarrow$ (i) $G$ je slabo súvislý a (ii) $G$ je rovnovážne orientovaný.





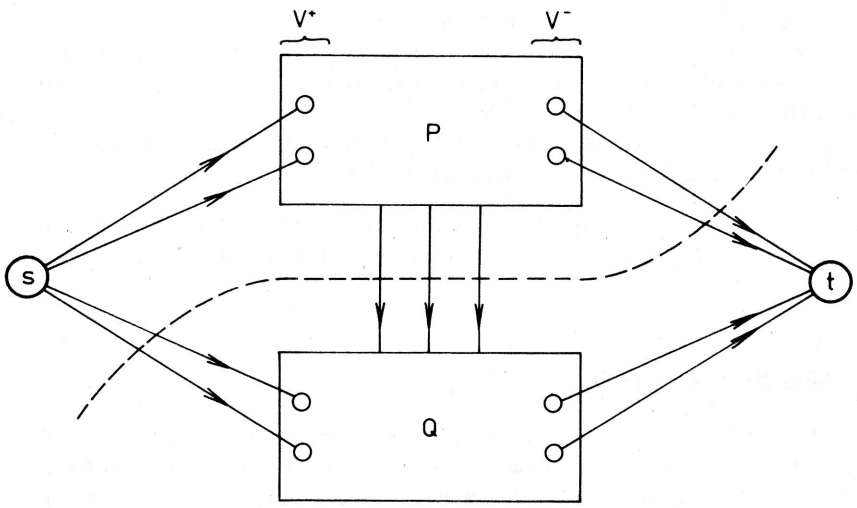
Tarryho prieskumom digrafu (pozri 3.E) získame tiež analogický labyrintový algoritmus (9.Z.2(b)). Situácia je však ťažšia pre migrafy, ako vidieť z nasledujúceho Fordovho a Fulkersonovho výsledku (1962).

**Veta 4.** Migraf $G$ je eulerovský $\Leftrightarrow$ sú splnené nasledujúce podmienky:
(i) $G$ je slabo súvislý,
(ii) celkový stupeň každého vrcholu je párny,
(iii) pre každú množinu $S \subset V(G)$ je $||(S, \bar{S})_G| - |(\bar{S}, S)_G|| \leq |[S, \bar{S}]_G|$, kde $\bar{S} := V(G) - S$, prvé dva rezy sa týkajú šípov a posledný rebier.

DÔKAZ. ($\Rightarrow$) Orientujúc rebrá v $G$ v smere, ktorým prechádza eulerovský ťah, získame z $G$ nejaký multidigraf $H$, kde $|(S, \bar{S})_H| = |(\bar{S}, S)_H|$.

($\Leftarrow$) Najprv časť rebier zorientujeme tak, aby v získanom migrafe $H$ bolo $\deg^+_H(v) = \deg^-_H(v)$ pre každé $v$, a potom už ľahko zostrojíme v $H$ eulerovský ťah, ktorý bude zrejme aj eulerovským ťahom v $G$.

Požadovanú orientáciu nájdeme transformáciou na tokovú úlohu v istom digrafe $F$. Nech pre každý vrchol $v$ je $d(v) := \deg^-_G(v) - \deg^+_G(v)$. Potom: (1) vynecháme z $G$ všetky šípy; (2) každé rebro nahradíme dvojicou protišípov s kapacitami 1 a (3) pridáme dva nové vrcholy $s$ a $t$, pričom $s$ spojíme s každým vrcholom $v \in V^+ := \{v | d(v) > 0\}$ šípom $sv$ kapacity $d(v)$ a každý vrchol $v \in V^- :=$




Obr. 1. Minimový rez v $F$.

$\{v|d(v)<0\}$ spojíme s $t$ šípom $vt$ kapacity $-d(v)$. V takto utvorenom digrafe $F$ nájdeme maximálny celočíselný $s$—$t$ tok $x$, ktorého veľkosť je rovná priepustnosti minimového $s$—$t$ rezu $(\{s\}\cup P, Q\cup\{t\})_F$, ako ilustruje obr. 1. Budeme predpokladať, že $P$ je minimálna množina a dokážeme, že $P=\emptyset$.

Nech $P\neq\emptyset$. Pre kapacity šípov platí: $b(\{s\}, V^+\cap P) > b(P, Q) + b(P\cap V^-, \{t\})$ alebo po prepísaní:

$$\sum_{v\in V^+\cap P} d(v) + \sum_{v\in P\cap V^-} d(v) > |[P, Q]_G|.$$

Podľa definície čísel $d(v)$ z toho máme:

$$(*) \qquad \sum_{v\in P} \deg_G^-(v) - \sum_{v\in P} \deg_G^+(v) > |[P, Q]_G|.$$

Avšak $\sum_{v\in P} \deg^+(v) = |(P, Q)_G| + |(P, P)_G|$ a $\sum_{v\in P} \deg^-(v) = |(Q, P)_G| + |(P, P)_G|$. Takže pre $S := Q$ je $\bar{S} = P$ a z $(*)$ získame spor s predpokladom (iii). Preto $P=\emptyset$. Potom teda tok $x$ nasycuje všetky hrany odchádzajúce z $s$ a tiež všetky hrany prichádzajúce do $t$ (lebo $\sum_{v\in V(G)} d(v)=0$).

Teraz utvoríme z $G$ multimigraf $H$ takto: Každé rebro $[u, v]\in E(G)$ bude rebrom aj v $H$, ak tok $x(u, v) = x(v, u)$: inak namiesto rebra $[u, v]$ dáme ten zo šípov $(u, v)$ a $(v, u)$, na ktorom bola hodnota toku $x$ väčšia (t. j. rovná 1). Zrejme platí: $\deg_H^+(v) = \deg_H^-(v)$ pre každý vrchol $v$.

Nech rebrá multimigrafu $H$ vytvárajú faktor $H^0$ a šípy faktor $H^1$. Každý komponent grafu $H^0$ je podľa vety 1 eulerovský a každý slabý komponent multidigrafu $H^1$ je tiež eulerovský. Z predpokladu (i) vyplýva, že jednotlivé eulerovské ťahy možno ľahko pospájať do jediného eulerovského ťahu migrafu $H$. Tento bude aj eulerovským ťahom v $G$. ■

Uvedený dôkaz poskytuje algoritmus pre nájdenie eulerovského ťahu v migrafe. Jeho zložitosť je určená zložitosťou algoritmu pre nájdenie maximálneho toku v $F$.

## 9.B
## Úloha čínskeho poštára

Vtedy, keď eulerovský ťah neexistuje, uspokojíme sa aj s *eulerovským sledom*, čo je uzavretý sled obsahujúci všetky hrany a vrcholy. Pravda, v tomto prípade nás bude zaujímať najkratší taký sled, ako je to aj v nasledujúcej úlohe, kde pristupujú ohodnotenia hrán.





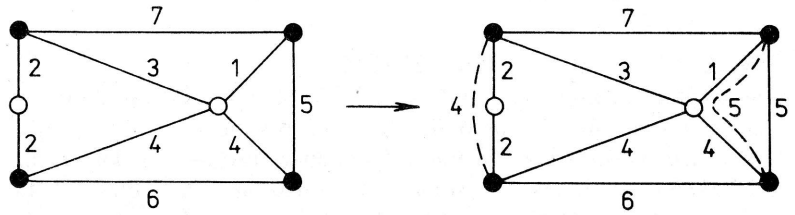
Úloha čínskeho poštára: V danom silne súvislom migrafe (resp. aj pseudomigrafe) $G$, kde každá hrana $ij$ má reálnu dĺžku $c_{ij}\geq 0$, treba nájsť najkratší eulerovský sled.

Ani tu netreba uvažovať slučky a násobné hrany, lebo napr. vsunutím nových vrcholov sa ich zbavíme. Uvážime zvlášť úlohu v grafoch, digrafoch a migrafoch.

## ČÍNSKY POŠTÁR V GRAFOCH

Táto úloha je historicky prvou. Pôvodne ju navrhol a riešil čínsky matematik Kwan (1962) v súvislosti s úlohou poštára, ktorý má prejsť všetky ulice svojho mesta (obvodu) a vrátiť sa späť. Odtiaľ je aj názov úlohy. Kwan navrhol začať s ľubovoľným prípustným riešením a postupne ho vylepšovať. Avšak pri tom vznikali úlohy identifikácie cyklov zápornej dĺžky, ktoré už neriešil.

Schodnejší sa ukázal prístup, ktorý ako prví uvádzajú Busacker a Saaty (1965) a nezávisle Edmonds (1965c): Ak má graf všetky stupne párne, tak každý eulerovský ťah je optimálnym riešením úlohy. Inak treba ísť niektorými hranami viackrát. Preto sa ponúka myšlienka popáriť vrcholy nepárneho stupňa novými hranami (najkratšími cestami), čím sa získa eulerovský multigraf, ktorého eulerovský ťah zodpovedá eulerovskému sledu v $G$. Obr. 1 ilustruje túto ideu.




Obr. 1. Optimálne doplnenie grafu na eulerovský multigraf.

Takýto postup je zdôvodnený nasledujúcimi úvahami. Nech $S^*$ je eulerovský sled v $G$. Nech $G^*$ je multigraf získaný z $G$ pridaním tzv. umelých hrán ku $G$: pre každú hranu $e\in E(G)$, ak $S^*$ obsahuje hranu $e$ $k$-krát, tak ku $e$ pridáme $k-1$ násobných hrán dĺžky $c(e)$. Keďže sledu $S^*$ v $G$ vieme priradiť ťah v $G^*$, tak $G^*$ je eulerovský, a preto má všetky stupne párne. Nech $V_1$ je množina vrcholov nepárneho stupňa v $G$. Multigraf $H := G^* - E(G)$ má len umelé hrany a tú istú množinu $V_1$ vrcholov nepárneho stupňa ako $G$ a možno v ňom nájsť $q := |V_1|/2$ hranovo disjunktných ciest $u_1$—$v_1$, $u_2$—$v_2$, ..., $u_q$—$v_q$, kde $\{u_1, v_1, \ldots, u_q, v_q\} = V_1$. (Také cesty možno ľahko nájsť: Zvolíme $u_1\in V_1$ a nájdeme nepredĺžiteľný ťah $T_1$ začínajúci v $u_1$: zrejme $T_1$ končí v nejakom vrchole $v_1$ nepárneho stupňa a možno z neho vybrať $u_1$—$v_1$ cestu $P_1$. Potom zvolíme $u_2\in V_1 - \{u_1, v_1\}$

a v $H - E(P_1)$ analogicky nájdeme $u_2$—$v_2$ cestu $P_2$ atď.) Týmto cestám zodpovedajú cesty v $G$, ktoré sú určené tými istými postupnosťami vrcholov.

Nasledujúci algoritmus je preto zameraný na nájdenie systému ciest, ktoré spolu pária všetky vrcholy nepárneho stupňa a majú sumárnu dĺžku minimálnu. Jeho správnosť vyplýva z uvedených úvah.

**Algoritmus**

KROK 1: Ak $V_1 = \emptyset$, tak ($G$ je eulerovský) položíme $\hat{G} := G$ a ideme na krok 4. Inak pre každé dva vrcholy $u$, $v \in V_1$ nájdeme (podľa kap. 4) najkratšiu $u$—$v$ cestu $P_{uv}$ (jej dĺžka je $d_G(u, v)$).

KROK 2: V kompletnom grafe $K$ na množine vrcholov $V_1$ predpíšeme každej hrane $uv$ cenu rovnú $d_G(u, v)$ a nájdeme v $K$ najlacnejšie úplné párenie $M$ (podľa kap. 8; $M$ existuje, lebo $|V_1|$ je párne číslo).

KROK 3: Ku $G$ pridáme hrany párenia $M$, čím získame nejaký multigraf $\hat{G}$.

KROK 4: V eulerovskom multigrafe $\hat{G}$ nájdeme (podľa 9.A) nejaký eulerovský ťah $T$.

KROK 5: Z ťahu $T$ zostrojíme eulerovský sled $S$ grafu $G$ tak, že v ňom každú hranu $uv \in M$ nahradíme cestou $P_{uv}$. Potom $S$ je najkratší eulerovský sled grafu $G$; STOP. ●

## ČÍNSKY POŠTÁR V DIGRAFOCH

Pre tento problém Edmonds a Johnson (1973) navrhli takýto postup:

Keďže $G$ je silne súvislý, eulerovský sled v $G$ existuje. Nech $S^*$ je najkratší eulerovský sled. Utvorme multidigraf $G^*$ pridaním nových hrán ku $G$: ak $S^*$ obsahuje hranu $ij$ $k$-krát, tak k nej pridáme $k-1$ násobných hrán dĺžky $c_{ij}$. Takto $G^*$ je eulerovský a $c(G^*) = c(S^*)$. Preto tu vzniká úloha doplniť $G$ na rovnovážne orientovaný multidigraf $\hat{G}$ hranami minimálnej sumárnej dĺžky. Potom eulerovský ťah v $\hat{G}$ bude zodpovedať najkratšiemu eulerovskému sledu v $G$. Túto úlohu však možno sformulovať ako úlohu nájsť najlacnejší celočíselný tok s požiadavkami, ktorú vieme riešiť (pozri 6.D). Ak neznáma $x_{ij}$ označuje počet umelých hrán násobných s $ij \in E(G)$, tak máme úlohu:
minimalizovať

$$\sum_{ij \in E(G)} c_{ij} x_{ij}$$

pri ohraničeniach

$$\sum_{j \in V^+(i)} x_{ij} - \sum_{j \in V^-(i)} x_{ji} = d(i) := \deg^-(i) - \deg^+(i), \qquad i \in V(G),$$





$$\left.\begin{array}{l} x_{ij} \geq 0 \\ x_{ij} - \text{celé} \end{array}\right\} ij \in E(G).$$

(Úlohu vieme previesť na úlohu s jedným vstupom a výstupom a vyriešiť v polynomiálnom počte krokov aj algoritmom polocyklov a polociest, lebo kapacity hrán budú čísla $|d(i)|$.) ●

## ČÍNSKY POŠTÁR V MIGRAFOCH

Túto úlohu už nevieme dobre riešiť. Naviac Papadimitriou (1976) dokazuje:

**Veta 1.** Úloha čínskeho poštára v migrafoch je NP-ťažká, dokonca pre planárne neohodnotené migrafy s maximálnym (celkovým) stupňom 3.

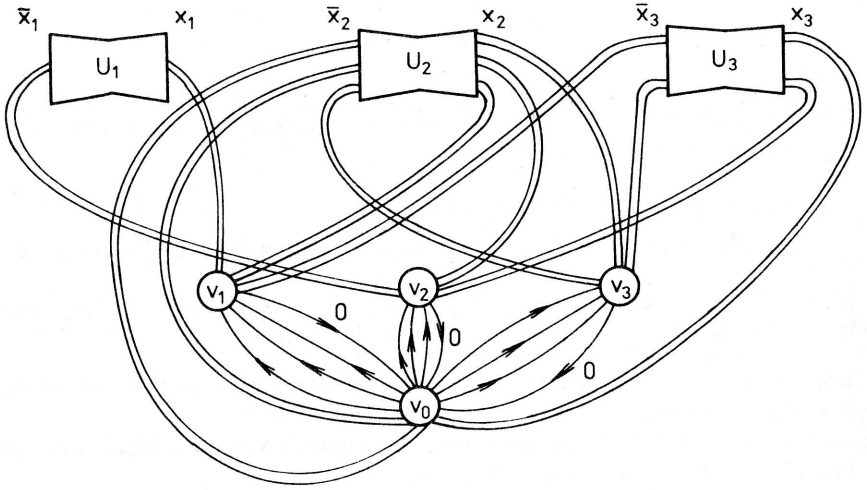
DÔKAZ. (1) Najprv dokážeme vetu pre ohodnotené migrafy bez ohraničení, a potom postupne prejdeme k jednotlivým ohraničeniam.

*Cenou eulerovského sledu* nazývame rozdiel jeho dĺžky a súčtu dĺžok všetkých hrán migrafu (táto cena je nulová pre eulerovské ťahy).

Na našu úlohu transformujeme problém splniteľnosti boolovského výrazu v konjunktívnej normálnej forme s tromi premennými v každej klauzule. Nech forma $F = C_1 \wedge C_2 \wedge \ldots \wedge C_q$, kde každá klauzula $C_i$ je 3-členná disjunkcia premenných $x_1, x_2, \ldots, x_p$, resp. ich negácií $\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_p$. Postup budeme ilustrovať na príklade, kde $F = (x_1 \vee x_2 \vee \bar{x}_3) \wedge (\bar{x}_1 \vee x_2 \vee x_3) \wedge (x_2 \vee \bar{x}_2 \vee \bar{x}_3)$: tu je $p = q = 3$. Nech pre každé $i$ označuje $s_i$ počet výskytov symbolu $x_i$ a $\bar{s}_i$ počet výskytov symbolu $\bar{x}_i$ v $F$. Položme $t_i = 2 \max\{s_i, \bar{s}_i\}$.

Predpokladajme, že pre každé $i$ máme k dispozícii tzv. *usmerňovač* $U_i$ kapacity $t_i$ a ceny $a_i$, ktorý predbežne budeme považovať za „dvojpólový vrchol" s vlastnosťami: Každý jeho pól inciduje s $t_i$ rebrami. Ak je zapojený v nejakom migrafe, tak prispieva k cene ľubovoľného eulerovského sledu $S$ príspevkom aspoň $a_i$. Pritom tento príspevok je presne $a_i$ práve vtedy, ak v $S$ prechádzame buď všetky rebrá pri ľavom póle v smere do $U_i$ a tie pri pravom póle smerom od $U_i$ alebo naopak, pričom každé rebro sa vyskytuje v $S$ iba raz. Neskôr ukážeme, že takýto usmerňovač možno realizovať migrafom.

K forme $F$ priradíme multimigraf $G$ s $p$ usmerňovačmi $U_1, U_2, \ldots, U_p$ a $q+1$ (obyčajnými) vrcholmi $v_0, v_1, \ldots, v_q$ nasledovne (pre náš príklad formy $F$ je $G$ na obr. 2): Pre každé $i = 1, 2, \ldots, p$ a každé $j = 1, 2, \ldots, q$ vedieme dve rebrá z $v_j$ do ľavého pólu v $U_i$, ak $C_j$ obsahuje $\bar{x}_i$ a do pravého pólu, ak $C_j$ obsahuje $x_i$. Z voľby $t_i$ je vidieť, že takto zaplníme aspoň jeden pól v $U_i$. Pre úplné zaplnenie vedieme potrebný počet dvojíc rebier do $v_0$. Všetky rebrá majú dĺžku 1 (na obrázkoch vyznačujeme len nejednotkové dĺžky hrán). Ďalej pre každé $j$ pridáme tri násobné šípy $v_0 v_j$ dĺžky 1 a jeden šíp $v_j v_0$ dĺžky 0.




Obr. 2. Konštrukcia multimigrafu $G$ pre formu $F=(x_1 \vee x_2 \vee \bar{x}_3) \wedge (\bar{x}_1 \vee x_2 \vee x_3) \wedge (x_2 \vee \bar{x}_2 \vee \bar{x}_3)$.

Dokážeme, že v $G$ existuje eulerovský sled ceny nanajvýš $a := a_1 + \ldots + a_p$ práve vtedy, keď je forma $F$ splniteľná.

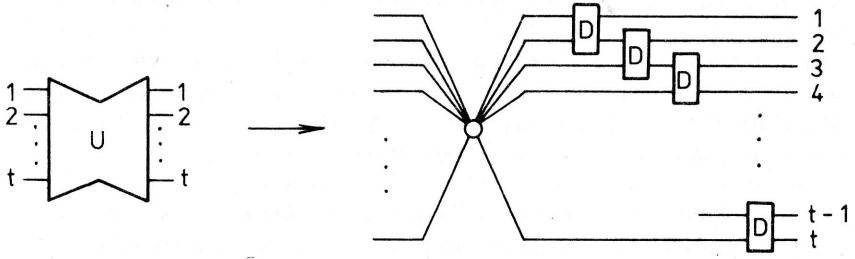
(i) Nech $F(x_1^*, x_2^*, \ldots, x_p^*) = 1$ (pravdivá hodnota). Potom všetky rebrá pri $U_i$ zorientujeme zľava doprava (prichádzajúce sú pri $\bar{x}_i$ a odchádzajúce pri $x_i$), ak $x_i^* = 1$ a sprava doľava, ak $x_i^* = 0$. Keďže každá klauzula $C_j$ je splnená, tak pri každom $v_j$ je aspoň jedna dvojica násobných hrán prichádzajúcich do $v_j$. Doteraz máme preto zabezpečené, že $\deg^-(v_j) \geq 5$, a teda, že $\deg^+(v_j) \leq 5$ (lebo celkove je pri každom $v_j$ 10 hrán). Ak $\deg^-(v_j) - \deg^+(v_j) > 0$, tak pridáme toľko kópií šípa $v_j v_0$ (dĺžky 0), až bude tento rozdiel nulový. Získaný multidigraf $\hat{G}$ má všetky vrcholy $v_1, \ldots, v_q$ aj „vrcholy" $U_1, \ldots, U_p$ rovnovážne ($\deg^+ = \deg^-$), a preto aj vrchol $v_0$ musí byť taký. Takto $\hat{G}$ má eulerovský ťah $\hat{T}$, ktorému zodpovedá eulerovský sled $T$ v $G$ tej istej dĺžky (šípy $v_j v_0$ majú dĺžku 0). Podľa predpokladu o usmerňovačoch $U_1, \ldots, U_p$, sled $T$ má cenu $a$.

(ii) Obrátene, nech $G$ má eulerovský sled $T$ ceny nanajvýš $a$. Z príspevkov jednotlivých $U_i$ vyplýva, že musí mať cenu práve $a$. Okrem toho musia byť všetky hrany pri $U_i$ prechádzané buď zľava doprava alebo všetky sprava doľava. Keďže už žiadny ďalší príspevok k cene sledu nie je možný, musí sled $T$ pre každý vrchol $v_j$ prechádzať aspoň jednou dvojicou hrán od nejakého $U_i$ k $v_j$, lebo inak by musel opakovane prechádzať nejakou hranou nenulovej dĺžky. Potom však postup v (i) možno obrátiť a získať splnenie formy $F$.

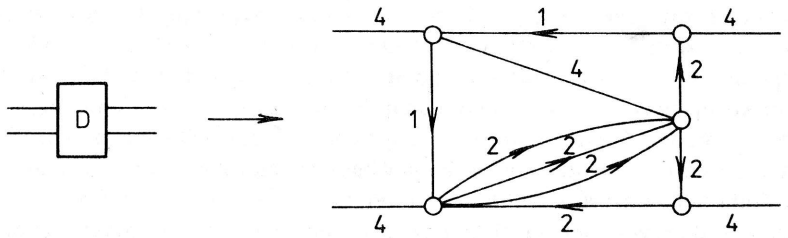




(2) Teraz ukážeme, že usmerňovače možno simulovať migrafmi. Menovite usmerňovač kapacity $t$ môžeme nahradiť štruktúrou z obr. 3, kde $D$ je usmerňovač kapacity 2 a je realizovaný na obr. 4. Prenechávame čitateľovi overiť, že $D$ je správne realizovaný a má cenu 4. Takto potom usmerňovač kapacity $t$ má podľa obr. 3 cenu 4 $(t-1)$. Teda v časti (1) tohto dôkazu kladieme $a_i = 4(t_i - 1)$. Aby sme boli presní, teraz vyžadujeme, aby v $G$ (porovnaj obr. 2) mali všetky rebrá dĺžku 4, a nie 1. Časť (1) dôkazu však zrejme platí aj vtedy (dôležité bolo iba to, že majú nenulovú dĺžku). Tým je ukázané, že problém je NP-ťažký v multimigrafoch (bez ohraničení).

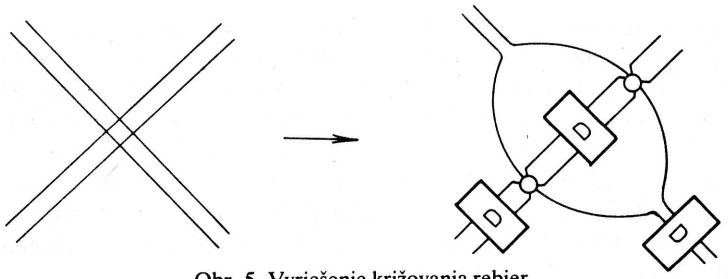



Obr. 3. Usmerňovač kapacity $t$.




Obr. 4. Realizácia usmerňovača kapacity 2 a ceny 4.

(3) Toto tvrdenie teraz zosilníme ohraničením na planárne migrafy. Samotné $D$ je planárne, a teda aj každé $U_i$ (obr. 4 a 3). Môžeme predpokladať, že migraf $G$ je zakreslený v rovine tak, že násobné hrany sú vedené blízko seba, že pretínajú sa vždy dvojice rebier a že v dostatočne malom kruhu sa nachádzajú priesečníky nanajvýš dvoch dvojíc rebier (porovnaj obr. 2). Potom križovanie možno vyriešiť tak, ako na obr. 5. Pravda, pri každej takejto substitúcii musíme celkovú cenu $a$ zväčšiť o 12. Čitateľovi nechávame overiť, že cena eulerovského sledu sa zväčší



Obr. 5. Vyriešenie križovania rebier.

práve o 12 iba vtedy, ak tento sled zachováva smer prechádzania rebrami z ľavej časti obr. 5: inak sa zväčší o viac.

(4) Teraz odstránime veľké stupne. Ak nejaký vrchol $w$ má stupeň väčší ako 3, tak zvolíme dve hrany $e_1$, $e_2$ incidentné s $w$ a ležiace v tej istej stene rovinného (diagramu) multimigrafu. Potom pridáme nový vrchol $w'$ a hrany $e_1$, $e_2$ odpojíme od $w$ a pripojíme ich do $w'$. Ďalej pridáme rebro $ww'$ nulovej dĺžky. Tým sme znížili stupeň vrcholu $w$ o 1 a utvorili len vrchol stupňa 3 bez porušenia planárnosti. Ľahko zistiť, ako sa transformuje najkratší eulerovský sled a že jeho cena sa nemení. Opakovaním tohto postupu získame požadované stupne ($\leqslant 3$).

(5) Nejednotkové dĺžky hrán odstránime takto: Nech $m$ je počet hrán v doteraz získanom multimigrafe $H$. Dĺžku $c(e)$ každej hrany $e$ nahradíme dĺžkou $c'(e)=1$, ak $c(e)=0$ a dĺžkou $c'(e) = m^2c(e)$, ak $c(e)>0$. Tvrdíme: pri dĺžkovej funkcii $c$ existuje eulerovský sled $S$ ceny nanajvýš $A$ práve vtedy, keď pri dĺžkovej funkcii $c'$ existuje eulerovský sled $S'$ ceny nanajvýš $(A+1)m^2-1$. Môžeme predpokladať, že $S$ neobsahuje viac ako $m^2-1$ hrán dĺžky 0. Potom tento istý sled pri dĺžkovej funkcii $c'$ má cenu nanajvýš $m^2A+(m^2-1)1$. Obrátene, ak rozdelíme hrany z $S'$ prispievajúce do ceny sledu na hrany dĺžky aspoň $m^2$ a hrany jednotkovej dĺžky, tak hrany prvej skupiny budú prispievať k cene číslom nanajvýš $A$ a druhé nulou pri dĺžkovej funkcii $c$. Preto za $S$ môžeme zobrať $S'$.

Ak teraz do každej hrany $e$ dĺžky $c'(e)>1$ vsunieme $c'(e)-1$ nových vrcholov, hranu $e$ rozdelíme na hrany jednotkovej dĺžky.

Tým sme zabezpečili, že každá hrana bude mať jednotkovú dĺžku. Zároveň sme sa zbavili násobných hrán.

Triviálne možno overiť, že všetky transformácie, ktoré sme urobili, sú polynomiálne. ■

## $k$ ČÍNSKYCH POŠTÁROV

Ak namiesto jedného máme $k \geqslant 2$ poštárov, tak budeme požadovať, aby v zjednotení obsiahli všetky hrany a pritom žiaden z nich neprešiel príliš mnoho.





Formálne: Nech $G$ je pseudomigraf, kde každá hrana $e$ má reálnu dĺžku $c(e) \geqslant 0$. Pre daný vrchol $v_1 \in V(G)$ treba nájsť $k$ takých $v_1$—$v_1$ sledov $S_1, S_2, \ldots, S_k$, ktoré v zjednotení obsahujú všetky hrany a pritom $\max\{c(S_j) | 1 \leqslant j \leqslant k\}$ je minimálne.

Túto úlohu študujú Fredericson, Hecht a Kim (1978) a nasledujúci výklad je založený na ich článku.

**Veta 2.** Pre každé $k \geqslant 2$ je úloha $k$ čínskych poštárov NP-ťažká, dokonca pre grafy alebo digrafy.

DÔKAZ. Na takúto úlohu transformujeme problém rozdelenia čísel $a_1, \ldots, a_p$ na $k$ tried s rovnakými súčtami. K tomu stačí zobrať za $G$ pseudograf (alebo pseudodigraf) s jediným vrcholom $v_1$ a $p$ slučkami dĺžok $a_1, \ldots, a_p$. Aby sme sa vyhli slučkám, vsunieme do každej dva vrcholy, pričom stredná hrana dostane dĺžku slučky a obe krajné hrany budú mať nulovú dĺžku. ■

V prípade pseudografov však možno zabezpečiť v polynomiálnom čase aspoň istú aproximáciu optima.

Nasledujúci *štiepiaci algoritmus* pre pseudograf $G$ štiepi optimálny sled $S^*$ z problému jedného čínskeho poštára na $k$ úsekov, ktoré po predĺžení do vrcholu $v_1$ dávajú množinu prípustných „takmer rovnakých" sledov pre $k$ poštárov. Najprv zavedieme niekoľko označení.

Nech $S^* = (v_1, e_1, v_2, e_2, \ldots, v_q, e_q, v_1)$, kde $v_i$ sú vrcholové a $e_j$ hranové prvky a nech $S^*_{i,j}$ označuje $v_i$—$v_j$ úsek sledu $S^*$. Ďalej označíme:

$$d_{\max} := \frac{1}{2} \max_{2 \leqslant j \leqslant q-1} \{d(v_1, v_j) + c(e_j) + d(v_{j+1}, v_1)\},$$

$$c_j := \frac{j}{k}[c(S^*) - 2d_{\max}] + d_{\max}, \quad \text{ak } 1 \leqslant j \leqslant k.$$

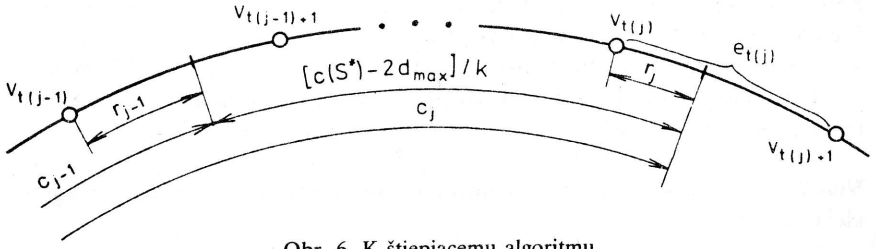
**Algoritmus**

KROK 1: Nájdeme najkratší eulerovský sled $S^*$ (úloha jedného čínskeho poštára).

KROK 2: Pre každé $j = 1, 2, \ldots, k-1$ urobíme:
Nájdeme najväčší index $t(j)$ taký, že $c(S^*_{1,t(j)}) \leqslant c_j$ (pozri obr. 6):
$r_j := c_j - c(S^*_{1,t(j)})$:

$$z(j) := \begin{cases} t(j), \text{ ak } r_j + d(v_{t(j)}, v_1) \leqslant c(e_j) - r_j + d(v_{t(j)+1}, v_1), \\ t(j)+1, \text{ inak.} \end{cases}$$

Okrem toho nech $z(0) := z(k) := 1$.




Obr. 6. K štiepiacemu algoritmu.

KROK 3: Pre každé $j=1, 2, \ldots, k$ úsek $S^*_{z(j-1),z(j)}$ predĺžime o najkratšiu $v_1$—$v_{z(j-1)}$ cestu a najkratšiu $v_{z(j)}$—$v_1$ cestu v $G$. Získané $v_1$—$v_1$ sledy $S_1, S_2, \ldots, S_k$ tvoria výstup algoritmu. STOP. ●

**Veta 3.** Nech $L^k := \max\{c(S_j) \mid 1 \leq j \leq k\}$ a nech $L^k_{\text{opt}}$ je hodnota optimálneho riešenia pre úlohu $k$ čínskych poštárov. Potom

$$\frac{L^k}{L^k_{\text{opt}}} \leq 2 - \frac{1}{k}.$$

DÔKAZ. Nech sledy $T_1, T_2, \ldots, T_k$ tvoria optimálne riešenie. Potom zrejme $c(S^*) \leq \Sigma\, c(T_j) \leq k\, L^k_{\text{opt}}$. Nech sa $d_{\max}$ realizuje pri indexe $g$. Hrana $e_{t(g)}$ leží v nejakom slede $T_i$, a preto

$$2d_{\max} = d(v_1, v_{t(g)}) + c(e_{t(g)}) + d(v_{t(g)+1}, v_1) \leq c(T_i) \leq L^k_{\text{opt}}.$$

Z definície čísla $d_{\max}$ pre každé $j$ platí:

$$(*)\quad \min\{d(v_1, v_{t(j)}) + r_j,\ c(e_{t(j)}) - r_j + d(v_{t(j)+1}, v_1)\} \leq d_{\max}.$$

Podľa toho, či $z(j) = t(j)$, alebo $t(j)+1$ a či $z(j-1) = t(j-1)$ alebo $t(j-1)+1$, sú štyri možnosti. Tu preskúmame len jednu (ostatné sú podobné): $z(j-1) = t(j-1)$, $z(j) = t(j)+1$.

Použijúc $(*)$ pre $j-1$ a pre $j$, získame:

$$d(v_1, v_{t(j-1)}) + r_{j-1} \leq d_{\max},$$

$$c(e_{t(j)}) - r_j + d(v_{t(j)+1}, v_1) \leq d_{\max}.$$

Na základe týchto a predchádzajúcich nerovností máme:

$$c(S_j) \leq d_{\max} + \frac{1}{k}[c(S^*) - 2d_{\max}] + d_{\max} \leq 2\left(1 - \frac{1}{k}\right) d_{\max} + \frac{1}{k}\, c(S^*) \leq \left(2 - \frac{1}{k}\right) L^k_{\text{opt}}. \blacksquare$$





## 9.C
## Úloha obchodného cestujúceho

*Hamiltonovský cyklus* migrafu je cyklus obsahujúci všetky vrcholy. Analogicky *hamiltonovský sled* je uzavretý sled obsahujúci všetky vrcholy. Ak cesta obsahuje všetky vrcholy, tak sa tiež nazýva *hamiltonovská,* hoci (s výnimkou triviálnej) nie je uzavretá. Je zrejmé, že migraf má hamiltonovský sled práve vtedy, keď je silne súvislý. Charakterizovať *hamiltonovské migrafy* (t. j. také, ktoré majú hamiltonovský cyklus) sa doteraz nepodarilo. Sú však známe viaceré postačujúce podmienky. Väčšina z nich je založená na myšlienke, že ak má migraf dostatočne veľké stupne, tak má hamiltonovský cyklus. Vzor pre také tvrdenia dal Dirac (1952): Ak v grafe $G$ s $n \geq 3$ vrcholmi je $\deg(v) \geq n/2$ pre každý vrchol $v$, tak $G$ má hamiltonovský cyklus. Pre digrafy máme [Ghouila–Houri (1960)]: Ak v silne súvislom digrafe $G$ je $\deg^+(v) + \deg^-(v) \geq n$ pre každý vrchol $v$, tak $G$ má hamiltonovský cyklus. Zosilnenie týchto výsledkov a iné tvrdenia možno nájsť napr. v [Behzad, Chartrand, Lesniak–Foster (1979)].

Prirodzeným zovšeobecnením na ohodnotené migrafy sa dostávame k tzv. *úlohe obchodného cestujúceho*: Daný je migraf $G$, v ktorom každá hrana $e$ má reálnu dĺžku $c(e) \geq 0$: treba nájsť najkratší hamiltonovský cyklus (stručne: NH *cyklus*).

Táto úloha je bezprostrednou formuláciou úlohy z praxe: obchodný cestujúci má navštíviť každé z $n$ miest práve raz, vrátiť sa do východzieho mesta, ale precestovať čo najmenej. Prvé zmienky o úlohe sa vyskytujú v súvislosti s pokusmi riešiť ju pomocou lineárneho programovania [Dantzig, Fulkerson, Johnson (1954)]. K tejto úlohe vedie aj vyberanie poštových schránok alebo vytvorenie optimálnej postupnosti technologických operácií (dĺžky hrán sú dané nákladmi na prestavovanie alebo čistenie strojov a pod.).

Pokiaľ sa obchodný cestujúci nevyhýba mestám, ktoré už navštívil, tak namiesto požiadavky navštíviť každé mesto práve raz, stačí žiadať, aby každé mesto navštívil aspoň raz, čo môže viesť k menšej dĺžke jeho pochôdzky (napr. podvodník sa obáva prísť niekam druhýkrát). Formálne: treba nájsť najkratší hamiltonovský sled (stručne: NH *sled*).

### TRANSFORMÁCIE ÚLOH

Uvedieme niektoré známe transformácie medzi zavedenými problémami.

(i) Rebro možno nahradiť dvojicou protišípov tej istej dĺžky (v probléme NH cyklu aj v probléme NH sledu), a tým previesť úlohu na digrafy.

(ii) Problém NH cyklu v grafe, resp. v digrafe, možno pretransformovať na problém NH sledu takto: Dĺžku každej hrany zväčšíme o veľké číslo $M$, napr. $M > \Sigma\, c(e)$. Potom NH cyklu pri pôvodných dĺžkach hrán zodpovedá NH sled

dĺžky nepresahujúcej $\Sigma\, c(e) + Mn$ pri nových dĺžkach a obrátene (lebo hamiltonovský sled, ktorý nie je cyklom, má aspoň $n+1$ hrán).

(iii) Platí aj obrátene. Problém NH sledu možno pretransformovať na problém NH cyklu: Daný graf (alebo digraf) $G$ prerobíme na kompletný graf (digraf) $\hat{G}$ na tých istých vrcholoch, pričom dĺžka $\hat{c}(uv)$ hrany $uv \in E(\hat{G})$ je rovná $d_G(u, v)$. Potom každému NH sledu v $G$ zodpovedá NH cyklus rovnakej dĺžky v $\hat{G}$ a obrátene.

(iv) Teraz si všimnime prechod z digrafov na grafy. Nasledujúcu transformáciu dáva Tarjan (pozri [Karp (1972)]): K digrafu $G$ priradíme graf $\tilde{G}$ takto: $V(\tilde{G}) := \{v^0,\ v^1,\ v^2 \mid v \in V(G)\}$, $E(\tilde{G}) := \{v^0v^1,\ v^1v^2 \mid v \in V(G)\} \cup \{v^0w^2 \mid vw \in E(G)\}$. Ľahko je nájsť korešpondenciu medzi hamiltonovskými cyklami v $G$ a hamiltonovskými cyklami v $\tilde{G}$. Problém NH sledu v digrafe možno potom pretransformovať na problém NH sledu v grafe pomocou (iii), (iv) a (ii).

(v) Konečne poznamenajme, že vo všetkých úlohách sa stačí obmedziť na kompletné grafy či digrafy (doplníme hranami veľkej dĺžky).

## OBŤAŽNOSŤ A APROXIMÁCIA

Už v kap. 2 sme videli, že nájsť hamiltonovský cyklus v grafe, resp. digrafe, sú NP-ťažké problémy, a teda takými sú aj úlohy NH cyklu. Z uvedených polynomiálnych transformácií potom vyplýva, že aj úlohy NH sledu sú NP-ťažké. Sahni a Gonzalez (1976) však ukazujú, že úloha obchodného cestujúceho je obzvlášť ťažká.

Pre dané reálne číslo $\varrho \geq 1$ nazvime úlohou $\varrho$-aproximácie NH cyklu, nájsť taký hamiltonovský cyklus $Z$ v kompletnom ohodnotenom grafe, resp. digrafe, že $c(Z)/c(Z^*) \leq \varrho$, kde $Z^*$ je NH cyklus. Analogicky možno hovoriť o úlohe $\varrho$-aproximácie NH sledu.

**Veta 1.** Pre každé $\varrho \geq 1$ je úloha $\varrho$-aproximácie NH cyklu NP-ťažká (v grafoch aj digrafoch).

DÔKAZ. Urobíme transformáciu problému nájsť hamiltonovský cyklus v grafe (dôkaz je rovnaký aj pre digraf) na náš problém $\varrho$-aproximácie. Daný graf $G$ doplníme hranami dĺžky $M > \varrho n - n + 1$ na kompletný graf $H$. Pôvodné hrany sú jednotkovej dĺžky. Nech $Z^*$ je NH cyklus v $H$ a nech $Z$ je riešenie úlohy $\varrho$-aproximácie NH cyklu v $H$. Ak $c(Z) = n$, tak $Z$ je hamiltonovský cyklus v $G$. Inak $c(Z) \geq n - 1 + M$. Keďže $c(Z)/c(Z^*) \leq \varrho$, tak aj $(n - 1 + M) \leq \varrho c(Z^*)$, a teda podľa voľby $M$, $\varrho n < \varrho c(Z^*)$. Takto $c(Z^*) > n$, a preto $G$ nemá hamiltonovský cyklus. ■





Podľa vety 1 máme malú nádej nájsť polynomiálny algoritmus aspoň pre približné riešenie úlohy NH cyklu. Avšak pre úlohu NH sledu v grafe sú už aproximačné algoritmy. Úloha aproximovať NH sled v digrafe zostáva teoreticky neriešená a v ďalšom sa obmedzíme len na grafy.

Doteraz bolo vytvorených mnoho heuristík pre úlohu obchodného cestujúceho (kvôli prehľadu pozri napr. [Christofides (1975 a 1979)]: rozsiahle experimentálne porovnávania rôznych heuristík dávajú Golden, Bodin, Doyle a Stewart (1980)). Rosenkrantz, Stearns a Lewis (1977) preskúmali viaceré heuristiky a ukázali, že niektoré z nich poskytujú prijateľnú aproximáciu optima za predpokladu takej dĺžkovej funkcie v kompletnom grafe, ktorá spĺňa trojuholníkovú nerovnosť ($c_{ij} + c_{jk} \geq c_{ik}$ pre každé tri vrcholy $i, j, k$). Tento predpoklad umožňuje ku každému NH sledu priradiť rovnako dlhý hamiltonovský cyklus. Na druhej strane hľadanie NH sledu môžeme uvedenou transformáciou (iii) previesť na hľadanie NH cyklu, čo môže byť niekedy prehľadnejšie (pri navrhovaní heuristík alebo pri konštruovaní rôznych príkladov).

Teraz uvedieme dve heuristiky. Predpokladáme, že je daný kompletný graf $G$ s ohodnoteniami $c(e) \geq 0$ spĺňajúcimi trojuholníkovú nerovnosť.

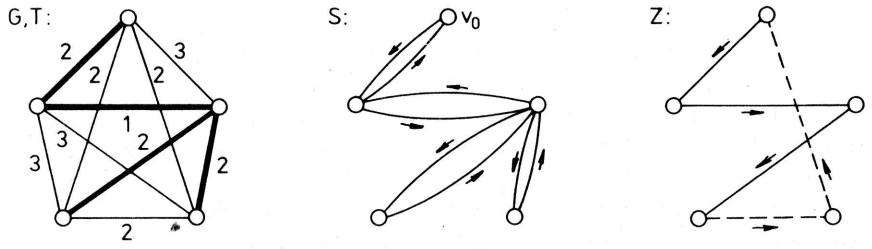
## METÓDA ZDVOJENIA KOSTRY

Túto metódu navrhol Kim (1975) a pozostáva z troch krokov:

KROK 1: Považujúc dĺžky hrán za ich ceny, nájdeme najlacnejšiu kostru $T$ (pozri kap. 5).

KROK 2: Zvolíme ľubovoľný vrchol $v_0$ a utvoríme hamiltonovský $v_0$—$v_0$ sled $S$ pre $T$ (napr. použitím Tarryho prieskumu), ktorý obsahuje každú hranu v $T$ práve 2-krát.

KROK 3: Sled $S$ nahradíme cyklom $Z$ takto: Prechádzame sledom $S$ a kedykoľvek by sme mali ísť znova po prejdených vrcholoch, tak príslušný $i$—$j$




Obr. 1. Kompletný graf a najlacnejšia kostra : hamiltonovský sled : hamiltonovský cyklus.

úsek sledu $S$ nahradíme hranou $ij$. Získaný cyklus $Z$ je aproximáciou NH cyklu v $G$: STOP. ●

Obr. 1 ilustruje uvedený postup.

Nech $Z^*$ je NH cyklus v $G$. Budeme predpokladať, že $c(Z^*)>0$ (inak z trojuholníkovej nerovnosti vyplýva, že všetky $c(e)=0$).

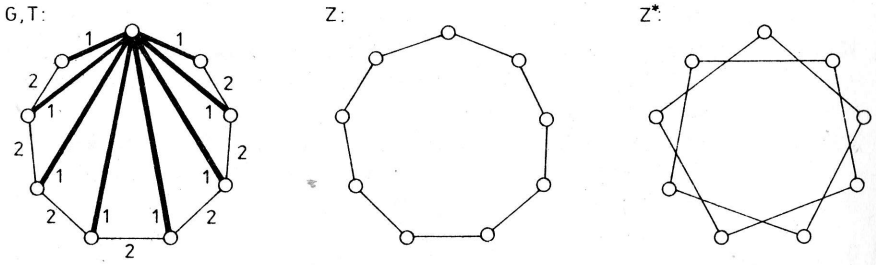
**Veta 2.** Metóda zdvojenia kostry dáva vždy hamiltonovský cyklus $Z$ s

$$\frac{c(Z)}{c(Z^*)}<2.$$

Naviac, ohraničenie 2 nemožno zmenšiť.

DÔKAZ. Keďže vynechaním najdlhšej hrany zo $Z^*$ získame kostru, tak $c(T)<c(Z^*)$. Ďalej máme $c(S) = 2c(T)$. Trojuholníková nerovnosť (prípadne jej opakovaná aplikácia) dáva: $c(Z)\leq c(S)$. Takto spolu máme: $c(Z)\leq 2c(Z^*)$.

Na druhej strane, uvažujme zovšeobecnenie príkladu z obr. 2. Ide o kompletný graf s nepárnym počtom vrcholov $n$, kde každá z $n-2$ vyznačených hrán na obvode má dĺžku 2 a každá iná dĺžku 1. Kostra $T$ je hrubo vyznačená. Získame $c(Z) = 2(n-2)+2 = 2n-2$, ale vidíme, že $c(Z^*)=n$. Preto $c(Z)/c(Z^*) = 2-2/n$, čo pre $n\to\infty$ dáva 2. ■




Obr. 2. Hraničný príklad pre metódu zdvojenia kostry.

Poznamenajme, že ak nám ide len o aproximáciu NH sledu v (nekompletnom) súvislom grafe $G$, tak prvé dva kroky metódy kostry nám stačia na nájdenie hamiltonovského sledu $S$ s $c(S)/c(S^*)\leq 2$.

Myšlienka aproximácie pomocou kostry sa vyskytla vo viacerých 2-aproximačných algoritmoch buď explicitne alebo implicitne. Niektoré z nich uvedieme v cvičeniach. Avšak žiadny z nich nedáva $\varrho$-aproximáciu pre $\varrho<2$.





## CHRISTOFIDESOVA METÓDA KOSTRY A PÁRENIA

Christofides (1976) dosiahol zlepšenie aproximácie tým, že namiesto zdvojenia kostry dopĺňa kostru na eulerovský multigraf pomocou párenia vrcholov nepárneho stupňa. Jeho metóda pozostáva z piatich krokov:

KROK 1: V $G$ nájdeme najlacnejšiu kostru $T$.
KROK 2: Určíme $V_0 := \{v\in V(G)\,|\,\deg_T(v)$ je nepárne$\}$ (zrejme $|V_0|$ je párne číslo).
KROK 3: V $G(V_0)$ nájdeme najlacnejšie úplné párenie $X$ (pozri kap. 8).
KROK 4: Pridáme hrany párenia $X$ ku kostre $T$ a v získanom eulerovskom multigrafe $\hat{T}$ nájdeme eulerovský ťah $S$.
KROK 5: K ťahu $S$ vytvoríme cyklus $Z$ preskakovaním vrcholov, ktoré sa opakujú (ako v kroku 3 metódy zdvojenia kostry): STOP. ●

**Veta 3.** Christofidesova metóda kostry a párenia dáva vždy hamiltonovský cyklus $Z$ s

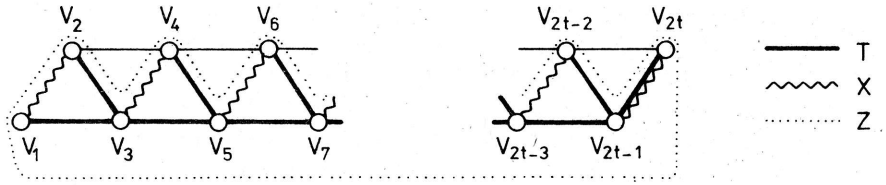
$$\frac{c(Z)}{c(Z^*)}<\frac{3}{2}.$$

Naviac, ohraničenie 3/2 nemožno zmenšiť.

DÔKAZ. Zrejme $c(T) < c(Z^*)$. Nech $Z_0^*$ je hamiltonovský cyklus v grafe $G(V_0)$, ktorý vznikne zo $Z^*$ nahradením úsekov nepatriacich do $V_0$ hranami z $G(V_0)$. Z trojuholníkovej nerovnosti máme $c(Z_0^*)\leq c(Z^*)$. Keďže $Z_0^*$ možno rozložiť na dve párenia grafu $G(V_0)$, tak $c(X)\leq \frac{1}{2}c(Z_0^*)$. Sumarizujúc máme:

$$c(Z)\leq c(\hat{T})=c(T)+c(X)<c(Z^*)+\frac{1}{2}c(Z_0^*)\leq\frac{3}{2}c(Z^*).$$

Cornuejols a Nemhauser (1978) ukázali, že pomer 3/2 nemožno znížiť. Na tento účel uvažujeme kompletný graf $G$ s $2t$ vrcholmi, ktorého podstatná časť je na




Obr. 3. Hraničný príklad pre Christofidesovu metódu.

obr. 3. Hrana $v_1v_{2t}$ má dĺžku $t$, ostatné nakreslené hrany majú dĺžku rovnú 1 a nenakreslené hrany $ij$ majú dĺžky rovné vzdialenostiam $d(i, j)$ v nakreslenom grafe. Hrany patriace do kostry $T$,párenia $X$ a cyklu $Z$ sú vyznačené. Vidíme, že $c(T) = 2t-1$, $c(X) = t$ a $c(Z) = 3t-1$. Avšak optimálny cyklus $Z^* = v_1v_3 \ldots v_{2t-1}v_{2t}v_{2t-2} \ldots v_4v_2v_1$ má $c(Z^*) = 2t$. Preto $c(Z)/c(Z^*) \to 3/2$ pre $t \to \infty$. ■

Poznamenajme, že lepší ako uvedený 3/2-aproximačný algoritmus nie je známy.

## 9.Z
## Cvičenia a poznámky

1. Dokážte, že problém nájsť eulerovský faktor grafu je NP-ťažký dokonca pre planárne bipartitné grafy s maximálnym stupňom 3 [Pulleyblank (1979)]. Tiež dokážte analógiu pre digrafy.
2. (a) Dokážte vetu 9.A.3. (b) Navrhnite lineárny algoritmus pre hľadanie eulerovského ťahu v digrafe. [Utvorte analógiu labyrintového algoritmu a obráťte spätnú postupnosť. Iný algoritmus dávajú Edmonds a Johnson (1973)].
3. Overte správnosť realizácie usmerňovača z obr. 9.B.4.
4. Dokážte, že v algoritme čínskeho poštára v grafe každé dve páriace cesty $P_{uv}$ a $P_{xy}$ majú nanajvýš jeden spoločný vrchol.
5. Preskúmajte variácie úlohy obchodného cestujúceho a čínskeho poštára, keď začiatok a koniec pochôdzky môžu byť rôzne, resp. sú predpísané. Pretransformujte ich na základné úlohy.
6. Ak $G$ je graf, tak nech $G^s$ označuje graf s $V(G^s) = V(G)$ a $xy \in E(G^s)$ ak $d_G(x, y) \leq s$. Sekanina (1960) dokázal, že ak $G$ je súvislý, tak $G^3$ má hamiltonovský cyklus. Ukážte, že zistiť, či $G^2$ má hamiltonovský cyklus je NP-ťažký problém. [Underground (1978) transformuje problém hamiltonovského cyklu tak, že ku každému vrcholu pridá cestu dĺžky 2.]
7. Uvedieme dva špeciálne prípady, kedy vieme pomerne ľahko vyriešiť úlohu obchodného cestujúceho.

   (a) Kompletný digraf $G$ s dĺžkami hrán $c_{ij} \geq 0$, kde $c_{ij} = 0$, ak $i \geq j$ $(i, j = 1, 2, \ldots, n)$. Lawler (1971) navrhuje utvoriť bipartitný digraf s partiami $V_1 = \{1', 2', \ldots, (n-1)'\}$, $V_2 = \{2'', 3'', \ldots, n''\}$ a hranami $i'j''$ dĺžky $c_{ij}$. Potom hrany najlacnejšieho párenia množiny $V_1$ do $V_2$ dávajú po prechode ku $G$ 1—$n$ cestu a nejaké cykly, ktoré možno manipuláciou s hranami nulovej dĺžky prerobiť na najkratší hamiltonovský cyklus. Upresnite a dokážte. (Podobnými





manipuláciami Gilmore a Gomory (1964) vyriešili jeden významný technologický problém.)

   (b) Nech $H$ je digraf a $G$ je tzv. hranový digraf, t. j. $V(G) = E(H)$ a $(e_1, e_2) \in E(G)$ práve vtedy, keď $e_1e_2$ je sled v $H$. Ukážte, že existuje bijekcia medzi eulerovskými ťahmi v $H$ a hamiltonovskými cyklami v $G$. Ďalej ukážte, že ak pre každú hranu $xy \in E(G)$ majú všetky hrany odchádzajúce z $x$ a všetky hrany prichádzajúce do $y$ rovnaké dĺžky, tak stačí nájsť nejaký eulerovský ťah v $H$ a zodpovedajúci cyklus v $G$ už bude optimálny [Sysło (1973)].
8. Uvažujte nasledujúce tri heuristiky pre úlohu obchodného cestujúceho v kompletnom grafe s nezápornými dĺžkami hrán spĺňajúcimi trojuholníkovú nerovnosť.

   (a) Ak máme vytvorený cyklus $C_i$ s vrcholmi $v_1, v_2, \ldots, v_i$ (na začiatku je $i = 2$ a $C_i$ je len najkratšia hrana), kde $2 \leq i < n$, tak nájdeme vrchol $v_{i+1}$ mimo $C_i$, ale najbližší k $C_i$. Pre $i = 2$ pridáme k $C_i$ hrany $v_1v_3$ a $v_2v_3$ a získame $C_3$. Pre $i \geq 3$ nájdeme v $C_i$ takú hranu $xy$, aby cyklus $C_{i+1} := C_i - xy + xv_{i+1} + yv_{i+1}$ bol najkratší. Výsledkom je cyklus $C_n$.

   (b) Zvolíme vrchol $v_1$. Ak máme vytvorenú cestu $P_i = v_1v_2 \ldots, v_i$, kde $i < n$, tak mimo $P_i$ nájdeme taký vrchol $v_{i+1}$, ktorý je najbližší k $v_i$. Potom $P_{i+1} := v_1v_2 \ldots v_{i+1}$. Výsledkom je cyklus $P_n + v_nv_1$.

   (c) Nájdeme tzv. 2-*optimálny* hamiltonovský cyklus, t. j. cyklus s vlastnosťou: po vynechaní ľubovoľných jeho dvoch hrán a pridaní iných dvoch hrán, nezískame kratší hamiltonovský cyklus.

   Dokážte, že žiadna z týchto heuristík nezaručuje $\varrho$-aproximáciu pre $\varrho < 2$. (Pre (a) možno položiť $\varrho = 2$, ale napr. pre (b) neexistuje pevné $\varrho$.) Rosenkrantz et al. (1977)].
9. Frederickson et al. (1978) študovali tzv. *úlohu autožeriavu:* Daný je migraf $G$ s nezápornými dĺžkami hrán a vrchol $v_0$; treba nájsť najkratší $v_0$—$v_0$ sled obsahujúci všetky šípy (napr. niektoré ulice treba prejsť v požadovanom smere, lebo na pravej strane sa bude nakladať).

   (a) Dokážte, že táto úloha je NP-ťažká.

Porovnaním výsledkov dvoch heuristík bol získaný 9/5-aproximačný algoritmus pre silne súvislý migraf, kde (i) pri každom vrchole je práve jeden šíp, (ii) pre každý šíp $ij$ dĺžky $c_{ij}$ existuje $j$—$i$ sled dĺžky nanajvýš $c_{ij}$. V prvej heuristike sa po kontrakcii šípov použije Christofidesova metóda (kostry a párenia). V druhej sa optimálne popária začiatky a konce šípov a vzniknuté cykly sa po kontrakcii spoja cez optimálnu kostru. Prvá heuristika lepšie aproximuje pri krátkych šípoch a druhá pri dlhých.

(b) Ukážte transformácie na zabezpečenie predpokladu (i).

Pre $k$ autožeriavov bol získaný štiepiaci $(9/5 + 1 + 1/k)$-aproximačný algoritmus.

10. Uvažujte nasledujúci všeobecný problém pochôdzky [Orloff (1974)]. Daný je pseudomigraf $G$ s nezápornými dĺžkami hrán a množina $Q \subseteq V(G) \cup E(G)$: úlohou je nájsť najkratší uzavretý sled v $G$ obsahujúci všetky prvky z $Q$. Ak $Q \subseteq E$, tak hovoríme o *úlohe vidieckeho poštára.*

(a) Dokážte, že úloha vidieckeho poštára je NP-ťažká v grafoch aj digrafoch [Lenstra, Rinnooy Kan (1976)].

(b) Zosilnite tvrdenie z (a) pre planárne kubické grafy (resp. digrafy) s jednotkovými dĺžkami hrán. [Návod: V probléme hamiltonovského cyklu nahraďte každý vrchol grafom s vrcholmi 1, ..., 7 a hranami 12, 16, 23, 25, 34, 45, 47, 56, 67; do $Q$ dajte hrany 12, 34 a 67.]

(c) Sformulujte všeobecný problém pochôdzky ako 0—1 úlohu lineárneho programovania.

# 10
# PLANÁRNE GRAFY

Mnoho reálnych objektov možno reprezentovať planárnymi grafmi, prípadne pseudomigrafmi (komunikačné siete na zemskom povrchu, tlačené elektronické spoje a i.). Uvedieme niektoré kvantitatívne vzťahy pre planárne grafy, potom si všimneme rôzne rovinné vnorenia, zastavíme sa pri charakterizácii planárnych grafov a nakoniec načrtneme algoritmus na zisťovanie planarity grafov.

V našich úvahách sa obmedzujeme takmer všade na grafy (niektoré z výsledkov platia len pre grafy a niektoré sa dajú triviálne rozšíriť).

Pripomeňme, že graf (alebo aj pseudomigraf) $G$ sa nazýva planárny, ak existuje rovinný diagram pre $G$ (t. j. taký, v ktorom sa hrany nepretínajú). Vtedy tiež hovoríme, že $G$ je vnoriteľný do roviny. Vrcholy sú reprezentované bodmi a hrany jordanovskými čiarami (definíciu a vlastnosti možno nájsť napr. v [Čech (1966)]). Jednoducho však hovoríme o vrcholoch a hranách. Ak z roviny vynecháme všetky body rovinného diagramu, tak zvyšná množina bodov roviny pozostáva z niekoľkých súvislých oblastí, ktoré sa nazývajú *steny* diagramu. Práve jedna stena je neohraničená a nazýva sa *vonkajšia*: ostatné steny sa nazývajú *vnútorné.* (Bodová) hranica každej steny zodpovedá istým vrcholom a hranám, ktoré spolu tvoria podgraf grafu $G$ a nazývame ho jednoducho *hranica.* Hranica môže byť aj nesúvislým podgrafom, ale ak $G$ je vrcholovo 2-súvislý, tak je to vždy cyklus (bodová hranica je vždy uzavretá jordanovská čiara).

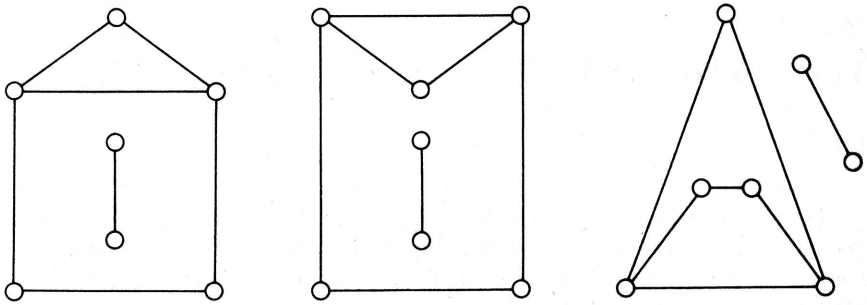
Poznamenávame, že často hovoríme o diagrame $G$, pod čím rozumieme diagram grafu $G$.

## 10.A
## Eulerov vzorec

Jeden graf môže mať viac rozličných diagramov, ako ilustruje obr. 1. (Slovo „rozličný" tu má intuitívny zmysel: spresnenie je v 10.B.) Avšak už Euler (1752) si v súvislosti s 3-rozmernými polyédrami všimol, že počet stien sa nemení.

**Veta 1** (Eulerov vzorec). Pre rovinný diagram $G$ pseudografu s $n$ vrcholmi, $m$ hranami, $r$ stenami a $k$ komponentmi platí:

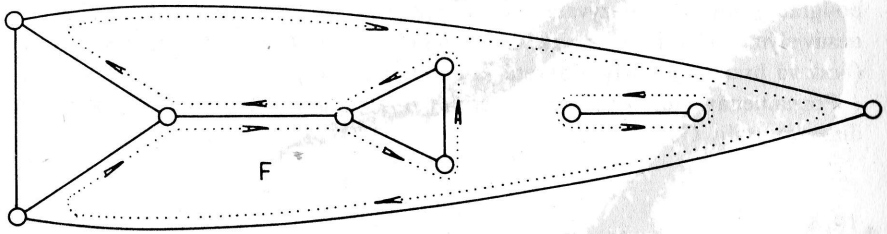
$$n - m + r = k + 1.$$



Obr. 1. Tri diagramy toho istého grafu.

DÔKAZ. (Indukciou podľa $m$.) Zvolíme ľubovoľnú hranu $e$ a uvážime, či je mostom alebo nie. Z predpokladu o $G_1 := G - e$ tvrdenie hneď vyplynie: V prvom prípade je $n_1 = n$, $m_1 = m - 1$, $r_1 = r$ a $k_1 = k + 1$; v druhom $n_1 = n$, $m_1 = n - 1$, $r_1 = r - 1$ a $k_1 = k$. ■

Nech $F$ je stena rovinného diagramu. Urobme „prechádzku tesne vedľa hranice" steny $F$ tak, že najbližší bod hranice budeme mať vždy vľavo (pozri obr. 2), až sa vrátime do východiskového bodu. Ak takto obídeme každý




Obr. 2. Určenie dĺžky hranice steny.

komponent hranice a pritom každej hrane, vedľa ktorej sme prešli, priradíme šípku, tak každá hrana steny $F$ získa jednu alebo dve šípky, pritom dve šípky získa práve vtedy, keď je mostom. Celkový počet priradených šípok nazveme *dĺžkou hranice* steny $F$.

Nasledujúce dôsledky vety 1 ukazujú, že planárne grafy majú pomerne málo hrán.





**Dôsledok 1(a).** Ak každá stena rovinného diagramu grafu má dĺžku hranice aspoň $g$, kde $g \geq 3$, tak

$$m \leq \frac{g}{g-2}(n-2).$$

DÔKAZ. Nech $r_i$ je počet takých stien, ktorých dĺžka hranice je $i$. Potom celkový počet stien $r = \Sigma r_i$ (tu aj ďalej $i$ prebieha hodnoty 3, 4, ...). Takto z Eulerovho vzorca získame:

$$gn - gm + g\Sigma r_i = g(1+k).$$

Ak urobíme prechádzku z obr. 2 pre každú stenu, tak zrejme každá hrana dostane práve dve šípky. Preto

$$2m = \Sigma i r_i.$$

Sčítaním posledných dvoch rovností získame:

$$(*) \qquad gn + \Sigma (g-i) r_i - g(1+k) = (g-2)m.$$

Keďže $k \geq 1$ a $r_i = 0$, ak $i < g$, tak dostávame $gn - 2g \geq (g-2)m$. ■

**Dôsledok 1(b).** Ak $G$ je planárny graf s $n \geq 3$ vrcholmi, tak $m \leq 3n - 6$.

DÔKAZ. Eventuálnym pridaním nových hrán získame z rovinného diagramu grafu $G$ súvislý rovinný diagram, kde každá stena už má dĺžku hranice aspoň 3. ■

Ak $G$ je planárny graf a pre každú novú hranu $e$ už $G + e$ nie je planárny graf, tak $G$ sa nazýva *maximálny planárny* graf. Ľahko sa dokáže, že v každom rovinnom diagrame maximálneho planárneho grafu $G$ je každá stena trojuholníkom (hranica steny je trojuholník), ak $n \geq 3$. Taký diagram sa nazýva *rovinná triangulácia*. Potom zo vzťahu (*) vidíme, že $m = 3n - 6$.

**Dôsledok 1(c).** Ak $G$ je planárny graf s $n \geq 3$ vrcholmi a bez trojuholníkov, tak $m \leq 2n - 4$.

DÔKAZ. Prípadným doplnením hranami možno získať súvislý rovinný diagram, kde každá stena má dĺžku hranice aspoň 4. ■

**Dôsledok 1(d).** Každý planárny graf s $n \geq 4$ vrcholmi má aspoň štyri vrcholy stupňa nanajvýš 5.

DÔKAZ. Tvrdenie stačí dokázať pre maximálne planárne grafy. V takom grafe však každý vrchol má stupeň aspoň 3. Ak by tam boli nanajvýš tri vrcholy stupňa nepresahujúceho 5, tak sčítajúc všetky stupne získame

$$2m = \Sigma \deg(v) \geq 3.3 + (n-3)6 = 2(3n-6)+3,$$

čo je spor s dôsledkom 1(b). ■

## 10.B
## Rôzne rovinné vnorenia

Podobne ako do roviny možno definovať vnorenia grafov do iných plôch (tiež žiadame, aby sa hrany nepretínali). V súvislosti s planárnymi grafmi je významná guľová plocha. Ľahko je totiž vidieť, že *graf je planárny práve vtedy, keď je vnoriteľný do guľovej plochy*. To možno dokázať napr. pomocou tzv. *stereografickej projekcie*: Ak sa guľa opiera o rovinu bodom $\bar{x}$ a $x$ je protiľahlý bod gule („severný pól"), tak úsečka $[x, y]$, kde $y$ je bod roviny, pretína guľovú plochu v jedinom bode $y'$. Takto možno zobraziť diagram z roviny do guľovej plochy. Zrejme inverzné zobrazenie zobrazí každý bod $y' \neq x$ z guľovej plochy do roviny, a teda ak guľu postavíme na rovinu tak, aby bod $x$ ležal vo vnútri steny sférického diagramu, tak získame rovinný diagram.

Teraz si upresníme, kedy budeme dva rovinné alebo aj sférické diagramy $D_1$, $D_2$ grafu $G$ považovať za „rovnaké". Hovoríme, že $D_1$ a $D_2$ sú *ekvivalentné*, ak každá hranica v $D_1$ je hranicou v $D_2$ a obrátene. Hovoríme, že $D_1$ a $D_2$ sú *silne ekvivalentné*, ak sú ekvivalentné a pri každom vrchole $u$ platí: hrany incidujúce s $u$ sa pri pravotočivom obiehaní vrcholu $u$ vyskytujú v $D_1$ a v $D_2$ v tom istom poradí.

Takto žiadne dva z diagramov na obr. 10.A.1 nie sú ekvivalentné. Obr. 1 ukazuje ekvivalentné, ale nie silne ekvivalentné diagramy.

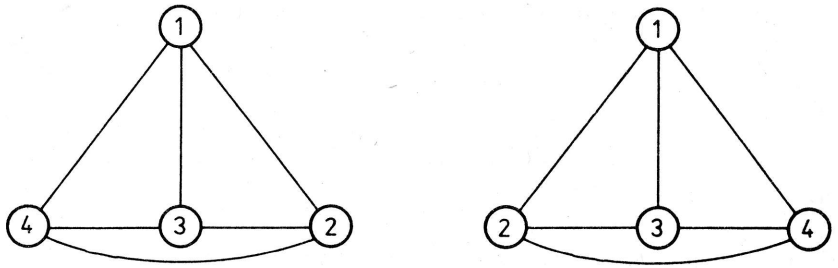
**Veta 1.** Nech je daný rovinný diagram $D$ grafu $G$ a jeho stena $F$. Potom existuje silne ekvivalentný s $D$ diagram $D'$ grafu $G$, v ktorom korešpondujúca stena $F'$ k $F$ bude vonkajšia.

DÔKAZ. Najprv stereografickou projekciou premietneme $D$ na guľovú plochu a potom zvoliac za severný pól bod vo vnútri steny $F$, premietneme diagram z gule na rovinu. Získaný diagram $D'$ má požadovanú vlastnosť. ■

**Veta 2** [Whitney (1933)]. Všetky rovinné diagramy vrcholovo 3-súvislého grafu $G$ sú navzájom ekvivalentné.

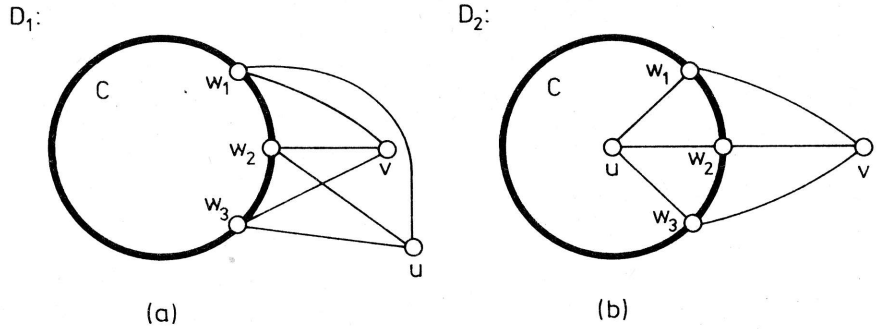







Obr. 1. Ekvivalentné, ale nie silne ekvivalentné diagramy.

DÔKAZ. Nech $D_1$ a $D_2$ sú dva rovinné diagramy grafu $G$. Ukážeme, že sú ekvivalentné. V opačnom prípade existuje cyklus, ktorý je hranicou steny v jednom diagrame a nie je v druhom. Nech $C$ je najkratší taký cyklus a nech je hranicou steny v $D_1$. Potom v $D_2$ existuje vrchol $u$ ležiaci vo vnútri oblasti ohraničenej cyklom $C$ a vrchol $v$ ležiaci v komplementárnej oblasti (obr. 2(b)). Podľa vety 7.A.2 v $G$ existujú tri vnútorne disjunktné $u$—$v$ cesty a vidíme, že každá z nich má vrchol spoločný s $C$. Keďže $u$ a $v$ ležia v $D_1$ vo vonkajšku cyklu $C$, tak tieto tri cesty sa musia pretínať (obr. 2(a)), čo je spor. ■




Obr. 2. K dôkazu vety 2.

**Dôsledok 2(a).** Každé dva rovinné diagramy vrcholovo 3-súvislého grafu sú alebo silne ekvivalentné, alebo také získame, ak jeden z nich preklopíme.

DÔKAZ. Ak diagramy nie sú silne ekvivalentné, tak pri každom vrchole majú hrany v navzájom obrátenom poradí, ako sa ľahko ukáže, lebo sú ekvivalentné. ■

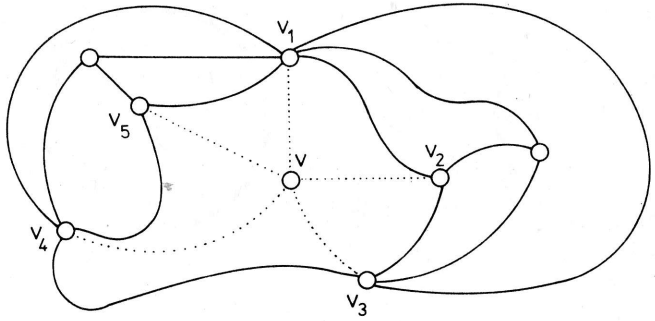
Ešte uvedieme vzťahy planárnych grafov s geometriou. K danému 3-rozmernému konvexnému polyédru $P$ možno priradiť graf $G$ tak, že krajné body (vrcholy) v $P$ budú vrcholmi grafu a hrany v $P$ budú hranami grafu. $G$ sa nazýva *skelet* polyédra $P$.

**Veta 3** [Steinitz (1922)]. Graf $G$ je skelet nejakého konvexného polyédra $\Leftrightarrow$ $G$ je planárny a vrcholovo 3-súvislý.

DÔKAZ tu neuvádzame. Čitateľ ho môže nájsť napr. v [Jucovič (1981)]. Z vety 3 možno ľahko dokázať nasledujúci výsledok, objavený až oveľa neskôr [Wagner (1936)]. Tu však dávame nezávislý dôkaz.

**Veta 4.** Ku každému rovinnému diagramu $D$ existuje silne ekvivalentný rovinný diagram $\hat{D}$, v ktorom je každá hrana reprezentovaná úsečkou.

DÔKAZ. Môžeme predpokladať, že $D$ je rovinná triangulácia (inak $D$ doplníme). Potom stačí nájsť hocijaký úsečkový rovinný diagram toho istého grafu (dôsledok 2(a)). Pre indukciu podľa $n$ predpokladáme, že $n \geq 4$. Podľa dôsledku 10.A.1(d), existuje vrchol $v$ neležiaci na hranici vonkajšej trojuholníkovej steny a taký, že $3 \leq \deg(v) \leq 5$. Potom po vynechaní vrcholu $v$ zostane nám „diera", ktorej hranicou je nejaký $k$-uholník $U = v_1 v_2 \ldots v_k v_1$, kde $3 \leq k \leq 5$ (obr. 3).




Obr. 3. K dôkazu vety 4.

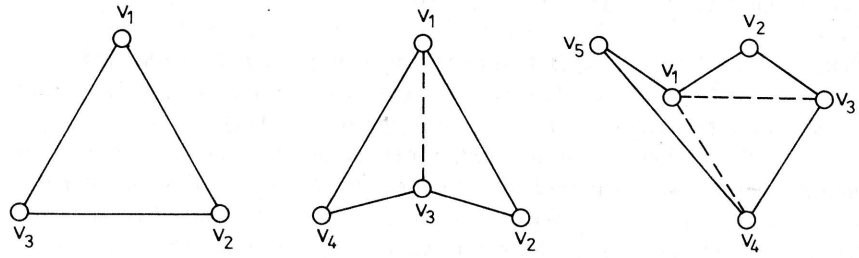
Ak $k = 4$, tak zrejme aspoň jedna z hrán $v_1 v_3$, $v_2 v_4$ nepatrí do $D$ ($D$ je rovinný). Môžeme predpokladať, že $v_1 v_3 \notin E(D)$.

Ak $k = 5$, tak existuje aspoň jeden vrchol $v_i$ ($1 \leq i \leq 5$) taký, že $v_i v_j \notin E(G)$ ak $|j - i| = 2$ (lebo napr. ak $v_1 v_3 \in E(G)$, tak $v_2 v_4$, $v_2 v_5 \notin E(G)$). Môžeme predpokladať, že $v_1$ je taký.





Takto vidíme, že eventuálnym pridaním nových hrán možno $D - v$ zmeniť na rovinnú trianguláciu $D_1$ (ak $k = 3$, nepridávame nič: ak $k = 4$, pridáme $v_1 v_3$: pre $k = 5$ pridáme $v_1 v_3$ a $v_1 v_4$). Podľa indukčného predpokladu, k $D_1$ existuje úsečkový diagram $\hat{D}_1$. V $\hat{D}_1$ je $U$ úsečkový $k$-uholník, ktorý spolu s pridanými hranami dáva konfigurácie podobné tým na obr. 4, kde sú tie „najhoršie".




Obr. 4. Niektoré možnosti pre $U$ v $\hat{D}_1$.

Ak z $\hat{D}_1$ vynecháme hrany, ktoré sme k $D - v$ pridávali, tak do vnútra $k$-uholníka $U$ možno vrátiť vrchol $v$ a úsečky $vv_i$ ($i = 1, 2, \ldots, k$), čím získame požadovaný diagram $\hat{D}$. Pre $k = 3$ možno $v$ umiestniť hocikde vnútri trojuholníka $U$. Pre $k = 4$ možno $v$ umiestniť vnútri úsečky $v_1 v_3$. Konečne pre $k = 5$, vrchol $v$ možno zvoliť v prieniku vnútra päťuholníka $U$ a priamky, ktorá rozdeľuje uhol úsečiek $v_1 v_2$ a $v_1 v_5$ pri vrchole $v_1$ na polovice. ■

Doteraz sme nekládli na rozmiestnenie vrcholov diagramu žiadne obmedzenia. V istom zmysle extrémna situácia vzniká, ak každému vrcholu predpíšeme bod v rovine.

**Veta 5.** Nech $D$ je rovinný diagram s vrcholmi $v_1, v_2, \ldots, v_n$ a nech v rovine je daných $n$ rôznych bodov $u_1, u_2, \ldots, u_n$. Potom existuje diagram $D'$ silne ekvivalentný s $D$ a taký, že pre každé $i = 1, 2, \ldots, n$ bude vrchol $v_i$ umiestnený v bode $u_i$.

DÔKAZ. Zrejme existuje v rovine taká priamka $L$, že žiadne dva vrcholy v $D$ neležia na rovnobežke s $L$. V smere priamky $L$ možno diagram $D$ zmenšiť a zostrojiť rovinný diagram $D_\varepsilon$, ktorý sa zmestí do pásika predpísanej šírky $\varepsilon > 0$. Pásik obsahujúci $D_\varepsilon$ možno ohýbaním, naťahovaním a skracovaním umiestniť v rovine tak, že $v_i$ sa umiestni do bodu $u_i$, $i = 1, 2, \ldots, n$, ak $\varepsilon$ je dostatočne malé. ■

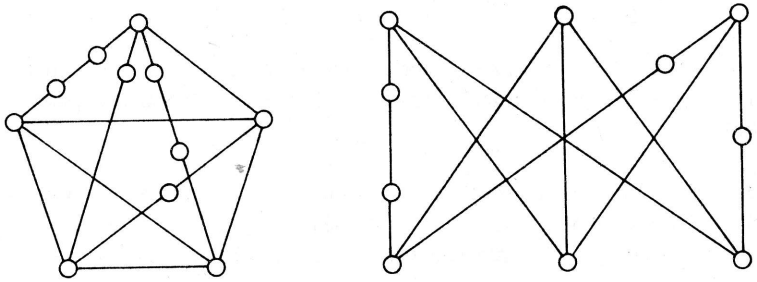
## 10.C
## Testovanie planarity

Hneď vidieť, že graf je planárny práve vtedy, keď každý jeho komponent je planárny. Whitney (1932a) ukázal, že možno ísť ešte ďalej:

**Veta 1.** Graf $G$ je planárny $\Leftrightarrow$ každý blok grafu $G$ je planárny.

DÔKAZ. ($\Rightarrow$) Zrejmé. ($\Leftarrow$) Dokazujeme indukciou podľa počtu blokov. Môžeme predpokladať, že $G$ je súvislý a má viac ako jeden blok, a teda má aj nejaký koncový blok $B$ (pozri kap. 3). Nech $c$ je jediná artikulácia (grafu $G$) ležiaca v $B$. Graf $G-B$, získaný z $G$ vynechaním bloku $B$ (okrem vrcholu $c$), má podľa indukčného predpokladu nejaký rovinný diagram $D_{G-B}$. Blok $B$ je tiež planárny a podľa vety 10.B.1 má aj taký rovinný diagram $D_B$, v ktorom $c$ leží na hranici vonkajšej steny. Ak $D_B$ „vytiahneme pri $c$ do špičky" a vhodne zmenšíme, tak ho zrejme môžeme spojiť s diagramom $D_{G-B}$ vo vrchole $c$ a získať rovinný diagram pre $G$. ■

Hovoríme, že dva pseudografy sú *homeomorfné*, ak sú buď izomorfné, alebo z nich možno získať izomorfné pseudografy vsúvaním nových vrcholov do hrán. Napr. kompletný graf $K_5$ a prvý z grafov na obr. 1 sú homeomorfné, druhý graf na obr. 1 je homeomorfný s kompletným bipartitným grafom $K_{3,3}$.



Obr. 1. Graf homeomorfný s $K_5$ a graf homeomorfný s $K_{3,3}$.

Nasledujúcu úplnú charakterizáciu planárnych grafov dáva Kuratowski (1930). Jej dôkaz neuvádzame: možno ho nájsť napr. v [Harary (1969)].

**Veta 2** (Kuratowski). Graf $G$ je planárny $\Leftrightarrow$ $G$ neobsahuje podgraf homeomorfný s $K_5$ alebo s $K_{3,3}$.





Žiaľ, táto veta neposkytuje polynomiálny algoritmus pre testovanie planarity, lebo nevieme polynomiálne zistiť, či $G$ obsahuje podgraf homeomorfný s $K_5$. (Jej dôkazy však taký algoritmus naznačujú.)

Prv navrhované algoritmy na zisťovanie planarity (napr. [Tutte (1963)], [Lempel, Even, Cederbaum (1967)] sa spočiatku neskúmali z hľadiska výpočtovej zložitosti. Až neskôr sa ukázalo, že sú polynomiálne. Hopcroft a Tarjan (1974) ako prví dali lineárny algoritmus. Vychádzajú z návrhu Auslandera a Partera (1961), postupne pridávať k cyklu cesty, až sa získa vnorenie celého grafu, alebo sa zistí, že je to nemožné. Algoritmus z práce [Lempel et al. (1967)] sa tiež podarilo implementovať na lineárny (pozri [Even (1979)]); tam sa postupne pridávajú vrcholy.

Tu načrtneme algoritmus Hopcrofta a Tarjana. Podrobnú implementáciu nájde čitateľ v ich práci; niektoré opravy tam uvedeného algolovského programu dáva Deo (1976).

### HOPCROFTOV A TARJANOV ALGORITMUS PRE PLANARITU

Vzhľadom na dôsledok 10.A.1(b) a vetu 1 sa obmedzíme na zisťovanie planarity bloku $G$ s $n \geq 3$ vrcholmi a s $m \leq 3n-6$ hranami (identifikáciu blokov možno urobiť v lineárnom čase, pozri 3.B). Uvádzame jednotlivé kroky algoritmu a komentujeme ich.

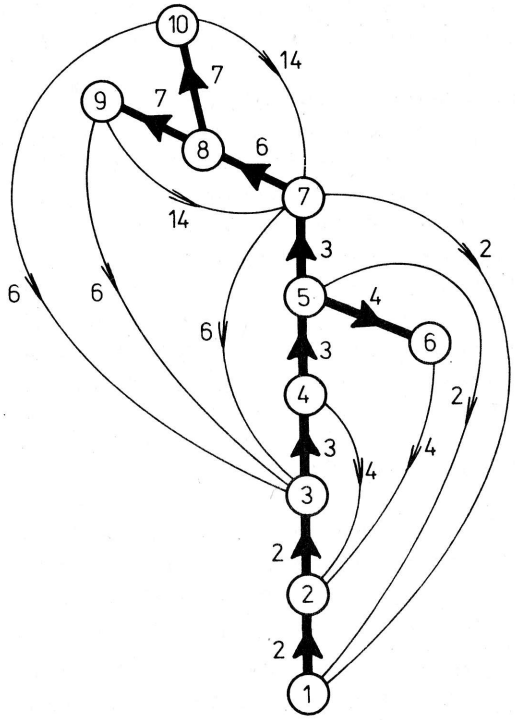
KROK 1: Postupom do hĺbky reprezentujeme $G$ ako palmu (digraf) $P$. Získané poradové čísla objavenia vrcholov budú v ďalšom slúžiť ako ich mená. Základom palmy $P$ je zdrojový strom (kostra) $T$ so zdrojom vo vrchole 1.

Na obr. 2 je uvedený príklad palmy, na ktorom budeme náš postup ilustrovať. Hrubé čiary sú stromové hrany a tenké sú chordy (spätné hrany). Taký sled v $P$, ktorého prvý úsek je v $T$ a zvyšok tvorí jedna chorda, nazývame *sled tvaru* $T+c$.

KROK 2: Pre každý vrchol $v$ zistíme čísla $L_1(v)$ a $L_2(v)$, kde $L_1(v)$ je najmenší vrchol, pre ktorý existuje $v$—$L_1(v)$ sled tvaru $T+c$. $L_2(v)$ je druhý najmenší vrchol nie väčší ako $v$ a taký, že existuje $v$—$L_2(v)$ sled tvaru $T+c$; ak týmto $L_2(v)$ nie je definované, tak kladieme $L_2(v) := v$.

Pretože palma $P$ je blokom, tak zrejme vždy $L_1(v) \leq v$. Tiež $L_1(v) < L_2(v) \leq v$ okrem prípadu, keď $L_1(v) = L_2(v) = v = 1$. Prenechávame čitateľovi navrhnúť lineárny algoritmus pre nájdenie čísel $L_1(v)$ a $L_2(v)$ (postupujte od koncových vrcholov stromu $T$ ku zdroju 1). Pre náš príklad sú tieto čísla na obr. 2.

KROK 3: Pre každý šíp $uv$ vypočítame




| v | $L_1(v)$ | $L_2(v)$ | $V_P^+(v)$ |
|---|---|---|---|
| 1 | 1 | 1 | 2 |
| 2 | 1 | 2 | 3 |
| 3 | 1 | 2 | 4 |
| 4 | 1 | 2 | 5, 2 |
| 5 | 1 | 2 | 1, 7, 6 |
| 6 | 2 | 6 | 2 |
| 7 | 1 | 3 | 1, 3, 8 |
| 8 | 3 | 7 | 9, 10 |
| 9 | 3 | 7 | 3, 7 |
| 10 | 3 | 7 | 3, 7 |

Obr. 2. Príklad ilustrujúci algoritmus pre planaritu.

$$f(u, v) := \begin{cases} 2v, & \text{ak } uv \text{ je chorda,} \\ 2L_1(v), & \text{ak } uv \in T \text{ a } L_2(v) \geq u, \\ 2L_1(v)+1, & \text{ak } uv \in T \text{ a } L_2(v) < u. \end{cases}$$

Potom všetky hrany v $P$ usporiadame do neklesajúcej postupnosti podľa hodnôt funkcie $f$. (To možno urobiť v čase $O(n)$, lebo $m \leq O(n)$ a $2 \leq f(u, v) \leq 2n+1$.) Konečne na základe tejto postupnosti utvoríme usporiadané zoznamy nasledovníkov $V_P^+(v)$: $u$ je v zozname pred $w$, ak $f(v, u) < f(v, w)$.

Pre náš príklad sú hodnoty $f(u, v)$ vyznačené pri hranách na obr. 2, kde sú aj usporiadané zoznamy $V_P^+(v)$.

KROK 4: Pri takto usporiadaných zoznamoch digrafu $P$ aplikujeme postup do





hĺbky (z vrcholu $v$ vždy ideme na prvý voľný vrchol vo $V_P^+(v)$) začínajúc vo vrchole 1. Z generovaného polosledu dávame na výstup vždy maximálny úsek $S_i$ pozostávajúci zo súhlasných hrán. Takto zrejme všetky hrany palmy $P$ rozložíme do sledov $S_1, S_2, \ldots, S_q$, tvaru $T+c$, kde $q$ je počet chord palmy $P$.

V našom príklade dostaneme: $S_1 = (1, 2, 3, 4, 5, 1)$, $S_2 = (5, 7, 1)$, $S_3 = (7, 3)$, $S_4 = (7, 8, 9, 3)$, $S_5 = (9, 7)$, $S_6 = (8, 10, 3)$, $S_7 = (10, 7)$, $S_8 = (5, 6, 2)$, $S_9 = (4, 2)$.

**Lema 1.** Pre generované sledy platí:

(a) Nech $S_j$ je $u$—$v$ sled. Potom $v$ je najmenší vrchol dosiahnuteľný z $u$ po nejakom slede tvaru $T+c$ v digrafe $P$ bez chord použitých v predchádzajúcich sledoch $S_i$. Navyše, v $T$ existuje $v$—$u$ sled.

(b) Prvý sled (t. j. $S_1$) je 1—1 cyklus a každý ďalší sled $S_j$ je nejakou $u$—$v$ cestou, pričom $[V(S_1) \cup \ldots \cup V(S_{j-1})] \cap V(S_j) = \{u, v\}$.

(c) Nech $S_i$ je $u_i$—$v_i$ sled a $S_j$ je $u_j$—$v_j$ sled generovaný neskôr. Ak v $T$ existuje $u_i$—$u_j$ cesta, tak $v_i \leq v_j$.

(d) Nech $S_i$ a $S_j$ sú dva $u$—$v$ sledy, pričom $S_j$ je generovaný neskôr. Nech $w_i$ je druhý vrchol v $S_i$ a $w_j$ druhý vrchol v $S_j$. Potom ak $w_i \neq v$ a $L_2(v_i) < u$, tak $v_j \neq v$ a $L_2(v_j) < u$.

DÔKAZ vyplýva z usporiadania nasledovníkov a prenechávame ho čitateľovi. ■

Nech cyklus $C := S_1 = 1v_1v_2 \ldots v_p1$. Vieme, že $1 < v_1 < v_2 < \ldots < v_p$ a že $v_p1$ je chorda. Ak si palmu $P$ predstavíme ako bodový diagram v 3-rozmernom priestore a taký, že hrany sa nám nepretínajú, tak $P - C$ má niekoľko súvislých častí a (bodový) uzáver každej z nich nazývame *kúsok*. Na identifikáciu kúskov je lepšia nasledujúca ekvivalentná grafová definícia: Kúsok pozostáva alebo z jedinej chordy $v_iw$, kde $1 \leq i \leq p-1$ alebo z hrany $v_iw \in T$, kde $1 \leq i \leq p$ a $w \notin C$, hrán maximálneho podstromu $T_w$ (stromu $T$) so zdrojom $w$ a všetkých chord odchádzajúcich z $T_w$. Z postupu do hĺbky vyplýva, že každý kúsok $K$ pozostáva z niekoľkých po sebe idúcich ciest: $S_j, S_{j+1}, \ldots, S_{j+k}$. Týmto je dané aj usporiadanie kúskov. V našom príklade prvý kúsok pozostáva z ciest $S_2, S_3, \ldots, S_7$, druhý z $S_8$ a tretí z $S_9$.

KROK 5: Postupne vytvárame rovinný diagram pre $P$. Najprv zakreslíme cyklus $C$, a to tak, že cesta $1v_1v_2 \ldots v_p$ zodpovedá zvislej úsečke $R$ vedenej zdola nahor a chorda $v_p1$ je realizovaná oblúkom, ktorý leží v pravej polrovine od $R$. V ďalšom postupe zakreslíme aj jednotlivé kúsky opierajúc sa o ďalej uvedené lemy, resp. zistíme, že to nie je možné. STOP. ●

Je zrejmé, že ak chceme získať rovinný diagram, tak každý kúsok musí ležať alebo celý vo vnútri cyklu $C$ alebo celý vo vonkajšku. V prvom prípade budeme jednoducho hovoriť o umiestnení kúska vpravo od $C$ a v druhom vľavo od $C$.

**Lema 2.** Nech $K$ je kúsok. Predpokladajme, že všetky predchádzajúce kúsky sa nám už podarilo rovinne umiestniť. Nech $S$ je prvá cesta (tvaru $T+c$) v $K$. Keďže $S$ začína a končí na $C$, tak $S$ je nejaká $v_i$—$v_j$ cesta. Potom $S$ možno umiestniť vľavo (vpravo) od $C \Leftrightarrow$ pre žiadnu už vľavo (vpravo) umiestnenú chordu $xv_k$ neplatí $v_j < v_k < v_i$.

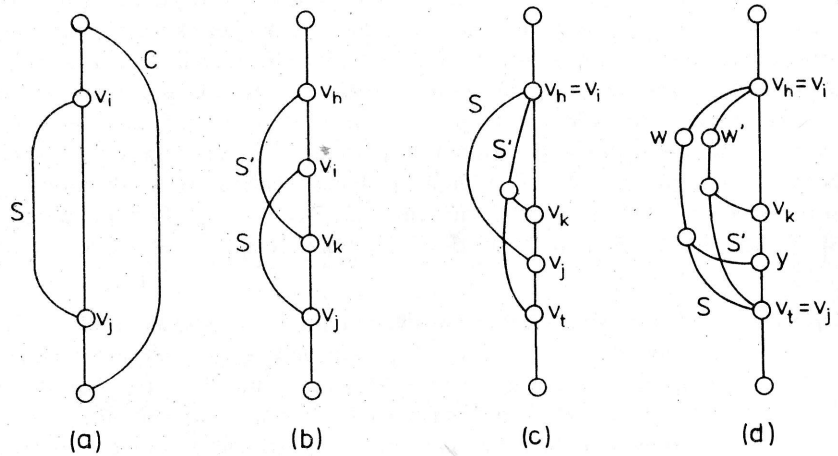
DÔKAZ. Budeme dokazovať len prípad „vľavo". „Vpravo" sa dokazuje rovnako.

($\Leftarrow$) Z postupu do hĺbky vyplýva, že žiadna predchádzajúca cesta tvaru $T+c$ nezačína medzi $v_j$ a $v_i$. Preto ak žiadna chorda vľavo medzi týmito vrcholmi nekončí, tak zrejme $S$ možno vľavo umiestniť (obr. 3(a)).

($\Rightarrow$) Nech nejaká chorda $xv_k$ umiestnená vľavo spĺňa vzťah: $v_j < v_k < v_i$. Ukážeme, že $S$ nemožno vľavo umiestniť. Chorda $xv_k$ je súčasťou nejakého kúska $K'$, ktorý má zdroj v nejakom vrchole $v_h \in C$. Z postupu do hĺbky vyplýva, že $v_h \geq v_i$, lebo $K'$ bol prv generovaný ako $K$. Uvážime niekoľko možností:

(1) $v_h > v_i$ (obr. 3(b)). Zrejme $S$ nemožno umiestniť vľavo.

(2) $v_h = v_i$. Nech $S'$ je prvá cesta v kúsku $K'$: je to nejaká $v_h$—$v_t$ cesta. Podľa lemy 1(c) máme $v_t \leq v_j$. Uvážime 2 prípady.




Obr. 3. K dôkazu lemy 2.





(2A) $v_t < v_j$ (obr. 3(c)). $S$ nemožno umiestniť vľavo, lebo z cesty $S'$ vedie odbočka do $v_k$.

(2B) $v_t = v_j$ (obr. 3(d)). Z cesty $S'$ je niekde odbočka do $v_k$, takže druhý vrchol cesty $S'$ je nejaký vrchol $w' \neq v_t$ a $L_2(w') \leq v_k < v_h$. Potom podľa lemy 1(d) druhý vrchol (nech je to) $w$ cesty $S$ je rôzny od $v_j$ a $L_2(w) < v_i$. Preto z $S$ ide odbočka, ktorá končí v nejakom vrchole $y \in C$, kde $v_j < y < v_i$. Ani teraz nemožno $S$ umiestniť. ■

Lemu 2 možno použiť na umiestnenie prvej cesty kúska vľavo, resp. vpravo. Ak to nejde, tak niektoré kúsky zľava tomu bránia, a preto ich presunieme vpravo. Potom však bude možno treba presúvať niektoré kúsky opačným smerom atď. Čitateľa odkazujeme na pôvodný článok Hopcrofta a Tarjana, kde je ukázaná efektívna manipulácia s dátami pre takéto presúvanie kúskov po skupinách.

Výpočtovo zložitejší, ale ideovo jednoduchší, je nasledujúci spôsob. Postupne ako spracovávame kúsky $K^1$, $K^2$, ... generujeme pomocný graf $H$, ktorého vrcholmi sú $K^1$, $K^2$, ..., pričom dva vrcholy $K^g$ a $K^h$ ($g < h$) spojíme hranou, ak existuje v $K^g$ chorda $xv_k$ taká, že pre prvú cestu v $K^h$ (vždy má tvar) $v_i$—$v_j$ platí: $v_j < v_k < v_i$. To podľa lemy 2 znamená, že kúsky $K^g$ a $K^h$ nemôžu byť na tej istej strane cyklu $C$. Potom stačí zistiť, či okamžitý graf $H$ je bipartitný (vtedy možno prvú cestu z $K^h$ umiestniť, inak je palma $P$ neplanárna).

Zostala nám otvorená otázka umiestnenia ostatných ciest kúska. To však rieši rekurentne nasledujúca lema, ktorej dôkaz prenechávame čitateľovi.

**Lema 3.** Nech $K$ je kúsok. Predpokladajme, že všetky predchádzajúce kúsky a prvú cestu z $K$ sme už rovinne umiestnili. Potom zvyšok (t. j. ostatné cesty) kúska $K$ možno tiež umiestniť $\Leftrightarrow$ digraf $\hat{G} := C \cup K$ je planárny.

Na zistenie planarity digrafu $\hat{G} := C \cup K$ sa použije algoritmus rekurentne. Za základný cyklus v $\hat{G}$ zvolíme cyklus $\hat{C}$ pozostávajúci z $v_i$—$v_j$ cesty $S$ a (jedinej) $v_j$—$v_i$ cesty v strome $T$. (V našom príklade: ak $K$ pozostáva z ciest $S_2$, $S_3$, ..., $S_7$, tak $\hat{C} = (5, 7, 1, 2, 3, 4, 5)$, lebo $S = S_2$.) Hneď vidieť, že $\hat{G} - \hat{C}$ bude pozostávať z aspoň dvoch kúskov, pričom jeden z nich je časťou cyklu $C$, a preto je cestou, ktorú ľahko identifikujeme. Teda tento rekurentný proces sa nezacyklí. Naviac netreba robiť rozklad na cesty, lebo pôvodné cesty pre ostatné kúsky vyhovujú, ako sa ľahko overí. ●

## 10.Z
## Cvičenia a poznámky

1. Dokážte, že ak v rovinnom diagrame grafu $G$ je hranica každej vnútornej steny cyklom párnej dĺžky, tak $G$ je bipartitný.
2. Konvexný mnohosten sa nazýva pravidelný, ak jeho steny sú zhodné pravidelné mnohouholníky a pri každom vrchole je rovnaký počet hrán (napr. kocka). Dokážte, že existuje práve päť (až na podobnosť) takýchto mnohostenov (majú 4, 6, 8, 12 a 20 stien). Tieto mnohosteny poznali už starí Gréci a sú známe ako *platónske telesá*.
3. K danému rovinnému diagramu súvislého pseudografu možno zostrojiť tzv. *duálny diagram* takto: Do každej steny umiestnime jeden vrchol a dva takéto vrcholy spojíme toľkými hranami, koľko hrán majú spoločných príslušné steny (ich hranice) a k vrcholu dáme toľko slučiek, koľko príslušná stena obsahuje mostov. Takto duálny diagram má práve toľko hrán ako pôvodný. (i) Nakreslite duálne diagramy k platónskym telesám. (ii) Dokážte, že duál k duálu je pôvodný diagram.
4. Graf sa nazýva *vonkajškovo planárny* (VP), ak má taký rovinný diagram, v ktorom všetky vrcholy ležia na hranici vonkajšej steny. (i) Nech $G$ je graf a nech graf $\hat{G}$ vznikne z $G$ pridaním nového vrcholu, ktorý sa spojí hranami so všetkými pôvodnými vrcholmi. Dokážte, že $G$ je VP $\Leftrightarrow$ $\hat{G}$ je planárny. (ii) Využite (i) na dôkaz existencie lineárneho algoritmu pre zisťovanie vonkajškovej planarity (pozri tiež [Mitchell (1979)]). (iii) Dokážte, že VP graf má nanajvýš $2n-3$ hrán. (iv) Vyslovte analógiu dôsledku 10.A.1(d) pre VP grafy.
5. (i) Dokážte lemy 10.C.1 a 10.C.3. (ii) Algoritmom z 10.C. ukážte, že tzv. *Petersenov graf* (t. j. kubický graf s $n = 10$, priemeru 2 a s najmenším cyklom dĺžky 5) nie je planárny.
6. Pridajme ku guli $g$ „úch" (alebo prevŕtajme $g$ tunelov). Povrch takéhoto telesa sa nazýva (orientovateľná) *plocha rodu* $g$. (i) Ukážte, že kompletný graf $K_7$ je vnoriteľný do plochy rodu 1. (ii) Ukážte, že ak v diagrame grafu vnoreného do plochy rodu $g$ je každá stena jednoducho súvislá oblasť, tak platí zovšeobecnenie Eulerovho vzorca: $n-m+r = 2-2g$.

   Poznamenajme, že zistiť najmenšie $g$, pre ktoré sa daný graf dá vnoriť do plochy rodu $g$, je otvorený problém a známe sú len čiastkové výsledky (pozri napr. [Behzad et al. (1979)]. Filotti, Miller a Reif (1979) však dali $O(n^{O(g)})$ algoritmus na zistenie toho, či stačí rod $g$.
7. Pre grafy boli navrhnuté ďalšie miery neplanárnosti:
   (i) *Minimálny počet priesečníkov* hrán v diagrame (v jednom priesečníku pripúšťame pretínanie iba dvoch hrán): dokážte, že toto číslo je 1 pre kompletný graf $K_5$ a 3 pre $K_6$.
   (ii) *Minimálny počet planárnych grafov,* na ktoré možno daný graf hranovo rozložiť. Ohraničte toto číslo zdola pre kompletné grafy.

   Prehľad dosiahnutých výsledkov pozri napr. v [Behzad et al. (1979)]; z novších príspevkov odporúčame [Širáň (1981)]. Algoritmicky neboli dosiahnuté významnejšie výsledky.
8. (i) Navrhnite nejakú heuristiku na grafoch pre hľadanie planárneho podgrafu s maximálnym počtom hrán. (ii) Zapíšte túto úlohu ako 0—1 úlohu lineárneho programovania.

   Tento problém je NP-ťažký [Liu a Geldmacher (1978)].

# 11
# IZOMORFIZMUS

Problém zisťovania izomorfizmu grafov (pseudomigrafov) siaha do počiatkov teórie grafov. Jeho význam však stúpol v súvislosti s aplikáciami, najmä pri porovnávaní chemických zlúčenín (resp. ich katalogizácii), pri rozpoznávaní obrazcov a inde. Nie je dokázané, že tento problém je NP-ťažký, ale existujú výsledky o polynomiálnej ekvivalencii s inými podobnými problémami v algebraických štruktúrach. Na druhej strane polynomiálny algoritmus tiež nie je známy. Pre špeciálne triedy grafov sa však úlohu podarilo zvládnuť. Tu si všimneme izomorfizmus stromov a planárnych grafov a tiež problém izomorfného podstromu. Odkazy na ďalšie výsledky sú najmä v časti 11.Z.

## 11.A
## Izomorfizmus stromov

Týmto problémom sa zaoberali viacerí matematici a iní odborníci. Navrhuje sa najprv pretransformovať stromy na nejaké „kanonické" tvary, ktoré by boli rovnaké iba pre izomorfné stromy. Tu uvádzame základnú myšlienku z práce [Hopcroft a Tarjan (1972)], kde je istý Edmondsov algoritmus (pozri [Busacker a Saaty (1965)] trochu rozšírený tak, aby bolo možné porovnávať označkované stromy.

Predpokladá sa, že ku každému vrcholu $u$ daného neorientovaného stromu $T$ je priradená značka $L(u)$, čo je nejaká multimnožina prirodzených čísel (napr. {2, 3, 3, 3, 7, 7}). Ak sú všetky značky rovnaké (napr. prázdne množiny), tak vlastne máme prípad neoznačkovaného stromu.

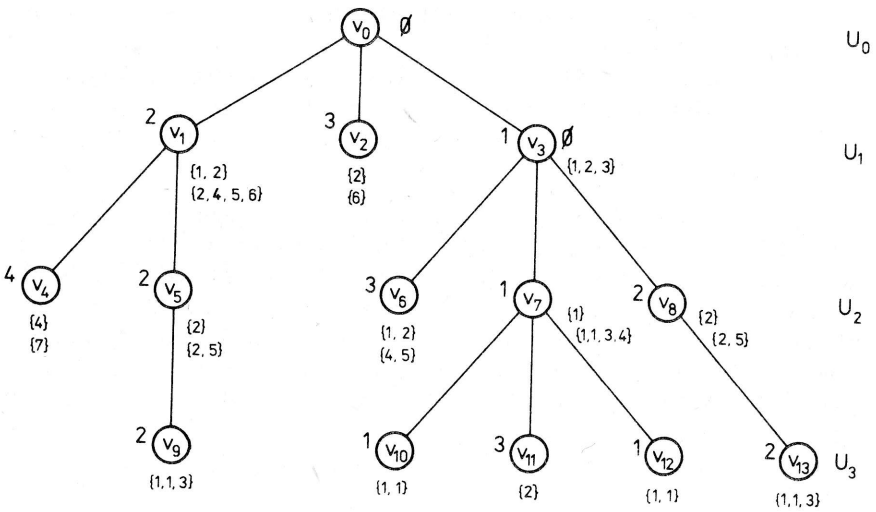
Dva označkované stromy $T_1$, $T_2$ pokladáme za izomorfné, ak existuje (obyčajný) izomorfizmus $\varphi: T_1 \to T_2$, ktorý zachováva značky (t. j. ak $\varphi(u) = v$, tak aj $L(u) = L(v)$).

Uvedieme metódu zostrojenia *kanonického stromu* k danému označkovanému stromu $T$. Postup ilustrujeme na strome z obr. 1.

KROK 1: Nájdeme centrum stromu $T$. Ak pozostáva z jediného vrcholu $v_0$, prejdeme na ďalší krok. Inak centrum pozostáva z dvoch susediacich vrcholov $v_1$, $v_2$. Potom do hrany $v_1v_2$ vsunieme nový vrchol $v_0$ (so značkou ∅) a máme strom, ktorý má jediný centrálny vrchol $v_0$.








Obr. 1. Zostrojenie kanonického tvaru pre strom.

KROK 2: Urobíme distančný rozklad množiny $V(T)$ z vrcholu $v_0$. (Nech úrovne stromu $T$ sú $U_0$, $U_1$, ..., $U_s$.) Strom $T$ môžeme uvažovať ako orientovaný so zdrojom $v_0$ a hovoriť o predchodcoch a nasledovníkoch.

KROK 3: Značky vrcholov úrovne $U_s$ *usporiadame lexikograficky* od najmenšej po najväčšiu. [Najprv čísla vo vnútri každej značky $L(v)$ usporiadame do neklesajúcej postupnosti. Potom dve rôzne značky $L(u)$, $L(v)$ porovnávame tak, že z nich vynecháme začiatočný spoločný úsek maximálnej dĺžky a zisťujeme, ktorý zvyšok je prázdny, resp. má menší prvý prvok; tú zo značiek prehlásime za lexikograficky menšiu (označenie: $\prec$).] Pre náš príklad máme: $L(v_{10}) = L(v_{12}) \prec L(v_9) = L(v_{13}) \prec L(v_{11})$. V súlade s usporiadaním značiek, vrcholy v úrovni $U_s$ dostanú poradové čísla 1, 2, ..., $p_s$, kde $p_s$ je počet rôznych značiek (v príklade je $p_3 = 3$).

KROK 4: Ak už máme usporiadané vrcholy v úrovni $U_k$, $(2 \leq k \leq s)$, tak usporiadame vrcholy $u \in U_{k-1}$ takto: Najprv zväčšíme všetky prvky vo vnútri značiek $(k-1)$-ej úrovne o hodnotu $p_k$, čím získame značky $L'(u)$. Potom každému vrcholu $u \in U_{k-1}$ priradíme značku $L''(u)$, ktorej prvé zložky sú usporiadané poradové čísla nasledovníkov vrcholu $u$ a ďalšie zložky sú prvky značky $L'(u)$. Takto získané značky

sú vo vnútri usporiadané podľa veľkosti. (V našom príklade je: $L''(v_4) = \{7\}$, $L''(v_5) = \{2, 5\}$, $L''(v_6) = \{4, 5\}$, $L''(v_7) = \{1, 1, 3, 4\}$, $L''(v_8) = \{2, 5\}$.) Značky $L''(u)$ usporiadame lexikograficky, a tým získame poradové čísla vrcholov $u \in U_{k-1}$: 1, 2, ..., $p_{k-1}$ (u nás je $p_2 = 4$). Krok 4 opakujeme, kým neusporiadame vrcholy každej úrovne: STOP. ●

Usporiadaním vrcholov v jednotlivých úrovniach je kanonický tvar stromu zostrojený (porovnaj obr. 1). Na tomto základe možno potom strom zakódovať takto: Každému vrcholu $x$ prislúcha okrúhla zátvorka a všetci nasledovníci vrcholu $x$ sú v tejto zátvorke usporiadaní podľa veľkosti poradových čísel pre ich úroveň. Pred zátvorkou okrem toho môže byť symbol vrcholu, ktorý však možno vynechať. Do zátvorky na začiatok (ešte pred nasledovníkov) dávame značku vrcholu. Tento zátvorkový zápis kanonického stromu pre strom z obr. 1 je takýto (ak nepíšeme symboly vrcholov):
(Ø(Ø({1}({1, 1}) ({1, 1}) ({2})) ({2}({1, 1, 3})) ({1, 2})) ({2}) ({1, 2}({2}({1, 1, 3})) ({4}))).

Takéto zápisy stromov sa hodia pre katalogizáciu a samozrejme pre zisťovanie izomorfizmu označkovaných stromov. Totiž dva izomorfné označkované stromy musia mať rovnaké centrum a platí nasledujúce tvrdenie, dôkaz ktorého je zrejmý a prenechávame ho čitateľovi.

**Veta 1.** Pre ľubovoľné označkované stromy $T_1$, $T_2$ s rovnakým centrom nasledujúce tvrdenia sú ekvivalentné:
(a) $T_1$ a $T_2$ sú izomorfné.
(b) Kanonické stromy pre $T_1$ a $T_2$ sú rovnaké.
(c) Zátvorkové zápisy kanonických stromov pre $T_1$ a $T_2$ sú rovnaké.

Poznamenajme, že ak každá značka má len prvky z množiny $\{1, 2, ..., n\}$ a sumárna dĺžka zápisu všetkých značiek je $q$, tak pri vhodnej implementácii (triedenie je tu dôležité) už uvedená metóda dáva $O(n+q)$ algoritmus, ako ukazujú Hopcroft a Tarjan (1972). Takto máme $O(n)$ algoritmus na zisťovanie izomorfizmu dvoch $n$-vrcholových neorientovaných stromov. Analogicky možno postupovať aj v prípade, že stromy obsahujú aj šípy.

## 11.B
## Izomorfný podstrom

Tu uvažujeme nasledujúce zovšeobecnenie úlohy o izomorfizme dvoch stromov.

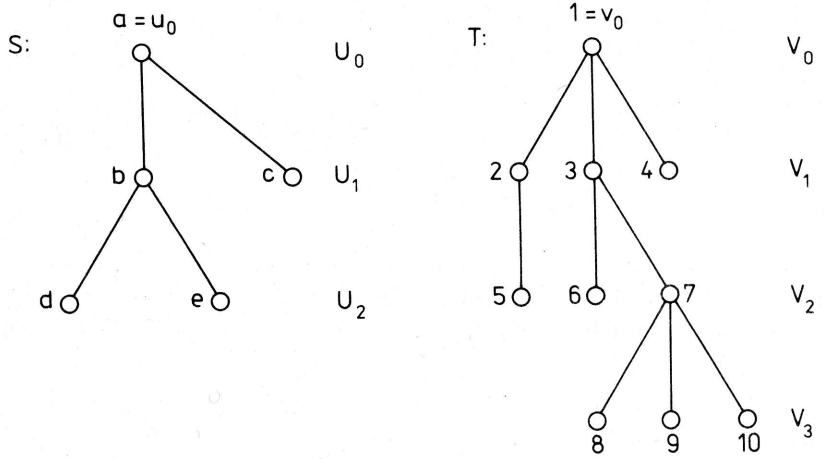




Daný je orientovaný strom $T$ so zdrojom $v_0$ a orientovaný strom $S$ so zdrojom $u_0$. Treba zistiť, či $T$ obsahuje podstrom so zdrojom $v_0$ izomorfný so stromom $S$. Prípadne zdroj podstromu nemusí byť predpísaný. V oboch prípadoch hovoríme, že $S$ možno izomorfne vnoriť do $T$.

Ďalej uvádzame metódu, ktorá rieši takéto úlohy. Presnejšie, poskytuje množinu $W(u_0)$ všetkých takých vrcholov stromu $T$, ktoré môžu slúžiť za zdroj hľadaného podstromu. Postup sa opiera o párenia v bipartitných grafoch. Túto myšlienku naznačujú Hopcroft a Karp (1973) a podrobne rozpracovávajú Matula (1978) a Stadel (1978).

Na obr. 1 možno ľahko overiť, že $W(a) = \{1, 3\}$.




Obr. 1. $S$ možno izomorfne vnoriť do $T$: $W(a) = \{1, 3\}$ (šípky nie sú vyznačené).

Nech pre daný vrchol $x \in V(S)$ symbol $S_x$ označuje maximálny podstrom stromu $S$ so zdrojom $x$; analogicky sa definuje $T_x$. Našou úlohou je nájsť množinu $W(u_0) = \{j \in V(T) |$ existuje izomorfné vnorenie $\varphi: S_{u_0} \to T_j$, kde $\varphi(u_0) = j\}$. Nasledujúca metóda umožňuje nejaké vnorenie aj nájsť.

Predpokladajme, že pre $x \in V(S)$ a $i \in V(T)$ máme zistiť, či $i \in W(x)$. K tomu zostrojíme bipartitný graf $G_{x,i}$, kde jedna partia vrcholov je $V_S^+(x)$, druhá $V_T^+(i)$ a $E(G_{x,i}) = \{yk | y \in V_S^+(x), k \in V_T^+(i) \cap W(y)\}$. Zrejme platí:

**Lema 1.** Vrchol $i \in W(x) \Leftrightarrow$ v grafe $G_{x,i}$ existuje párenie pokrývajúce všetky vrcholy partie $V_S^+(x)$.

Poznamenajme, že pre riešenie úlohy určiť $W(u_0)$ nie je potrebné zisťovať, či

napr. pre obr. 1 vrchol $1 \in W(e)$. Vrchol $e$ sa tiež nemôže zobraziť (pri izomorfnom vnorení $S$ do $T$) na žiadny z vrcholov 2, 3 a 4, lebo $d_S(a, e) = 2$ a napr. $d_T(1, 4) < 2$. Ďalej uvedený algoritmus preto uvažuje namiesto množín $W(x)$ množiny $\bar{W}(x)$, ktoré sú bez takýchto prvkov. Jeho správnosť je zrejmá.

**Algoritmus.**

KROK 0: Utvoríme distančný rozklad $V(S) = U_0 \cup U_1 \cup \ldots \cup U_p$ z vrcholu $u_0$ a distančný rozklad $V(T) = V_0 \cup V_1 \cup \ldots \cup V_q$ z vrcholu $v_0$.

KROK 1: Pre každý vrchol $x \in U_p$ položíme $\bar{W}(x) := V_p \cup V_{p+1} \ldots \cup V_q$ a pre každé $x \in V(S) - V_p$ kladieme $\bar{W}(x) := \emptyset$.

KROK 2: Pre každé $k = p-1, p-2, \ldots, 0$ urobíme:
- Pre každé $x \in U_k$ urobíme:
  - Pre každé $i \in V_k \cup V_{k+1} \cup \ldots \cup V_{q-p+k}$ urobíme:
    - Zostrojíme graf $G_{x,i}$ a zistíme, či v ňom existuje párenie pokrývajúce všetky vrcholy $V_S^+(x)$. Ak existuje, tak $\bar{W}(x) := \bar{W}(x) \cup \{i\}$, inak sa $\bar{W}(x)$ nemení.

$W(u_0) := \bar{W}(u_0)$: STOP. ●

Uvedený algoritmus je zrejme polynomiálny, podrobne ho však neanalyzujeme. Matula (1978) a Stadel (1978) uvádzajú algoritmy zložitosti $O(n_T^{3/2} n_S)$. Pripomeňme (kap. 2), že vo všeobecnosti je problém izomorfného podgrafu NP-ťažký (pozri tiež (11.Z.4)).

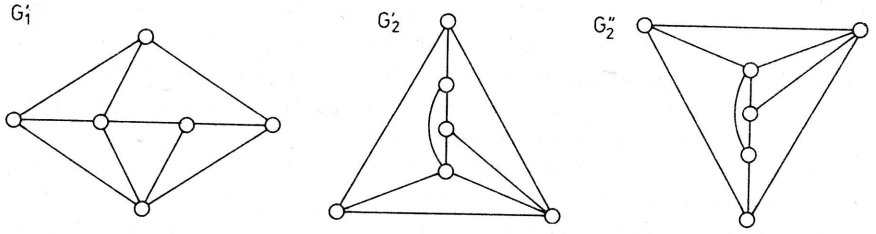
## 11.C
## Izomorfizmus 3-súvislých planárnych grafov

Ak chceme porovnávať diagramy rôznych, ale izomorfných grafov, tak musíme pojmy „ekvivalentné", a „silne ekvivalentné" diagramy rozšíriť. Hovoríme, že rovinné diagramy $D_1$ pseudomigrafu $G_1$ a $D_2$ pseudomigrafu $G_2$ sú *izomorfné*, ak existuje izomorfizmus $\varphi: G_1 \to G_2$, ktorý navyše zobrazuje hranicu steny v $D_1$ na hranicu steny v $D_2$; $D_1$ a $D_2$ sú *silne izomorfné*, ak naviac $\varphi$ zachováva v diagrame usporiadanie hrán incidujúcich s vrcholom pri jeho obiehaní v pravotočivom smere. (Ak $D_1 = D_2$, tak izomorfizmus sa nazýva *automorfizmus.*)

Podľa Whitneyho vety, resp. jej dôsledku 10.B.2(a), každé dva rovinné diagramy vrcholovo 3-súvislého grafu sú alebo silne izomorfné, alebo takéto získame, ak jeden z nich preklopíme. Teda na zisťovanie izomorfizmu dvoch planárnych vrcholovo 3-súvislých grafov $G_1$ a $G_2$ stačí: (1) nájsť ich rovinné diagramy $G_1'$ a $G_2'$, (2)







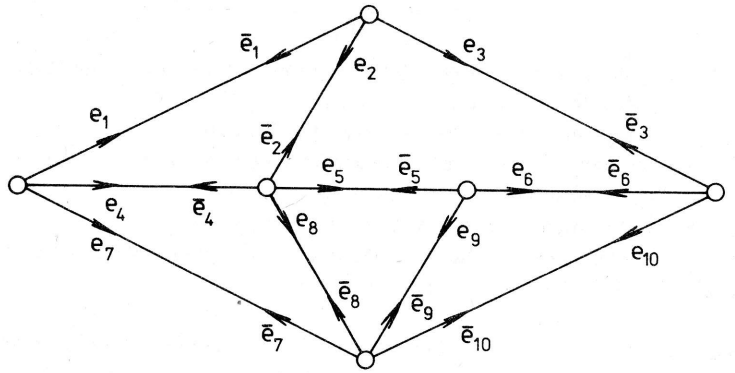

Obr. 1. Diagramy $G_1'$ a $G_2''$ sú silne izomorfné.

preklopením $G_2'$ získať diagram $G_2''$, (3) zistiť, či $G_1'$ a $G_2'$ alebo $G_1'$ a $G_2''$ sú silne izomorfné (ako neoznačkované diagramy). Je zrejmé, že grafy $G_1$ a $G_2$ sú izomorfné práve vtedy, keď získame v (3) kladnú odpoveď (porovnaj obr. 1).

Na základe tejto myšlienky dáva Weinberg (1966) $O(n^2)$ algoritmus. Tu vychádzame z prác [Hopcroft, Tarjan (1972, 1973b)], kde je daný $O(n \log n)$ algoritmus. Hopcroft a Wong (1974) oznamujú dokonca $O(n)$ algoritmus, avšak (ako sami uvádzajú) tento algoritmus má skôr teoretický význam a neodporúčajú ho pre praktické výpočty.

Diagramy $G_1'$, $G_2'$ a $G_2''$ môžeme považovať za diagram jedného grafu s tromi komponentmi a hľadať silné automorfizmy tohto diagramu. Kladnú odpoveď v (3) získame práve vtedy, keď existuje silný automorfizmus, pri ktorom sa nejaká hrana z $G_1'$ zobrazí na hranu z $G_2'$ alebo $G_2''$.

Hopcroft a Tarjan uvažujú takúto všeobecnú úlohu: Daný je rovinný diagram grafu, ktorého každý komponent je vrcholovo 3-súvislý. Každému rebru $e = uv$ priradíme dva šípy: $e = uv$ a $\bar{e} = vu$, ktoré však znázorňujeme na diagrame jedinou čiarou s dvoma opačnými šípkami (pozri obr. 2). Úlohou je rozdeliť všetky




Obr. 2. Ošípkovaný diagram grafu.

šípy do tried tak, aby dva šípy $f$, $g$ patrili do tej istej triedy práve vtedy, keď existuje silný automorfizmus takto ošípkovaného diagramu, pri ktorom sa $f$ zobrazí na $g$. (Poznamenajme, že $f$ je nejaký šíp $e$, alebo $\bar{e}$: kladieme $\bar{\bar{e}} = e$.)

Najprv sa šípy rozdelia do základných tried tak, že dve hrany sú v tej istej triede práve vtedy, keď majú rovnaké nasledujúce parametre:

(i) dĺžka hranice steny vpravo od šípa,
(ii) dĺžka hranice steny vľavo od šípa,
(iii) stupeň vrcholu pri začiatku šípa a
(iv) stupeň vrcholu pri konci šípa.

(Stupne počítame v neorientovanom diagrame.)

Tieto základné triedy sa potom ešte rozdelia na menšie. K tomu však potrebujeme zaviesť niekoľko pojmov.

Sled v ošípkovanom diagrame daný postupnosťou šípov $e_1, e_2, \ldots, e_k$ sa nazýva *hraničný*, ak pre každé $i = 1, 2, \ldots, k-1$ existuje práve jedna stena tak, že šípy $e_i$ a $e_{i+1}$ ležia na hranici tejto steny. Hraničný sled je determinovaný hranou $e_1$ a postupnosťou $x$ pozostávajúcou z $k-1$ symbolov L alebo R, ktoré označujú, na hranici ktorej z dvoch stien susediacich cez $e_i$ leží šíp $e_{i+1}$: ak na hranici ľavej, tak to vyjadríme symbolom L, inak dávame symbol R. Napr. na obr. 2 sledy $e_7\bar{e}_9$ a $e_7\bar{e}_7$ nie sú hraničné, ale sled $e_7\bar{e}_8e_5e_6\bar{e}_3\bar{e}_1e_7\bar{e}_{10}$ je hraničný a možno ho zadať takto: $e_7(\mathrm{LRLLRRR})\bar{e}_{10}$. Symbol $e_p(—)e_q$ označuje hraničný sled začínajúci šípom $e_p$ a končiaci šípom $e_q$: takýto sled existuje, ak existuje postupnosť $x$ symbolov L alebo R tak, že $e_p(x)e_q$ je hraničný sled.

Dva šípy $e$ a $f$ sa nazývajú *rozlíšiteľné*, ak existujú šípy $e_1$ a $f_1$ z rôznych základných tried a postupnosť $x$ tak, že $e(x)e_1$ a $f(x)f_1$ sú hraničné sledy; inak sa nazývajú *nerozlíšiteľné*. Naším najbližším cieľom je rozdeliť všetky šípy do tried nerozlíšiteľných hrán. Na to budeme potrebovať niektoré vlastnosti zavedených pojmov.

**Lema 1.** Nech $e$, $f$, $g$, $h$ sú šípy v ošípkovanom rovinnom diagrame vrcholovo 3-súvislého grafu. Potom platí:

(i) Ak existuje $e(—)f$, tak aj $f(—)e$ a $\bar{e}(—)\bar{f}$ existujú.
(ii) Ak $g$ a $h$ sú rozlíšiteľné a existuje $x$ tak, že $e(x)g$ a $f(x)h$ sú hraničné sledy, tak aj $e$ a $f$ sú rozlíšiteľné.
(iii) Ak $e$ a $f$ sú rozlíšiteľné, tak aj $\bar{e}$ a $\bar{f}$ sú rozlíšiteľné.
(iv) Ak $e$ a $f$ sú nerozlíšiteľné a existuje $x$ tak, že $e(x)g$ a $f(x)h$ sú hraničné sledy, tak aj $g$ a $h$, a tiež $\bar{g}$ a $\bar{h}$ sú nerozlíšiteľné.

DÔKAZ. (i) Nech $ee_1 \ldots e_kf$ je hraničný sled. Ak existuje $e_k(\mathrm{R})f$, tak existuje $f(\mathrm{RR}\ldots\mathrm{R})e_k$; ak existuje $e_k(\mathrm{L})f$, tak existuje $f(\mathrm{LL}\ldots\mathrm{L})e_k$. Existencia sledu $f(—)e$





vyplýva indukciou. Ak existuje $f(x)e$, tak aj $\bar{e}(x')\bar{f}$, kde $x'$ je postupnosť získaná obrátením postupnosti $x$ a vymenením symbolov L a R.

(ii) Existuje postupnosť $y$ a šípy $g_1$, $h_1$ z rozličných základných tried tak, že $g(y)g_1$ a $h(y)h_1$ sú hraničné sledy. Potom sledy $e(xy)g_1$ a $f(xy)h_1$ dávajú rozlíšiteľnosť $e$ a $f$.

(iii) Nech $g$ a $h$ sú z rôznych základných tried a také, že existujú $e(x)g$, $f(x)h$, kde $x$ má minimálnu dĺžku. Okrem $g$ a $h$, ľubovoľný šíp v prvom slede a zodpovedajúci šíp v druhom slede majú vpravo steny s rovnako dlhou hranicou a analogicky vľavo (inak by $x$ nemalo minimálnu dĺžku). Potom konštrukcia v dôkaze pre (i) dáva sledy $\bar{e}(y)\bar{g}$ a $\bar{f}(y)\bar{h}$ pre nejaké $y$. Keďže šípy $\bar{g}$ a $\bar{h}$ sú z rôznych základných tried, dôkaz je hotový.

(iv) Pre $g$ a $h$ to vyplýva z (ii) a pre $\bar{g}$ a $\bar{h}$ z (iii). ■

## ROZDELENIE ŠÍPOV DO TRIED

Ak existuje jediná základná trieda, tak graf je skeletom niektorého z piatich platónskych telies (pozri 10.Z.2), čo sa ľahko identifikuje. Inak existujú aspoň dve základné triedy a z nich možno postupne vyrobiť menšie a menšie triedy, až získame triedy nerozlíšiteľných šípov. K tomu Hopcroft a Tarjan dávajú tento algoritmus:

**Algoritmus**

KROK 0: Nech $B_1, B_2, \ldots, B_k$ sú všetky základné triedy ($k \geq 2$). Každej z nich priradíme dva symboly: R a L. V ďalšom budeme tieto triedy štiepiť a nazývať jednoducho triedy.

KROK 1: Ak je každá trieda bez symbolu, tak STOP.
Inak zvolíme nejakú triedu $B_i$, ktorá má aspoň jeden symbol a zvolíme niektorý z nich. Ak sme zvolili R, tak utvoríme $M_i := \{f \mid \text{existuje } e(\mathrm{R})f,\ e \in B_i\}$ a symbol R pri $B_i$ zrušíme. Ak sme zvolili L, tak $M_i := \{f \mid \text{existuje } e(\mathrm{L})f,\ e \in B_i\}$ a symbol L pri $B_i$ zrušíme.
Potom pre každú triedu $B_j$ urobíme:
Ak sú obe množiny $B_j \cap M_i$ a $B_j - M_i$ neprázdne, tak urobíme:
Nech $B_{j1}$ a $B_{j2}$ označujú tieto dve množiny, kde $|B_{j1}| \leq |B_{j2}|$. Triedu $B_j$ nahradíme triedami $B_{j1}$ a $B_{j2}$, pričom $B_{j1}$ dostane oba symboly R a L a $B_{j2}$ len tie symboly, ktoré mala trieda $B_j$.
Inak $B_j$ zostáva.
Ideme na krok 1. ●

Poznamenajme, že po rozštiepení $B_j$ na $B_{j1}$ a $B_{j2}$ sme dali oba symboly R a L iba

k $B_{j_1}$. To sa ukazuje dôležité pre dosiahnutie nízkej zložitosti algoritmu. Na druhej strane pre správnu funkciu algoritmu to postačuje, ako dokážeme.

**Veta 1.** (a) Uvedený algoritmus vystačí s $O(n \log n)$ operáciami.

(b) Po skončení algoritmu sú dva šípy $e_1$, $e_2$ v rôznych triedach $\Leftrightarrow$ $e_1$ a $e_2$ sú rozlíšiteľné.

DÔKAZ. (a) Po utvorení $M_i$ vždy jeden symbol pri $B_i$ zrušíme a tento symbol sa niekedy znovu priradí iba podmnožine (triedy $B_i$), ktorej mohutnosť nepresahuje $|B_i|/2$. Teda každý šíp sa môže objaviť v nanajvýš 2 log $m$ triedach $B_i$, z ktorých sa odstránil symbol (R, resp. L). Na jedno prebehnutie kroku 1 stačí $O(|B_i|)$ operácií. Takto nám celkove stačí $O(n \log n)$ operácií.

(b) ($\Rightarrow$) Ukážme, že v každej fáze algoritmu platí: ak $e_1$, $e_2$ sú v rôznych triedach, tak sú rozlíšiteľné. Na začiatku je to zrejme pravda. Vo všeobecnosti ukážeme, že ak sa podľa algoritmu $B_j$ rozštiepi na $B_j \cap M_i$ a $B_j - M_i$, tak to platí tiež. Nech $e_1, e_2 \in B_j$, $e_1 \in M_i$ a $e_2 \notin M_i$. Potom existuje $e_1' \in B_i$ a $e_2' \notin B_i$ tak, že $e_1'(\mathrm{R})e_1$ a $e_2'(\mathrm{R})e_2$, alebo $e_1'(\mathrm{L})e_1$ a $e_2'(\mathrm{L})e_2$ sú hraničné sledy. Potom však existujú $\bar{e}_1(\mathrm{L})\bar{e}_1'$ a $\bar{e}_2(\mathrm{L})\bar{e}_2'$ alebo $e_1(\mathrm{R})\bar{e}_1'$ $\bar{e}_1(\mathrm{R})\bar{e}_2'$. Podľa indukčného predpokladu sú $e_1'$ a $e_2'$, a teda podľa lemy 1(iii) aj $\bar{e}_1'$ a $\bar{e}_2'$ rozlíšiteľné, takže potom podľa lemy 1(ii) a (iii) aj $e_1$ a $e_2$ sú rozlíšiteľné.

($\Leftarrow$) Dokazujeme indukciou podľa dĺžky $s$ najkratšieho reťazca $y$ symbolov R, resp. L, cez ktorý sa $\bar{e}_1$ a $\bar{e}_2$ rozlišujú. Pre $s = 0$ je tvrdenie zrejmé. Inak existujú $e_1'$ a $e_2'$ z rôznych základných tried tak, že $\bar{e}_1(y)e_1'$ a $\bar{e}_2(y)e_2'$ sú hraničné sledy. Budeme predpokladať, že $y = \mathrm{R}x$ (pre $y = \mathrm{L}x$ je dôkaz analogický). Takto existujú $\bar{e}_3$ a $\bar{e}_4$ tak, že $\bar{e}_1(\mathrm{R})\bar{e}_3$ a $\bar{e}_2(\mathrm{R})\bar{e}_4$ sú hraničné sledy. Keďže existujú $\bar{e}_3(x)e_1'$ a $\bar{e}_4(x)e_2'$, tak podľa indukčného predpokladu sa $e_3$ a $e_4$ dostali do rôznych tried, a teda hneď aspoň jedna z týchto tried dostala oba symboly R a L: nech je to trieda $B_i$ a $e_3 \in B_i$. Potom ešte niekedy budeme rušiť symbol L pri $B_i$ a vtedy $e_1 \in M_i$, lebo existuje $e_3(\mathrm{L})e_1$, ale $e_2 \notin M_i$. Preto sa $e_1$ a $e_2$ dostanú (rozštiepením nejakej triedy $B_j$) do rôznych tried. ■

**Veta 2.** Nech je daný ošípkovaný rovinný diagram grafu, ktorého každý komponent je vrcholovo 3-súvislý a nech $e_1$ a $e_2$ sú ľubovoľné dva jeho šípy. Potom $e_1$ a $e_2$ sú nerozlíšiteľné $\Leftrightarrow$ existuje silný automorfizmus ošípkovaného diagramu zobrazujúci $e_1$ na $e_2$.

DÔKAZ. ($\Leftarrow$) Zrejmé.

($\Rightarrow$) Požadovaný automorfizmus zostrojíme tak, že komponent $D_1$ diagramu obsahujúci šíp $e_1$ zobrazíme na komponent $D_2$ obsahujúci šíp $e_2$ a pritom $e_1$ na $e_2$.



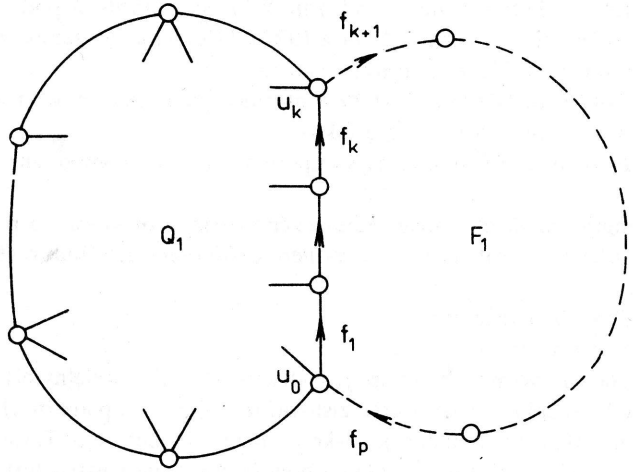


Ostatné komponenty sa zobrazia na seba identicky. Kedykoľvek zobrazíme jeden šíp, tak tým automaticky zobrazíme jeho začiatok aj koniec, a tým tiež zobrazíme obrátený šíp toho istého rebra.

Najprv zobrazíme $e_1$ na $e_2$ a celú hranicu steny vpravo od $e_1$ na hranicu steny vpravo od $e_2$ (obe hranice sú cykly tej istej dĺžky, lebo inak by $e_1$ a $e_2$ boli rozlíšiteľné). Týmto sme zobrazili sled $e_1(\mathrm{RR} \ldots \mathrm{R})e_1$ na sled $e_2(\mathrm{RR} \ldots \mathrm{R})e_2$. Toto zobrazenie v každom ďalšom kroku rozšírime o zobrazenie hranice jednej steny.

Predpokladajme, že už máme zobrazený nejaký vrcholovo 2-súvislý vnútorne maximálny poddiagram $Q_1$ diagramu $D_1$ (t. j. časť diagramu $D_1$ nachádzajúca sa vo vnútri hranice vonkajšej steny diagramu $Q_1$ celá patrí do $Q_1$). Nech $Q_1$ sa silne izomorfne zobrazí na nejaký poddiagram $Q_2$ diagramu $D_2$, pričom korešpondujúce šípy sú nerozlíšiteľné. Ak $Q_1 = D_1$, sme hotoví, a preto nech $Q_1 \neq D_1$. Potom v $D_1$ existuje stena $F_1$, ktorá má s vonkajšou hranicou diagramu $Q_1$ spoločnú aspoň jednu hranu. Naviac, stenu $F_1$ možno zvoliť tak, aby prienik ich hraníc bol súvislý, teda tvorený nejakou cestou $f_1 f_2 \ldots f_k$, pričom $F_1$ leží vpravo od $f_1$ (obr. 3). Nech $f_1$ odchádza z vrcholu $u_0$ a nech $f_{k+1}, \ldots, f_p$ sú zvyšné šípy cyklu tvoreného hranicou steny $F_1$, ako na obr. 3.

Nech $g_1$ je obrazom šípa $f_1$ a $v_0$ obrazom vrcholu $u_0$ v diagrame $Q_2$. Podľa predpokladu sú $f_1$ a $g_1$ nerozlíšiteľné, a preto môžeme sledu $S_1 = f_1(\mathrm{RR} \ldots \mathrm{R})f_p$ v $D_1$ priradiť sled $S_2 = g_1(\mathrm{RR} \ldots \mathrm{R})g_p$ v $D_2$. $S_2$ musí končiť vo $v_0$, lebo $f_1$ a $g_1$ majú vpravo steny s rovnako dlhými hranicami. Ďalej, $g_1, \ldots, g_k$ ležia v $Q_2$ podľa predpokladu o silnom izomorfizme medzi $Q_1$ a $Q_2$. Ak by napr. šíp $g_{k+1} \in Q_2$, tak




Obr. 3. K dôkazu vety 2.

jeho začiatok $v_k$ by mal v $D_2$ menší stupeň, ako má $u_k$ v $D_1$ (lebo v $Q_1$ a $Q_2$ majú stupne rovnaké). Potom by však $f_k$ a $g_k$ boli rozlíšiteľné, a teda aj $f_1$ a $g_1$ by boli také, čo je spor s indukčným predpokladom (korešpondujúce šípy v $Q_1$ a $Q_2$ sú nerozlíšiteľné). Analogicky je vidieť, že aj $g_{k+2}, \ldots, g_p \notin Q_2$.

Takto vieme zobraziť (silne izomorfne) $Q_1$ spolu s hranicou steny $F_1$ na $Q_2$ rozšírené o hranicu nejakej steny $F_2$ v $D_2$. Tieto nové diagramy $Q_1'$ a $Q_2'$ sú zrejme vnútorne maximálne poddiagramy a korešpondujúce šípy sú nerozlíšiteľné. Opakovaním tohto postupu silne izomorfne zobrazíme celý diagram $D_1$ na $D_2$. ■

Tým máme teoreticky vyriešený problém izomorfizmu vrcholovo 3-súvislých planárnych grafov. Tento postup zrejme možno použiť aj v prípade grafov, ktorých každý komponent je vrcholovo 3-súvislý. Tiež je vidieť, že takýto postup sa hodí aj v prípade, že hrany, resp. vrcholy sú nejako označkované (šípy s rôznymi značkami dáme do rôznych základných tried).

## 11.D
## Izomorfizmus planárnych grafov

V časti 11.A sme videli, že existuje polynomiálny algoritmus na zisťovanie izomorfizmu stromov a v 11.C sme uviedli taký algoritmus pre vrcholovo 3-súvislé planárne grafy. Tu ukážeme, že ani izomorfizmus planárnych grafov vo všeobecnosti nepredstavuje veľký problém, ak využijeme spomínané algoritmy pre špeciálne prípady. Túto možnosť uvádza už Weinberg (1966) a podrobnejšie sa tým zaoberali Hopcroft a Tarjan (1971 a 1972). Uvedieme len základnú myšlienku takéhoto postupu. Vhodné je uvažovať multigrafy.

Nech je daný planárny multigraf $G$ a planárny multigraf $H$, o ktorých treba zistiť, či sú izomorfné. Postupujeme takto:

Pre $G$ a $H$ vieme nájsť rozklad na komponenty a každý komponent rozložiť na bloky.

Bloky nemajú vo všeobecnosti jednoznačné rovinné diagramy, a preto sa ich snažíme na takéto rozložiť. Tzv. 3-*komponent* definujeme ako multigraf, ktorý má jeden z tvarov:

(a) vrcholovo 3-súvislý graf,
(b) cyklus dĺžky aspoň 3,
(c) pozostáva z dvoch vrcholov spojených aspoň tromi (paralelnými) hranami.

Podľa predchádzajúcej časti vieme zisťovať izomorfizmus planárnych multigrafov, ktorých každý komponent je 3-komponent. Hopcroft a Tarjan (1973a) dávajú lineárny algoritmus pre rozklad ľubovoľného (aj neplanárneho) multigrafu





na 3-komponenty. Tu sa týmto nebudeme zaoberať, ale len poznamenávame, že ich algoritmus je založený na prieskume do hĺbky a využíva podobné myšlienky ako algoritmus pre planaritu (kap. 10). Na účely zisťovania izomorfizmu však stačí vedieť rozpoznávať a odštiepovať tzv. *koncové* 3-*komponenty*.

Nech $B$ je koncový alebo izolovaný blok multigrafu $G$; nech $B$ má aspoň tri vrcholy a nie je 3-komponentom. Ak $B$ je koncový, nech $c$ je (jediná) artikulácia v $B$. Potom $B$ obsahuje také dva (rôzne) vrcholy $u$, $w$, každý stupňa aspoň 3 (v $G$), že vznikne niektorý z prípadov:

(i) $u$ a $w$ sú spojené $t \geq 2$ paralelnými hranami,

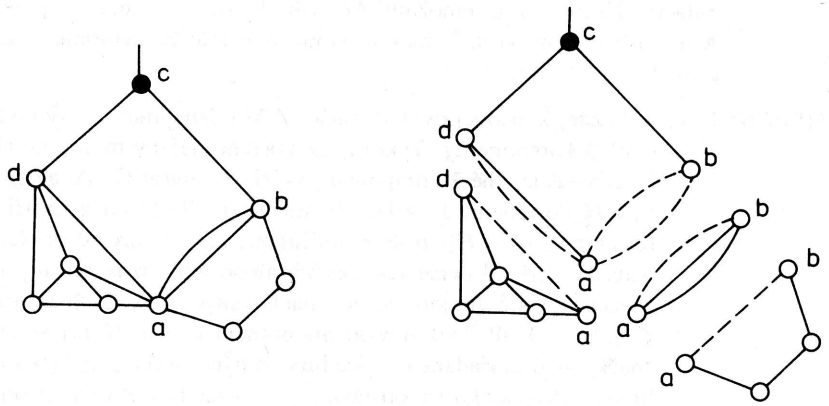
(ii) $u$ a $w$ sú spojené cestou $P$ dĺžky aspoň 2, ktorej každý vnútorný vrchol má v $G$ stupeň rovný 2,

(iii) multigraf $B - u - w$ je nesúvislý a má taký komponent $F$, ktorý neobsahuje $c$ (ak $B$ je koncový blok) a pritom $B(V(F) \cup \{u, w\}) + e_1$ je 3-súvislý multigraf, kde $e_1$ je nová $uw$ hrana.

V každom z týchto troch prípadov množinu $\{u, w\}$ nazývame *koncová dvojartikulácia*. Rozštiepením každého z vrcholov $u$, $w$ možno odštiepiť z $G$ istý multigraf $T$, ktorý nazveme *koncový* 3-*komponent*, a zostane nám menší multigraf $G_1$:

V prípade (i) $T$ pozostáva z vrcholov $u$ a $w$, všetkých hrán, ktoré ich spájali v $G$ a jednej novej $uw$ hrany $e_1$, ktorú pridávame. Potom $G_1$ vznikne z $G$ vynechaním $uw$ hrán, ktoré sme dali do $T$ (vrcholy $u$, $w$ tam budú, lebo pôvodné štiepime) a pridaním jednej novej $uw$ hrany (pozri obr. 1).

V prípade (ii) $T$ pozostáva z vrcholov a hrán cesty $P$ a novej $uw$ hrany $e_1$. Vynechaním všetkých hrán a vnútorných vrcholov cesty $P$ z $G$ a pridaním novej $uw$ hrany $e_2$ získame $G_1$ (obr. 1).




Obr. 1. Odštiepenie koncových 3-komponentov.

Konečne v prípade (iii) $T = B(V(F) \cup \{u, w\}) + e_1$ a $G_1 = G - V(F) + e_2$, kde $e_1$, $e_2$ sú nové $uw$ hrany (pozri obr. 1).

Hneď vidieť, že $T$ je vždy 3-komponent. Naviac možno ukázať, že koncové 3-komponenty sú jednoznačne určené (11.Z.7). Všimnime si (obr. 1), že niektorá koncová dvojartikulácia môže slúžiť na odštiepenie viacerých 3-komponentov súčasne. Tiež je dobré si uvedomiť, že každá nová hrana indikuje cestu, ktorá viedla cez odštiepenú časť.

To bolo štiepenie v izolovanom alebo koncovom bloku cez dvojartikulácie. Ešte môžeme odštiepiť celý koncový blok, ak je 3-komponentom. Takéto štiepenie pozostáva z rozštiepenia jediného vrcholu — artikulácie grafu $G$, pričom hrany koncového bloku budú v jednej odštiepenej časti a ostatné v druhej.

Vzhľadom na to, že izomorfné multigrafy musia mať izomorfné komponenty, koncové bloky a koncové 3-komponenty, možno navrhnúť nasledujúci algoritmus na zisťovanie izomorfizmu planárnych multigrafov. Jeho správnosť vyplýva z uvedenej diskusie.

**Algoritmus**

KROK 1: Dané dva planárne multigrafy $G$ a $H$, medzi ktorými hľadáme izomorfizmus, rozložíme na komponenty. Položíme $G_0 := G$, $H_0 := H$ a ideme na krok 2.

KROK 2: V každom komponente multigrafov $G_0$ a $H_0$ vyhľadáme všetky koncové bloky. Ak existuje aspoň jeden, ideme na krok 3, inak na krok 4.

KROK 3: Vo všetkých koncových blokoch vyhľadáme všetky koncové dvojartikulácie. Nech tvoria množinu $M$. Ak $M \neq \emptyset$, urobíme štiepenie A a vrátime sa na krok 3, inak urobíme štiepenie B a vrátime sa na krok 2.

ŠTIEPENIE A: V každej koncovej dvojartikulácii z $M$ odštiepime všetky koncové 3-komponenty. Takto z $G_0$ zostane nejaký multigraf $G_1$ a nech odštiepené 3-komponenty tvoria multigraf $G_1'$. Analogicky z $H_0$ dostaneme $H_1$ a $H_1'$. Zisťujeme (podľa 11.C) izomorfizmus medzi $G_1'$ a $H_1'$; pritom odlišujeme nové hrany od starých, označkované od neoznačkovaných alebo inak označkovaných, a označkované vrcholy od neoznačkovaných. Ak $G_1' \not\cong H_1'$ tak $G \not\cong H$ a STOP. Inak novým $uw$ hranám v $G_1$ a $H_1$ priradíme značky — usporiadané dvojice ľubovoľných čísel $(s_u, s_w)$ (tá istá hrana zapísaná ako $wu$ dostáva $(s_w, s_u)$), ale tak, aby dve takéto dvojice $(s_u, s_w)$ a $(s_x, s_y)$ boli rovnaké práve vtedy, keď medzi





príslušnými odštiepenými 3-komponentmi existuje izomorfizmus zobrazujúci $u$ na $x$ a $w$ na $y$. Položíme $G_0 := G_1$, $H_0 := H_1$.

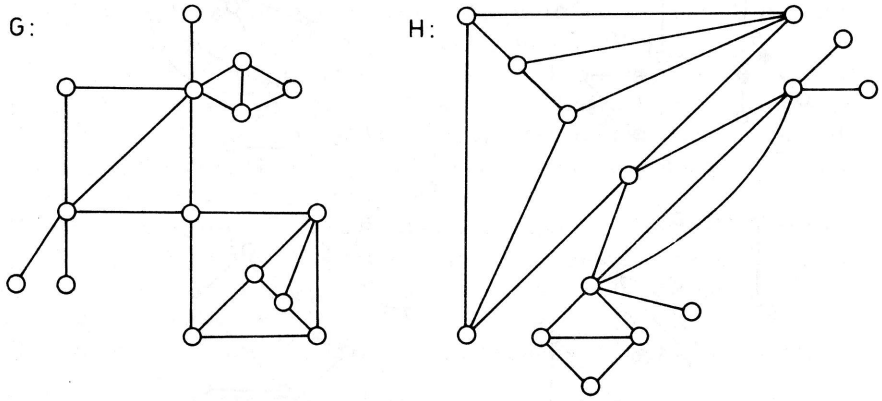
ŠTIEPENIE B: Odštiepime všetky koncové bloky v artikuláciách. Tým z $G_0$ zostane $G_1$ a nech odštiepené bloky tvoria multigraf $G_1'$. Nech $H_1$ a $H_1'$ vzniknú analogicky z $H_0$. Zisťujeme (podľa 11.C) izomorfizmus medzi $G_1'$ a $H_1'$; pritom odštiepené vrcholy odlišujeme od iných a zohľadňujeme značkovanie ako v štiepení A. Ak $G_1' \not\cong H_1'$, tak $G \not\cong H$ a STOP. Inak odštiepené vrcholy (predtým artikulácie) v $G_1$ a $H_1$ dostanú značky — každý vrchol $x$ jedno ľubovoľné číslo $t_x$, ale tak, aby dve takéto čísla $t_x$ a $t_y$ boli rovnaké práve vtedy, keď medzi príslušnými odštiepenými multigrafmi pri $x$ a $y$ (pozostávajúcimi z koncových blokov pri $x$ a $y$) existuje izomorfizmus zobrazujúci $x$ na $y$. Položíme $G_0 := G_1$ a $H_0 := H_1$.

KROK 4: Vo všetkých blokoch (máme len izolované) vyhľadáme množinu $M$ koncových dvojartikulácií. Ak $M \neq \emptyset$, urobíme štiepenie A a vrátime sa na krok 4, inak ideme na krok 5.

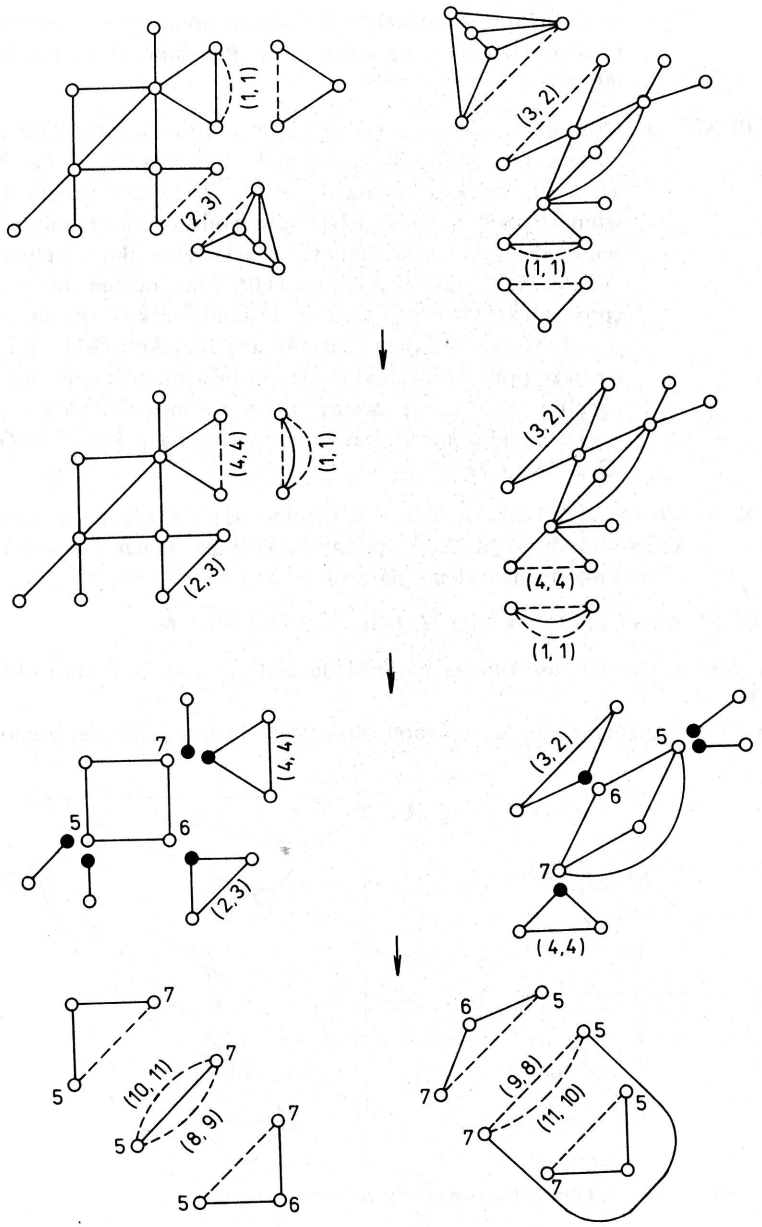
KROK 5: Ak $G_0 \not\cong H_0$, tak $G \not\cong H$. Inak $G \cong H$. STOP. ●

Uvedený algoritmus ilustrujeme na príklade grafov z obr. 2. Postup ukazuje obr. 3.

Na záver poznamenajme, že v polynomiálnom čase možno zisťovať aj izomorfiz-




Obr. 2. Planárne grafy na ilustráciu algoritmu.








mus grafov vnoriteľných na plochu rodu $r$ (pre ľubovoľné pevné $r \geq 0$) [Miller (1980)], [Filotti, Mayer (1980)].

## 11.Z
## Cvičenia a poznámky

1. Navrhnite nejaký postup pre zisťovanie izomorfizmu digrafov.

   Možno očakávať, že navrhnete alebo nepolynomiálny algoritmus alebo polynomiálnu heuristiku. Mnohé návrhy nájde čitateľ v literatúre. Napr. [Bertziss (1973)], [Corneil, Gotlieb (1970)]; prehľadné články sú [Read, Corneil (1977)], [Gati (1979)] a tiež [Tarjan (1980)]. K problému izomorfizmu existujú polynomiálne ekvivalentné úlohy z algebraických štruktúr a i. (pozri napr. [Garey, Johnson (1979)] a [Corneil, Kirkpatrick (1980)], kde sú tiež analyzované niektoré heuristiky).
2. Ukážte, že problémy izomorfizmu pre pseudomigrafy, digrafy a grafy sú polynomiálne ekvivalentné.
3. Ukážte, že v probléme izomorfizmu sa možno obmedziť na: (a) bipartitné grafy, (b) acyklické digrafy, (c) pravidelné grafy.

   Luks (1982) dáva polynomiálny algoritmus pre izomorfizmus grafov s pevným maximálnym stupňom. Využíva sa tam teória grúp. (Pozri tiež knihu [Hoffmann (1982)].)
4. (i) Dokážte, že problém izomorfného podstromu $H$ v grafe $G$ je NP-ťažký, dokonca ak $H$ je strom s maximálnym stupňom 2 (teda cesta) a $G$ je planárny bipartitný graf s maximálnym stupňom 3. (ii) Zapíšte úlohu izomorfného podgrafu $H$ v grafe $G$ ako 0—1 úlohu lineárneho programovania.

   Poznamenajme, že problém nájsť v danom strome $G$ podgraf izomorfný s daným lesom $H$ je NP-ťažký [Garey, Johnson (1979)].
5. Nech $G$ a $H$ sú grafy s $V(G) = \{v_1, \ldots, v_n\}$, $V(H) = \{u_1, \ldots, u_n\}$ $(n \geq 3)$ a nech $G - v_i \cong H - u_i$ pre $i = 1, \ldots, n$. V tzv. *Ulamovej hypotéze* sa tvrdí, že potom $G \cong H$, ale zatiaľ zostáva otvorená. (Mnoho čiastkových riešení nájde čitateľ v [Behzad et al. (1979)]. Dokážte túto hypotézu pre prípady: (a) $G$ a $H$ sú pravidelné, (b) $G$ a $H$ sú nesúvislé.
6. Nech je dané pevné $n$. Nájdite čo najväčšie číslo $s^*$ a $n$-vrcholový graf $G^*$ tak, aby každý $n$-vrcholový graf $G$ a graf $G^*$ obsahovali izomorfné $s^*$-vrcholové indukované podgrafy. Maximálne možné $s^*$ nie je známe [Zelinka (1975)].
7. Dokážte, že koncové 3-komponenty multigrafu sú jednoznačne určené.

◄ Obr. 3. Zisťovanie izomorfizmu grafov z obr. 2.

# LITERATÚRA


* Adeľson-Veľskij, G. M., Dinic, E. A., Karzanov, A. V. (1975): Potokovyje algoritmy. Moskva, Nauka, [4.F, 6.D, 6.E, 6.F].

Aho, A. V., Garey, M. R., Ullman, J. D. (1972): The transitive reduction of a directed graph. SIAM J. Comput., 1, 131—137, [3.G].

* Aho, A. V., Hopcroft, J. E., Ullman, J. D. (1974): The design and analysis of computer algorithms. Reading, Mass., Addison-Wesley, (ruský preklad: Moskva 1979), [1.G, 2.A, 2.C, 2.E].

Appel, K., Haken, E. (1976): Every planar map is four colorable. Bull. Amer. Math. Soc., 82, 711—712, [1.A].

Auslander, L., Parter, S. V. (1961): On imbedding graphs in the plane. J. Math. and Mech., 10, 517—523, [10.C].

Babai, L., Kučera, L. (1979): A graph canonization with linear expected time. Proc. 19th IEEE Symp. Found. Computer Sci., Puerto Rico, [2.E].

* Behzad, M., Chartrand, G., Lesniak-Foster, L. (1979): Graphs and digraphs. Boston, Prindle, Weber and Schmidt, [1.A, 9.C, 11.Z].

Bellman, R. E. (1958): On a routing problem. Quart. Appl. Math., 16, 87—90, [4.B].

Berge, C. (1957): Two theorems in graph theory. Proc. Nat. Acad. Sciences USA, 43, 842—844, [8.A].

* Berge, C. (1958): Théorie des graphes et ses applications. Paris, Dunod (ruský preklad: Moskva 1962), [5.Z, 6.D].

* Berge, C. (1970): Graphes et hypergraphes. Paris, Dunod, (anglický preklad: London 1973), [2.E, 3.Z].

* Berge, C., Ghouila-Houri, A. (1962): Programmes, jeux et réseaux de transport. Paris, Dunod, (anglický preklad: London 1965), [6.F].

Berztiss, A. T. (1973): A backtrack procedure for isomorphism of directed graphs. J. ACM, 20, 365—377, [11.Z].

* Biggs, N. L., Lloyd, E. K., Wilson, R. J. (1976): Graph theory 1736—1936. Oxford, Clarendon Press, [1.A].

Bílý, J., Fiedler, M., Nožička, F. (1958): Die Graphentheorie in Anwendung auf das Transportproblem. Czechoslovak Math. J., 8(83), 94—121, [6.D].

Blattner (pozri Dantzig).

Bock, F. C. (1971): An algorithm to construct a minimum directed spanning tree in a directed network. Developments in Operations Research. New York, Gordon and Breach, 29—44, [5.F].

Bodin (pozri Golden).

* Bondy, J. A., Murty, U.S.R. (1976): Graph theory with applications. London, Macmillan, [predhovor].







Borosch, I., Treybig, L. B. (1976): Bounds on positive integral solutions of linear Diophantine equations. Proc. Amer. Math. Soc., 55, 299—304, [2.B].

Borůvka, O. (1926): O jistém problému minimálním. Práce moravské přírodověd. spol. (Brno), III(3), 37—58, [5.B].

Bosák, J. (1979): Ako bol vyriešený problém štyroch farieb. Pokroky mat. fyz. astr., 24, 181—201, [1.A].

* Bosák, J. (1980): Grafy a ich aplikácie. Bratislava, Alfa, [predhovor, 1.B].

Bosák, J., Rosa, A., Znám, Š. (1968): On decompositions of complete graphs into factors with given diameters. Theory of graphs. Budapest, Akadémiai Kiadó, 37—56, [7.Z].

* Brunovský, P., Černý, J. (1980): Základy matematickej teórie systémov. Bratislava, Veda, [predhovor].

Bulitko, V. K. (1973): O grafach s zadannymi okruženijami veršin. Trudy matem. inst. im. V. A. Steklova, 133, 78—94, [2.A].

Bulitko, V. K. (1973a): Ob odnoj algoritmičeski nerazrešimoj massovoj zadače dľa grafov. Upravľajemyje sistemy, vyp. 2. Novosibirsk, Nauka, 61—73, [2.A].

Busacker, R. G., Gowen, P. J. (1961): A procedure for determining a family of minimal-cost network flow patterns. Operations Research Office, John Hopkins University, Technical report, 15, [6.D].

* Busacker, R. G., Saaty, T. L. (1965): Finite graphs and networks. New York, McGraw-Hill, (ruský preklad: Moskva 1974), [6.D, 9.B, 11.A].

Cantor, G. (1874): Über eine Eigenschaft des Inbegriffes aller reelen algebraischen Zahlen. J. reine angew. Math., 77, 258—262, [2.A].

Cartwright (pozri Harary).

Cayley, A. (1874): On the mathematical theory of isomers. Philos. Mag., 67, 444-446, [1.A].

Cederbaum (pozri Lempel).

Cook, S. A. (1971): The complexity of theorem-proving procedures. Proc. Third Annual ACM Symp. on Theory of Computing, 151—158, [2.C, 2.D].

Cook, S. A., Reckhow, R. A. (1973): Time-bounded random access machines. J. Computer Syst. Sci, 7, 354—373, [2.A].

Corneil, D. G., Gotlieb, C. C. (1970): An efficient algorithm for graph isomorphism. J. ACM, 17, 51—64, [11.Z].

Corneil, D. G., Kirkpatrick, D. G. (1980): A theoretical analysis of various heuristics for the graph isomorphism problem. SIAM J. Comput., 9, 281—297, [11.Z].

Corneil (pozri Read).

Cornuejols, G., Nemhauser, G. L. (1978): Tight bounds for Christofides' traveling salesman heuristic. Math. Programming, 14, 116—121, [9.C].

* Čech, E. (1966): Bodové množiny. Praha, Academia, [10 (úvod)].

Čerkasskij, B. V. (1973): Konečnyj algoritm rešenija zadači o dvuchproduktovom potoke. Ekonomika i matematičeskije metody, 9, 1147—1149, [6.E].

Čerkasskij. B. V. (1977): Algoritm postrojenija maximaľnogo potoka v seti s trudojomkostju $O(n^2p)$ dejstvij. Matematičeskije metody rešenija ekonomičeskich zadač, Sb. 7. Moskva, Nauka, 117—126, [6.B].

Černý, J. (1972): Transportnyje seti s usilenijem. Ekon.-mat. obzor, 8, 299—309, [6.Z].
Černý (pozri Brunovský).
* Čulík, K., Doležal, V., Fiedler, M. (1967): Kombinatorická analýza v praxi. Praha, SNTL, [predhovor, 5.A, 5.B, 5.C].
* Čupr, K. (1949): Geometrické hry a zábavy. Praha, Cesta k vědení, [3.A].
Dambit (pozri Grinberg).
Dantzig, G. B. (1951): Application of the simplex method to a transportation problem. Activity Analysis of Production and Allocation, Cowles Comission Monograph 13. New York, John Wiley, 359—373, [6.D].
Dantzig, G. B. (1962): Linear programming and extensions. Princeton, Princeton University Press (slovenský preklad: Bratislava 1966), [2.B, 6.D, 6.Z].
Dantzig, G. B. (1966): All shortest routes in a graph. Oper. Res. House, Stanford University Tech. Rep., 66—3, [4.Z].
Dantzig, G. B., Blattner, W., Rao, M. R. (1967): Finding a cycle in a graph with minimum cost to time ratio with application to a ship routing problem. Theory of graphs. Paris, Dunod, 77—84, [4.F].
Dantzig, G. B., Fulkerson, D. R. (1954): Minimizing the number of tankers to meet a fixed schedule. Naval Res. Logist. Quart., 1, 217—222, [6.C].
Dantzig, G. B., Fulkerson, D. R. (1956): On the max-flow mincut theorem of networks. Linear inequalities and related systems, Annals of Math. Study, 38. Princeton, Princeton University Press, 215—221, [7.A].
Dantzig, G. B., Fulkerson, D. R., Johnson, M. S. (1954): Solution of a large-scale traveling salesman problem. Operations Res., 2, 393—410, [9.C].
Davis, E. W., Patterson, H. J. (1975): A comparison of heuristic and optimum solutions in resource-constrained project scheduling. Management Sci., 21, 944—955, [4.Z].
* Deo, N. (1974): Graph theory with applications to engineering and computer science. Englewood Cliffs, New Jersey, Prentice-Hall, Inc., (poľský preklad: Warszawa 1980).
Deo, N. (1976): Note on Hopcroft and Tarjan planarity algorithm. J. ACM, 23, 74—75, [10.C].
Deo (pozri Reingold).
Dijkstra, E. W. (1959): A note on two problems in connexion with graphs. Numer. Math., 1, 269—271, [4.B, 5.B].
Dinic, E. A. (1970): Algoritm rešenija zadači o maximaľnom potoke v seti so stepennoj ocenkoj. Doklady Akad. Nauk SSSR, 194, 754—757, [6.B].
Dinic, E. A. (1973): Algoritm porazrjadnogo sokraščenija nevjazok i transportnyje zadači. Isledovanija po diskretnoj matematike. Moskva, Nauka, 46—57, [6.B].
Dinic, E. A., Zajcev, M. A., Karzanov, A. V. (1974): Ekonomnyj algoritm vydelenija blokov v grafe. Žurnal vyčisl. matem. i matem. fiziki, 14, 1309—1316, [3.B].
Dinic (pozri Adeľson–Velskij).
Dirac, G. A. (1952): Some theorems on abstract graphs. Proc. London Math. Soc., 2, 69—81, [9.C].
Doležal (pozri Čulík).
Doyle (pozri Golden).





Dreyfus, S. E., Wagner, R. A. (1971): The Steiner problem in graphs. Networks, 1, 194—207, [5.G].
Dulmage (pozri Mendelsohn).
Edmonds, J. (1965): Path, trees and flowers. Can. J. Math., 17, 449—467, [2.A, 8.B].
Edmonds, J. (1965a): Minimum partition of a matroid into independent subsets. J. Res. Nat. Bur. Standards, 69B, 67—72, [7.D].
Edmonds, J. (1965b): Maximum matching and a polyhedron with 0, 1 vertices. J. Res. Nat. Bur. Standards, 69B, 125—130, [8.C].
Edmonds, J. (1965c): The Chinese postman's problem. Bull. Operations Res. Soc. Suppl., 1, 73, [9.B].
Edmonds, J. (1967): Optimum branchings. J. Res. Nat. Bureau Standards, 71B, 233—240, [5.F].
Edmonds, J. (1967a): An introduction to matching. Mimeographed lecture notes. Engineering summer conference. Univ. of Michigan, Ann Arbor, [8.D].
Edmonds, J. (1973): Edge-disjoint branchings. Combinatorial Algorithms (Courant Inst. Sci. Symp. 9). New York, Algorithmic Press, 91—96, [7.C].
Edmonds, J. (1975): Some well-solved problems in combinatorial optimization. Combinatorial Programming: Methods and Applications. Dordrecht, Reidel, 285—301, [8.D].
Edmonds, J., Johnson, E. L. (1970): Matching: A well-solved class of integer linear programs. Combinatorial structures and their applications. New York, Gordon and Breach, 89—92, [8.D].
Edmonds, J., Johnson, E. L. (1973): Matching, Euler tours and the Chinese postman. Math. Programming, 5, 88—124, [9.A, 9.B, 9.Z].
Edmonds, J., Karp, R. M. (1970, 1972): Theoretical improvements in algorithmic efficiency for network flow problems. Combinatorial structures and their applications. New York, Gordon and Breach, 1970, 93—96 (abstract). J. AMC, 19 (1972), 248—264, [4.B, 4.C, 4.F, 6.A, 6.B, 6.D].
Egerváry, J. (1931): Matrixok kombinatorius tulajdonságoiról. Mat. Fiz. Lapok, 38, 16—28, [8.A, 8.B].
*Erdös, P., Spencer, J. (1974): Probabilistic methods in combinatorics. Budapest, Akadémiai Kiadó, (ruský preklad: Moskva 1976), [2.E].
Eswaran, K. P., Tarjan, R. E. (1976): Augmentation problems. SIAM J. Comput., 5, 653—665, [3.C].
Euler, L. (1736): Solutio problematis ad geometriam situs pertinentis. Comment. Acad. Sci. Imp. Petropol., 8, 128—140, [1.A].
Euler, L. (1752): Elementa doctrinae solidorum. Novi Comm. Acad. Sci. Imp. Petropol., 4, 109—140, [10.A].
Even, S. (1975): Algorithm for determining whether the connectivity of a graph is at least k. SIAM J. Computing, 4, 393—396, [7.A].
* Even, S. (1979): Graph algorithms. Potomac, Computer Science Press, [7.Z, 10.C].
Even, S., Itai, A., Shamir, A. (1976): On the complexity of timetable and multicommodity flow problems. SIAM J. Comput., 5, 691—703, [6.E, 6.F].
Even, S., Kariv, O. (1975): An $O(n^{2.5})$ algorithm for maximum matching in general graphs.

Proc. 16th Annual Symp. on Foundations of Computer Science. IEEE, New York, 100—112, [8.B].
Even, S., Tarjan, R. E. (1975): Network flow and testing graph connectivity. SIAM J. Computing, 4, 507—518, [7.A].
Even (pozri Lempel).
Faradžev, I. A. (1970): Effektivnyje algoritmy rešenija nekotorych zadač dľa orientirovannych grafov. Žurnal vyčisl. matem. i matem. fiziki, 10, 1049—1054, [3.E, 3.G].
Fiedler, M. (1973): Algebraic connectivity in graphs. Czech. Math. J., 23(98), 298—305, [7.A].
Fiedler, M. (1975): Algebraische Zusammenhangszahl der Graphen und ihre numerische Bedeutung. Numerische Methoden bei graphentheoretischen und kombinatorischen Problemen. Int. Schriftenreihe zur numer. Math. 29. Basel-Stuttgart, Birkhäuser Verlag, 69—85, [7.A].
*Fiedler, M. (1981): Speciální matice a jejich použití v numerické matematice. Praha, SNTL, [predhovor, 3.Z].
Fiedler, M., Nožička, F. (1961): Ob odnom kriterii v teorii transportnoj problemy. Czech. Math. J., 11, 204—212, [6.D].
Fiedler (pozri Bílý, Čulík).
Filotti, I. S., Mayer, J. N. (1980): A polynomial-time algorithm for determining the isomorphism of graphs of fixed genus. Proc. Twelfth Annual ACM Symposium Th. Computing, 236—243, [11.D].
Filotti, I. S., Miller, G. L., Reif, J. H. (1979): On determining the genus of a graph in $O(v^{O(g)})$ steps. Proc. Eleventh Ann. ACM Symposium on Theory of Computing, 27—37, [10.Z].
Fischer, M. J., Meyer, A. R. (1971): Boolean matrix multiplication and transitive closure. 12th Annual Symposium on Switching and Automata Theory. IEEE, New York, 129—131, [3.G].
Flood, M. M. (1953): On the Hitchcock distribution problem. Pacif. J. Math., 3, 369—386, [6.D].
Floyd, F. W. (1962): Algorithm 97: shortest path. Comm. ACM 5, 345, [4.C].
Ford, L. R. (1956): Network flow theory. The Rand Corp., P-923, [4.B].
Ford, L. R., Fulkerson, D. R. (1956): Maximal flow through a network. Canad. J. Math., 8, 399—404, [6.A].
Ford, L. R., Fulkerson, D. R. (1956a): Solving the transportation problem. Management Sci., 3, 24—32, [6.D].
Ford, L. R., Fulkerson, D. R. (1957): A simple algorithm for finding maximal network flows and an application to the Hitchcock problem. Canadian J. Math., 9, 210—218, [6.A].
*Ford, L. R., Fulkerson, D. R. (1962): Flows in networks. Princeton, Princeton U. Press, (ruský preklad: Moskva 1966), [6.A, 6.D, 6.Z, 8.B, 9.A].
Fortune, S., Hopcroft, J. E., Wyllie, J. C. (1980): The directed subgraph homeomorphism problem. Theor. Computer Sci., 10, 111—121 [6.Z].
*Frank, H., Frisch, I. T. (1971): Communication, transmission, and transportation networks. Reading, Mass., Addison-Wesley, (ruský preklad: Moskva 1980).





Frederickson, G. N., Hecht, M. S., Kim, C. E. (1978): Approximation algorithms for some routing problems. SIAM J. Comput., 7, 178—193, [9.B, 9.Z].
Frisch (pozri Frank).
Frobenius, G. (1912): Über Matrizen aus nicht negativen Elementen. Sitzungsber. König. Preuss. Akad. Wiss., 23, 456—477, [8.B].
Fulkerson, D. R. (1959): Increasing the capacity of a network, the parametric budget problem. Management Sci., 5, 472—483, [6.D].
Fulkerson, D. R. (1966): Flow networks and combinatorial operations research. Amer. Math. Monthly, 72, 115—138, [4.Z].
Fulkerson (pozri Dantzig, Ford).
Gabow, H. N., 1976: An efficient implementation of Edmond's algorithm for maximum matching on graphs. J. ACM, 23, 221—234, [8.B].
Gallai, T. (1959): Über extreme Punkt- und Kantenmengen. Ann. Univ. Sci. Budapest, Eötvös Sect. Math., 2, 133—138, [2.D, 8.Z].
Gallai, T. (1962): Graphen mit triangulierbaren ungeraden Vielecken. Magyar Tud. Akad. Kut. Int. Közleményei, 7, 3—36, [3.B].
Gallai, T. (1963): Neuer Beweis eines Tutte-schen Satzes. Magyar Tud. Akad. Matem. Kut. Int. Közl., 8, 135—139, [8.A].
Garey, M. R., Graham, R. L., Johnson, D. S. (1977): The complexity of computing Steiner minimal trees. SIAM J. Appl. Math., 32, 835—859, [5.G].
Garey, M. R., Graham, R. L. Johnson, D. S., Knuth, D. E. (1978): Complexity results for bandwidth minimization. SIAM J. Appl. Math., 34, 477—495, [3.Z].
Garey, M. R., Johnson, D. S. (1976): The complexity of near-optimal graph coloring. J. ACM, 23, 43—49, [2.D].
Garey, M. R., Johnson, D. S. (1977): The rectilinear Steiner tree problem in NP-complete. SIAM J. Appl. Math., 32, 826—834, [2.D, 2.Z: 5.G, 5.Z].
*Garey, M. R., Johnson, D. S. (1979): Computers and intractability, A guide to the theory of NP-completeness. San Francisco, Freeman (ruský preklad: Moskva 1982), [2.C, 2.D, 2.E, 2.Z, 4.D, 4.Z, 5.Z, 8.Z, 11.Z].
Garey, M. R., Johnson, D. S., Stockmeyer, L. (1976): Some simplified NP-complete graph problems. Theor. Comput. Sci., 1, 237—267, [2.D, 2.Z, 3.Z].
Garey, M. R., Johnson, D. S., Tarjan, R. E. (1976): The planar Hamiltonian circuit problem in NP-complete. SIAM J. Comput., 5, 704—714, [2.D].
Garey (pozri Aho).
*Garfinkel, R. S., Nemhauser, G. L. (1972): Integer Programming. New York, Wiley, (poľský preklad: Warszawa 1978), [4.Z].
*Gass, S. I. (1964): Linear programming: Methods and applications, 2nd ed. New York, McGraw-Hill, (slovenský preklad: Bratislava 1972), [6.D].
Gati, G. (1979): Further annotated bibliography on the isomorphism disease. J. Graph Theory, 3, 95—109, [11.Z].
Geldmacher (pozri Liu).
Ghouila-Houri, A. (1960): Une condition suffisante d'existence d'un circuit Hamiltonien. C. R. Acad. Sci. Paris, 156, 495—497, [9.C].

Ghouila-Houri (pozri Berge).
Gilbert (pozri Chung).
Gilmore, P. C., Gomory, R. E. (1964): Sequencing a one state-variable machine: a solvable case of the traveling salesman problem. Operations Res., 12, 655—679, [9.Z].
Gliviak, F. (1975): Maximal graphs with given connectivity and edge-connecitvity. Mat. čas., 25, 99—103, [7.Z].
Glover, F., Karney, D., Klingman, D. (1974): Implementation and computational comparisons of primal, dual and primal-dual computer codes for minimum cost network flow problems. Networks 4, 191—212, [6.D].
Glover, F., Klingman, D. (1975): Finding minimum spanning trees with a fixed number of links at a node. Combinatorial programming: methods and application. Dordrecht, Reidel, 191—202, [5.C].
Golden, B., Bodin, L., Doyle, T., Stewart, W. (1980): Approximate traveling salesman algorithms. Operations Res., 28, 694—711, [9.C].
Goldner (pozri Rosenthal).
* Goľštejn, E. G., Judin, D. B. (1969): Zadači linejnogo programmirovanija transportnogo tipa. Moskva, Nauka, [6.D].
* Golumbic, M. C. (1980): Algorithmic graph theory and perfect graphs. New York, Academic Press, [2.E].
Gomory, R. E., Hu, T. C. (1961): Multi-terminal network flows. J. SIAM, 9, 551—570, [7.B].
Gomory, R. E., Hu, T. C. (1962): An application of generalized linear programming to network flows. SIAM J. Appl. Math., 10, 260—283, [7.B].
Gomory (pozri Gilmore).
Gonzalez (pozri Sahni).
Goralčíková, A., Koubek, V. (1979): A reduct-and-closure algorithm for graphs. Lecture Notes in Computer Science 74. Mathematical Foundations of Computer Science 1979. Berlin, Springer–Verlag, 301—306, [3.E, 3.G].
Gotlieb (pozri Corneil).
Gowen (pozri Busacker).
Gödel, K. (1931): Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I. Monatshefte Math. Physik, 38, 173—198, [2.A].
Graham, R. L., Hwang, F. K. (1976): Remarks on Steiner minimal tress I. Bull. Inst. Math. Acad. Sinica, 4, 177—182 [5.G].
Graham (pozri Garey).
Grinberg, E. Ja., Dambit, Ja. Ja. (1966): Nekotoryje svojstva grafov soderžaščich kontury. Latv. mat. ježegodnik (Riga), 2, 65—70, [2.Z].
Grötschel, M., Lovász, L., Schrijver, A. (1981): The ellipsoid method and its consequences in combinatorial optimization. Combinatorica 1, 169—197, [2.B].
Gruska, J. (1975): Zložitosť konkrétnych algoritmov I. Základné princípy. (Učebné texty.) Výskumné výpočtové stredisko, Bratislava, [2.A, 2.E].
Haken (pozri Appel).





Hakimi, S. (1962): On the realizability of a set of integers as degrees of the vertices of a graph. J. SIAM Appl. Math., 10, 496—506, [1.Z].
Hakimi, S. L. (1971): Steiner's problem in graphs and its implications. Networks, 1, 113—133, [5.G].
Hakimi (pozri Kariv).
Hall, P. (1935): On representatives of subsets. J. London Math., Soc., 10, 26—30, [8.A].
Hanan, M. (1966): On Steiner's problem with rectilinear distance. SIAM J. Appl. Math., 14, 255—265, [5.G, 5.Z].
Harary, F. (1962): A graph theoretic approach to matrix inversion by partitioning. Numerische Math., 4., 128—135, [3.D].
* Harary, F. (1969): Graph theory. Reading, Addison–Wesley, (ruský preklad: Moskva 1973), [1.B, 1.Z, 3.B, 5.Z, 10.C].
* Harary, F., Norman, R. Z., Cartwright, D. (1965): Structural models: An introduction to the theory of directed graphs. New York, Wiley.
Havel, V. (1955): Poznámka o existenci konečných grafů. Časop. Pěst. Mat., 80, 477—480, [1.Z].
Hecht (pozri Frederickson).
Hilbert, D. (1926): Über das Unendliche. Math. Annalen, 95, 134—151, [2.A].
Hitchcock, F. L. (1941): The distribution of a product from several sources to numerous localities. J. Math. Phys., 20, 224—230, [6.D].
*Hoffmann, C. M. (1982): Group-theoretic algorithms and graph isomorphism. Lectures Notes in Computer science 136. Berlin, Springer-Verlag, [11.Z].
Holer, I. (1981): The NP-completeness of edge-colouring. SIAM J. Comput. 10, 718—720, [2.D].
Hopcroft, J. E., Karp, R. M. (1973): An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. SIAM J. Comput., 2, 225—231, [8B; 11.B].
Hopcroft, J. E., Tarjan, R. E. (1971): A $V^2$ algorithm for determining isomorphism of planar graphs. Inf. Processing Letters, 1, 32—34, [11.D].
Hopcroft, J. E., Tarjan, R. E. (1972): Isomorphism of planar graphs (working paper). Complexity of Computer Computations. New York, Plenum Press, 131—152, [11.A, 11.C, 11.D].
Hopcroft, J., Tarjan, R. E. (1973): Efficient algorithms for graph manipulation. Comm. ACM, 16, 372—378, [3.A, 3.B].
Hopcroft, J., Tarjan, R. E. (1973a): Dividing a graph into triconnected components. SIAM J. Computing, 2, 135—158, [7.A, 11.D].
Hopcroft, J. E., Tarjan, R. E. (1973b): A V log V algorithm for isomorphism of triconnected planar graphs. J. Computer System Sci., 7, 323—331, [11.C].
Hopcroft, J. E., Tarjan, R. E. (1974): Efficient planarity testing. J. ACM, 21, 549—568, [10.C].
Hopcroft, J. E., Wong, J. K. (1974): Linear time algorithm for isomorphism of planar graphs. Proc. Sixth Ann. ACM Symp. Th. Computing, 172—184, [11.C].
Hopcroft (pozri Aho, Fortune).

Horák, P. (1983): A note on removing a point of a strong digraph. Math. Slovaca 33, 85—86, [3.Z].
Hu, T. C. (1963): Multi-commodity network flows. J. ORSA, 11, 344—360, [6.E].
* Hu, T. C. (1969): Integer programming and network flows. Reading, Addison–Wesley, (ruský preklad: Moskva 1974), [2.B, 6.A].
Hu (pozri Gomory).
Hwang, F. K. (1976): On Steiner minimal trees with rectilinear distance. SIAM J. Appl. Math., 30, 104—114, [5.G].
Hwang (pozri Graham, Chung).
Čačijan, L. G. (1979): Polinomialnyj algoritm v linejnom programmirovanii. Dokl. Akad. Nauk SSSR, 244, 1093—1096, [2.B].
Čačijan (pozri Kozlov).
Chandrasekaran R. (1977): Minimal ratio spanning trees. Networks, 7, 335—342, [5.D].
Chartrand (pozri Behzad).
Cheriton, D., Tarjan, R. E. (1977): Finding minimum spanning tress. SIAM J. Comput., 5, 724—742, [5.B].
Choquet, G. (1938): Etude de certains réseaux de routes. C. R. Ac. Sciences Paris, 206, 310—311, [5.B].
* Christofides, N. (1975): Graph theory, An algorithmic approach. London, Academic Press, (ruský preklad: Moskva 1978), [2.Z, 4.F, 9.C].
Christofides, N. (1976): Worst-case analysis of a new heuristic for the travelling salesman problem. Technical Report No. 388, Graduate School of Industrial Administration, Pittsburgh, Carnegie-Mellon University, [9.C].
Christofides, N. (1979): The travelling salesman problem. Combinatorial optimization. New York, Wiley, 132—149, [9.C].
Chu, Y. J., Liu, T. H. (1965): On the shortest arborescence of a directed graph. Sci. Sinica, 14, 1396—1400, [5.F].
Chung, F. R. K., Gilbert, E. N. (1976): Steiner trees for the regular simplex. Bull. Inst. Math. Acad. Sinica, 4, 313—325, [5.G].
Chung, F. R. K., Hwang, F. K. (1978): A lower bound for the Steiner tree problem. SIAM J. Appl. Math., 34, 27—36, [5.G].
Church, A. (1936): An unsolvable problem of elementary number theory. American J. Math., 58, 345—363, [2.A].
Chvátal, V., Thomassen, C. (1978): Distances in orientations of graphs. J. Combin. Theory, 24B, 61—75, [7.Z].
Chvátal (pozri v [Garey, Johnson (1979)]), [8.Z].
Itai, A. (1978): Two commodity flow. J. ACM, 25, 596—611. [2.B, 6.C, 6.E, 6.F].
Itai, A., Perl, Y., Shiloach, Y. (1982): The complexity of finding maximum disjoint paths with length constraints. Networks, 12, 277—286, [7.Z].
Itai (pozri Even).
Jarník, V. (1930): O jistém problému minimálním. Práce mor. přírodověd. spol. (Brno), VI(4), 57—63, [5.B].





Jarník, V., Kössler, M. (1934): O minimálních grafech obsahujících n daných bodů. Časopis Pěst. Mat. Fyz., 63, 223—235, [5.G].
* Jermoliev, Ju. M., Meľnik, I. M. (1968): Ekstremaľnyje zadači na grafach. Kijev, Naukova dumka.
Jeroslow, R. G. (1973): There cannot by any algorithm for integer programming with quadratic constraints. Operations Res., 21, 221—224, [2.B].
Jewell, W. S. (1958): Optimal flow through networks. Massachusets Institute of Technology Inter. Tech. Report, 8, [6.D].
Jewell, W. S. (1962): Optimal flow through networks with gains. Operations Res., 10, 476—499, [6.Z].
Johnson, D. B. (1973): A note on Dijkstra's shortest path algorithm. J. ACM, 20, 385—388, [4.B, 4.C].
Johnson, D. B. (1977): Efficient algorithms for shortest paths in sparse networks. J. ACM, 24, 1—13, [4.B, 4.Z].
Johnson, D. S. (1974): Worst case behavior of graph coloring algorithms. Proc. 5th S—E Conf. Combinatorics, Graph Theory, Computing. Winnipeg, Utilitas Mathematica Publishing, 513—527, [2.Z].
Johnson, D. S., Lenstra, J. K., Rinnooy Kan, A. H. G. (1978): The complexity of the network design problem. Networks, 8, 279—285, [5.Z].
Johnson, D. S. (pozri Garey).
Johnson, E. L. (pozri Edmonds).
Johnson, M. S. (pozri Dantzig).
Jones, J. P. (1974): Recursive undecidability — an exposition. American Math. Monthly, 81, 724—738, [2.A].
* Jucovič, E. (1981): Konvexné mnohosteny. Bratislava, Veda, [10.B].
Judin (pozri Goľštejn).
Kajitani (pozri Kishi).
Kameda, T. (1976): On maximally distant spanning trees of a graph. Computing, 17, 115—119, [7.D].
Kariv, O., Hakimi, S. L. (1979): An algorithmic approach to network location problems I: the p-centers. SIAM J. Appl. Math., 37, 513—538, [2.Z].
Kariv, O., Hakimi, S. L. (1979a): An algorithmic approach to network location problems II: the p-medians. SIAM J. Appl. Math., 37, 539—560, [2.Z].
Kariv (pozri Even).
Karney (pozri Glover).
Karp, R. M. (1971): A simple derivation of Edmonds' algorithm for optimum branching. Networks, 1, 265—272, [5.F].
Karp, R. M. (1972): Reducibility among combinatorial problems. Complexity of computer computations. New York, Plenum Press, 85—104, [2.B, 2.C, 2.D, 2.Z].
Karp, R. M. (1977): Probabilistic analysis of partitioning algorithms for the traveling salesman problem in the plane. Math. Operations Res., 2, 209—224, [2.E].
Karp (pozri Edmonds, Hopcroft).

Karzanov, A. V. (1974): Nachoždenije maximaľnogo potoka v seti metodom predpotokov. Dokl. Akad. Nauk SSSR, 215, 49—53, [6.B].
Karzanov (pozri Dinic, Adeľson-Veľskij).
Kerschenbaum, A., Van Slyke, R. (1972): Computing minimum spanning trees efficiently. Proc. 25th Ann. Conf. of the ACM 1972, 518—527, [5.B].
Kim, C. (1975): A minimal spanning tree and approximate tours for a traveling salesman. Comp. Sci. Tech. Report, University of Maryland, [9.C].
Kim (pozri Frederickson).
Kirchhoff, G. (1847): Über die Auflösung der Gleichungen, auf welche man bei der Untersuchung der linearen Verteilung galvanischer Ströme geführt wird. Ann. Phys. Chem., 72, 497—508, [1.A, 5.A].
Kirkpatrick(pozri Corneil).
Kishi, G., Kajitani, Y. (1969): Maximally distant trees and principal partition of a linear graph. IEEE Trans. Circuit Theory CT-16, 323—330, [7.D].
Klee, V., Larman, D. (1979): Use of Floyd's algorithm to find shortest restricted paths. Discrete optimization I. Annals of discrete mathematics 4. Amsterdam, North-Holland, 237—249, [4.D].
Klein, M. (1967): A primal method for minimal cost flows. Management Sci., 14, 205—220, [6.D].
Klingman (pozri Glover).
*Knuth, D. E. (1973): The art of computer programming, 3: Sorting and searching. Reading, Mass., Addison-Wesley, (ruský preklad: Moskva 1978), [2.A].
Knuth (pozri Garey).
*Kotzig, A. (1961): Matematické metódy v hospodárskej praxi. Bratislava, Vydavateľstvo SAV, [predhovor].
Koubek, V., Říha, A. (1981): The maximum k-flow in a network. Mathematical Foundations of Computer Science 1981. Lecture Notes in Computer Science 118. Berlin, Springer–Verlag, 389—397.
Koubek (pozri Goralčíková).
Kozlov, M. K., Tarasov, S. P., Chačijan, L. G. (1979): Polinomiaľnaja razrešimosť vypuklogo kvadratičnogo programmirovanija. Dokl. Akad. Nauk SSSR, 248, 1049—1051, [2.Z].
König, D. (1931): Graphen und Matrizen. Mat. Fiz. Lapok, 38, 116—119, [8.A, 8.B].
*König, D. (1936): Theorie der endlichen und unendlichen Graphen. Leipzig, Akademische Verlagsgesellschaft, [1.A, 1.E, 3.B, 8.Z].
Kössler (pozri Jarník).
Kruskal, J. B. (1956): On the shortest spanning subtree of a graph and the traveling salesman problem. Proc. Amer. Math. Soc., 7, 48—50, [5.B, 5.Z].
Kučera, L. (1977): Expect behaviour of graph coloring algorithms. Lecture Notes in Computer Sci. 56, Berlin, Springer-Verlag, 447—451, [2.E].
Kučera, L. (1981): Maximum flow in planar networks. Mathematical foundations of Computer Science 1981. Lecture Notes in Computer Science 118. Berlin, Springer-Verlag, 418—422, [6.B].





Kučera, L., Nešetřil, J., Pultr, A. (1980): The complexity of dimension three and some edge covering characteristics of a graph, J. Theor. Computer Sci., 11, 1—14, [1.Z].
Kučera (pozri Babai).
Kuhn, H. W. (1955): The Hungarian method for the assignment problem. Naval Res. Logist. Quart., 2, 83—97, [8.C].
Kundu, S. (1974): Bounds on the number of disjoint spanning trees. J. Combin. Theory, 17B, 199—203, [7.Z].
Kuratowski, K. (1930): Sur le problème des courbes gauches en topologie. Fund. Math., 15, 271—283, [10.C].
Kwan, M. (1962): Graphic programming using odd and even points. Chinese Math., 1, 273—277, [9.B].
Kyš, P., Wawruch, A. (1981): Point-maximal subgraphs with a given diameter, (rukopis), [2.Z].
Larman (pozri Klee).
Lawler, E. L. (1967): Optimal cycles in doubly weighted directed linear graphs. Theory of graphs. Paris, Dunod, 209—214, [4.D, 4.F].
Lawler, E. L. (1971): A solvable case of the traveling salesman problem. Math. Programming, 1, 267—269, [9.Z].
Lawler, E. L. (1972): A procedure for computing the K best solutions to discrete optimization problems and its application to the shortest path problem. Management Sci., 18, 401—405, [4.E].
*Lawler, E. L. (1976): Combinatorial optimization: networks and matroids. New York, Holt, Rinehart and Winston, [2.E, 4.D, 4.E, 4.F, 8.B, 8.C, 8.Z].
Lempel, A., Even, S., Cederbaum, I. (1967): An algorithm for planarity testing of graphs. Theory of graphs. Paris, Dunod, 215—232, [10.C].
Lenstra, J. K., Rinnooy Kan, A. H. G. (1976): On general routing problems. Networks, 6, 273—280, [9.Z].
Lenstra (pozri Johnson).
Lesniak–Foster (pozri Behzad).
Lewis (pozri Rosenkrantz).
Liebl, P., Sedláček, J. (1966): Umformung von Quadratmatrizen auf Quasitrianguläre Form mit Mitteln der Graphentheorie. Aplikace mat., 11, 1—9, [3.E].
Liu, P. C., Geldmacher, R. C. (1978): On the deletion of nonplanar edges of a graph., (rukopis), [10.Z].
Liu (pozri Chu).
Lloyd (pozri Biggs).
Lovász, L. (1973): Covering and colorings of hypergraphs. Proc. 4th Southeastern Conference on Combinatorics, Graph Theory and Computing. Winnipeg, Utilitas Mathematica Publishing, 3—12, [2.D].
Lovász, L. (1976): On two minimax theorems in graphs. J. Combin. Theory (B) 21, 96—103, [7.C].
*Lovász, L. (1979): Combinatorial problems and exercises. Budapest, Akadémiai Kiadó, [2.E, 3.Z].

Lovász, L. (1980): A new linear programming algorithm — better or worse than the simplex method? The Mathematical Intelligencer, 2, 141—146, (český preklad: Pokroky mat. fyz. astr., 26 (1981), 193—202), [2.B].
Lovász, L., Neumann-Lara, V., Plummer, M. (1978): Mengerian theorems for paths of bounded length. Periodica Math. Hung., 9, 269—276, [7.Z].
Lovász, (pozri Grötschel).
* Lucas, E. (1882): Récréations mathématiques, 1. Paris, Gauthier-Villars, [3.A].
Luks, E. M. (1982): Isomorphism of graphs of bounded valence can be tested in polynomial time. J. Computer System Sci., 25, 42—65, [11.Z].
Mader, W. (1979): Eine Reduktionsmethode für den Kantenzusammenhang in Graphen. Math. Nachr., 93, 187—204, [7.A].
Mader, W. (1979a): Connectivity and edge-connectivity in finite graphs. Surveys in combinatorics. Cambridge, Cambridge University Press, 66—95, [7.A].
Martynjuk, V. V. (1962): Ekonomnoje postrojenije tranzitivnogo zamykanija binarnogo otnošenija. Žurnal vyčisl. matem. i matem. fiziki, 2, 723—725, [3.G].
Matijasevič, J. B. (1970): Diofantovosť prečislimych množestv. Dokl. Akad. Nauk. SSSR, 191, 279—282, [2.A].
Matsuyama (pozri Takahashi).
Matula, D. W. (1978): Subtree isomorphism in $O(n^{5/2})$. Algorithmic aspects of combinatorics. Annals of discrete mathematics 2. Amsterdam, North—Holland, 91—106, [11.B].
Mayer (pozri Filotti).
Megiddo (pozri v [Garey, Johnson (1979)]), [4.D].
Meľnik (pozri Jermoliev).
Mendelsohn, N. S., Dulmage, A. L. (1958): Some generalizations of the problem of distinct representatives. Canad. J. Math., 10, 230—241, [8.Z].
Menger, K. (1927): Zur allgemeinen Kurventheorie. Fund. Math., 10, 96—115, [7.A].
Meyer (pozri Fischer).
Miller, G. L. (1980): Isomorphism testing for graphs of bounded genus. Proc. Twelfth Annual ACM Symposium Th. Computing, 225—235, [11.D].
Miller (pozri Filotti).
Minieka, E. (1973): Maximal, dynamic and lexicographic flows. Operations Res., 21, 517—527, [6.C].
* Minieka, E. (1978): Optimization algorithms for networks and graphs. New York, Marcel Dekker, (ruský preklad: Moskva 1981), [2.Z, 8.Z].
Mitchell, S. L. (1979): Linear algorithms to recognize outerplanar and maximal outerplanar graphs. Inf. Proc. Letters, 9, 229—232, [10.Z].
* Moon, J. W. (1970): Counting labelled trees. Montreal, Canadian Mathematical Congress, [5.Z].
Moore, E. F. (1957): The shortest path through a maze. Proc. Int. Symp. on Theory of Switching 1957: Annals of the Computation Lab. of Harvard University 30, Harvard U. Press, Cambridge, 1959, 285—292, [4.B].
Morávek, J. (1969): On the complexity of discrete programming problems. Aplikace mat., 14, 442—474, [2.A].





Munro, I. (1971): Efficient determination of the transitive closure of a directed graph. Inform. Proc. Letters, 1, 56—58, [3.G].
Murty (pozri Bondy).
Nash-Williams, C. St. J. A. (1960): On orientations, connectivity and odd-vertex-pairings in finite graphs. Can. J. Math., 12, 555—567, [7.Z].
Nash-Williams, C. St. J. A. (1961): Edge-disjoint spanning trees of finite graphs. J. London Math. Soc., 36, 445—450, [7.D].
Nash-Williams, C. St. J. A. (1964): Decomposition of finite graphs into forests. J. London Math. Soc., 39, 12, [7.D].
* Nebeský, L. (1969): Algebraic properties of trees. Praha, Acta Univ. Carolinae Monograph, [predhovor].
* Nečas, J. (1978): Grafy a jejich použití. Praha, SNTL, [predhovor].
Nemhauser (pozri Cornuejols, Garfinkel).
* Nešetřil, J. (1979): Teorie grafů. Praha, SNTL, [predhovor, 2.E, 5.Z].
Nešetřil, J., Poljak, S. (1980): Geometrické a algebraické souvislosti kombinatorické optimalizace. Zborník SOFSEM '80. Výskumné výpočtové stredisko, Bratislava, 35—77, [2.B].
Nešetřil, J., Pultr, A. (1977): A Dushnik-Miller type dimension of graphs and its complexity. Lecture Notes in Computer Sci. 56. Berlin, Springer-Verlag, 482—493, [1.Z].
Nešetřil (pozri Kučera).
Neumann-Lara (pozri Lovász).
Niepel, Ľ. (1980): On decomposition of complete graphs into factors with given diameters and radii. Math. Slovaca, 30, 3—11, [7.Z].
Niepel, Ľ., Tomasta,P. (1981): Elevation of a graph. Czech. Math. J., 31(106), 475—483, [3.Z].
Nievergelt (pozri Reingold).
* Noltemeier, H. (1976): Graphentheorie mit Algorithmen und Anwendungen. Berlin, Walter de Gruyter, [4.B].
Norman, R. Z., Rabin, M. (1959): Algorithm for a minimal cover of a graph. Proc. Amer. Math. Soc., 10, 315—319, [8.Z].
Norman (pozri Harary).
Nožička (pozri Bílý, Fiedler).
Onaga, K. (1967): Optimal flows in general communication networks, J. Franklin Inst., 283, 308—327, [6.Z].
Orden, A. (1956): The transshipment problem. Management Sci., 3, 276—285, [6.D].
Orloff, C. S. (1974): A fundamental problem in vehicle routing. Networks, 4, 35—64, [9.Z].
Palumbíny, D., Znám, Š. (1973): On decompositions of complete graphs into factors with given radii. Mat. časop., 23, 306—316, [7.Z].
Pan (pozri v [Tarjan (1980)]), [3.G].
Papadimitriou, C. H. (1976): On the complexity of edge traversing. J. ACM, 23, 544—554, [9.B].
Parter (pozri Auslander).
Patterson (pozri Davis).

Perl (pozri Itai).
Petersen, J. (1891): Die Theorie der regulären Graphs. Acta Math., 15, 193—220, [8.A, 8.Z].
Plesník, J. (1975): Critical graphs of given diameter. Acta Fac. R. N. Univ. Comen. Math., 30, 71—93, [7.Z].
Plesník, J. (1976): Minimum block containing a given graph. Archiv Math. (Basel), 27, 668—672, [3.C].
Plesník, J. (1979): The NP-completeness of the hamiltonian cycle problem in planar digraphs with degree bound two. Inf. Processing Letters, 8, 199—201, [2.D].
Plesník, J. (1980): Coloring of graphs by partitioning. Math. Slovaca, 30, 121—126, [2.Z].
Plesník, J. (1980a): On the computational complexity of centers locating in a graph. Aplikace mat., 25, 445—452, [2.Z].
Plesník, J. (1981): A bound for the Steiner tree problem in graphs. Math. Slovaca, 31, 155—163, [5. G].
Plesník, J. (1981a): The complexity of designing a network with minimum diameter. Networks, 11, 77—85, [5.Z].
Plesník, J. (1981b): A note on the complexity of finding regular subgraphs, (rukopis), [8.Z].
Plesník, J. (1982a): Complexity of decomposing graphs into factors with given diameters or radii, Math. Slovaca, 32, 379—388, [7.Z].
Plesník, J. (1982): Equivalence between the minimum covering problem and the maximum matching problem, (rukopis), [8.Z].
Plesník, J., Wawruch, A. (1981): On the complexity of finding degree constrained subgraphs, Acta Math. Univ. Comen., 40—41, 215—218, [8.D].
Plummer (pozri Lovász).
Poljak, S., Rödl, V., Turzík, D. (1981): Set representation of graphs and its complexity. Discrete Appl. Math., (v tlači), [1.Z].
Poljak (pozri Nešetřil).
Ponstein, J. (1972): On the maximal flow problem with real arc capacities. Math. Programming, 3, 254—256, [6.A].
Pósa, L. (1976): Hamiltonian circuits in random graphs. Discrete Math., 14, 359—364, [2.E].
Prim, R. C. (1957): Shortest connection networks and some generalizations. Bell System Tech. J., 36, 1389—1401, [5.B].
Pulleyblank, W. R. (1979): A note on graphs spanned by Eulerian graphs. J. Graph Theory, 3, 309—310, [9.Z].
Pultr (pozri Kučera, Nešetřil).
Rabin (pozri Norman).
Rao (pozri Dantzig).
Read, R. C., Corneil, D. G. (1977): The graph isomorphism disease. J. Graph Theory, 1, 339—363, [11.Z].
Reckhow (pozri Cook).
Rédei, L. (1934): Ein kombinatorischer Satz. Acta Litt. Szeged, 7, 39—43, [1.A, 1.Z].
Reif (pozri Filotti).





*Reingold, E. M., Nievergelt, J., Deo, N. (1977): Combinatorial algorithms: Theory and practice. Englewood Cliffs, New Jersey, Prentice-Hall, Inc., (ruský preklad: Moskva 1980), [2.E].
Richardson, M. (1946): On weakly ordered systems. Bull. Amer. Math. Soc., 52, 113—116, [3.Z].
Rinnooy Kan (pozri Johnson, Lenstra).
Robbins, H. E. (1939): A theorem on graphs with an application to a problem of traffic control. Amer. Math. Monthly, 46, 281—283, [3.Z].
Rosa (pozri Bosák).
Rosenkrantz, D. J., Stearns, R. E., Lewis, P. M. (1977): An analysis of several heuristics for the traveling salesman problem. SIAM J. Comput., 6, 563—581, [9.C, 9.Z].
Rosenthal, A., Goldner, A. (1977): Smallest augmentations to biconnect a graph. SIAM J. Comput., 6, 55—66, [3.C].
Rotschild, B., Whinston, A. (1966): On two-commodity network flows. J. ORSA, 14, 1121↛1129, [6.E].
Rödl (pozri Poljak).
Říha (pozri Koubek).
Saaty (pozri Busacker).
Sahni, S. (1974): Computationally related problems. SIAM J. Comput., 3, 262—279, [2.Z, 6.F, 6.Z].
Sahni, S., Gonzalez, T. (1976): P-complete approximation problems. J. ACM, 23, 555—565, [2.C, 2Z, 9.C].
*Sachs, H. (1970): Einführung in die Theorie der endlichen Graphen. Teil I. Leipzig, Teubner.
*Sachs, H. (1972): Einführung in die Theorie der endlichen Graphen. Teil II. Leipzig, Teubner.
Sakarovitch, M. (1973): Two-commodity network flows and linear programming. Math. Programming, 5, 1—20, [6.E].
Sedláček, J. (1957): O konečných orientovaných grafech. Časop. Pěst. Mat., 82, 195—215, [3.B].
Sedláček, J. (1959): O jednom typu dobře orientovaných grafů. Časop. Pěst. Mat., 84, 7—15, [3.Z].
*Sedláček, J. (1964): Kombinatorika v teorii a praxi (Úvod do teorie grafů). Praha, Nakladatelství ČSAV, [predhovor].
*Sedláček, J. (1977, 1981): Úvod do teorie grafů. Praha, Academia, [predhovor, 1.B].
Sedláček (pozri Liebl).
Sekanina, M. (1960): On an ordering of the set of vertices of a connected graph. Spisy Přírod. Fak. Univ. Brno, 137—141, [9.Z].
Shamir (pozri Even).
Shiloach, Y. (1980): A polynomial solution to the undirected two path problem. J. ACM, 27, 445—456, [6.Z].
Shiloach (pozri Itai).

Schnorr, C. P. (1977): An algorithm for transitive closure with linear expected time. Fachbereich Mathematik, Universität Frankfurt, [3.G].
Schrijver (pozri Grötschel).
Slater, P. J. (1975): Leaves of trees. Proc. 6th S—E conf. combinatorics, graph theory and computing: Congressus Numerantium 17. Winnipeg, Utilitas Math., 549—559, [3.C].
Sleator, D., Tarjan, R. E. (1980): An O(n m log n) algorithm for maximum network flows. Twenty-first Ann. Symp. on Found. Computer Sci., [6.B].
Spencer (pozri Erdös).
Stadel, M. (1978): A remark on the time complexity of the subtree problem. Computing, 19, 297—302, [11.B].
Stearns (pozri Rosenkrantz).
Steinitz, E. (1922): Polyeder und Raumenteilungen. Encyklopädie der Mathematischen Wissenschaften IIIAB12, 1—139, [10.B].
Stewart (pozri Golden).
Stockmeyer (pozri Garey).
Strassen, V. (1969): Gaussian elimination is not optimal. Numerische Math., 11, 354—356, [3.G].
Sylvester, J. J. (1978): Chemistry and algebra. Nature, 17, 284, [1.A].
Sysło, M. M. (1973): A new solvable case of the traveling salesman problem. Math. Programming, 4, 347—348, [9.Z].
Širáň, J. (1981): Additivity of the crossing number of graphs with connectivity 2., (rukopis), [10.Z].
*Taha, H. A. (1975): Integer programming. New York, Academic Press, [2.B, 2.C].
Takahashi, H., Matsuyama, A. (1980): An approximate solution for the Steiner problem in graphs. Math. Japonica, 24, 573—577, [5.G].
Tarasov (pozri Kozlov).
Tarjan, R. E. (1972): Depth-first search and linear graph algorithms. SIAM J., Comput., 1, 146—160, [3.A, 3.B, 3.D, 3.E].
Tarjan, R. E. (1974): A good algorithm for edge-disjoint branchings. Inf. Processing Letters, 3, 51—53, [7.C].
Tarjan, R. E. (1975): Efficiency of a good but not linear set union algorithm. J. Assoc. Comput. Mach., 22, 215—225, [5.B].
Tarjan, R. E. (1976): Edge-disjoint spanning trees and depth-first search. Acta Informatica, 6, 171—185, [7.C, 7.D].
Tarjan, R. E. (1977): Finding optimum branchings. Networks, 7, 25—35, [5.F].
Tarjan, R. E. (1980): Recent developments in the complexity of combinatorial algorithms. Fifth IBM Symposium on Foundations of Computer Science (Computational Complexity), Tokyo, [11.Z].
Tarjan (pozri Eswaran, Even, Garey, Hopcroft, Cheriton, Sleator).
Tarjan (pozri v [Karp (1972)]), [2.D].
Tarry, G. (1895): Le problème des labyrinthes. Nouvelles Ann. Math., 14, 187—190, [3.A].
Thomassen (pozri Chvátal).





Toman, E. (1979): Geometričeskoje strojenije slučajnych bulevych funkcij. Problemy kibernetiki, 35, 111—132, [1.Z].
Tomasta (pozri Niepel).
Tomová, E. (1977): Decomposition of complete bipartite graphs into factors with given diameters. Math. Slovaca, 27, 231—237, [7.Z].
*Trávnik, I., Vlach, J. (1975): Sieťová analýza. Bratislava, Alfa, [4.A].
Trémaux (pozri v [Lucas (1882)]), [3.A].
Treybing (pozri Borosh).
Turing, A. M. (1936): On computable numbers, with an application to the Entscheidungs problem. Proc. London Math. Soc. (ser. 2), 42, 230—265, [2.A].
Turzík (pozri Poljak).
Tutte, W. T. (1947): The factorization of linear graphs. J. London Math. Soc., 22, 107—111, [8.A].
Tutte, W. T. (1954): A short proof of the factor theorem for finite graphs. Canad. J. Math., 6, 347—352, [8.D].
Tutte, W. T. (1961): On the problem of decomposing a graph into n connected factors. J. London Mat., Soc., 142, 221—230, [7.D].
Tutte, W. T. (1963): How to draw a graph. Proc. London Math. Soc., 13, 743 – 767, [10.C].
Ullman (pozri Aho).
Van Slyke (pozri Kerschenbaum).
Vlach (pozri Trávnik).
Wagner, K. (1936): Bemerkungen zum Vierfarbenproblem. Jahresber. Deutsch. Math.-Verein., 46, 26—32, [10.B].
Wagner, R. A. (1976): A shortest path algorithm for edge-sparse graphs. J. ACM, 23, 50—57, [4.B].
Wagner, R. A. (pozri Dreyfus).
*Walther, H. (1979): Anwendungen der Graphentheorie. Berlin, VEB Deutscher Verlag der Wissenschaften.
Warshall, D. (1962): A theorem on Boolean matrices. J. ACM, 9, 11—12, [3.G, 4.C].
Wawruch (pozri Kyš, Plesník).
Weinberg, L. (1966): A simple and efficient algorithm for determining isomorphism of planar triply connected graphs. IEEE Trans. on Circuit Theory CT-13, 142—148, [11.C, 11D].
Whinston (pozri Rotschild).
Whitney, H. (1932): Congruent graphs and the connectivity of graphs. Amer. J. Math., 54, 158—168, [7.A].
Whitney, H. (1932a): Non-separable and planar graphs. Trans. Amer. Math. Soc., 34, 339—362, [10.C].
Whitney, H. (1933): A set of topological invariants for graphs. Amer. J. Math., 55, 231—235, [10.B].
Wilson (pozri Biggs).
Wong (pozri Hopcroft).

Wyllie (pozri Fortune).
Yao, A. C. (1975): An O(|E| log log |V|) algorithm for finding minimum spanning trees. Inf. Processing Letters, 4, 21—23, [5.B].
Yen, J. Y. (1971): Finding the K shortest loopless paths in a network. Manag. Sci., 17, 712—716, [4.E].
Yen, J. Y. (1972): Finding the lengths of all shortest paths in N-node nonnegative-distance complete networks using $\frac{1}{2}N^3$ additions and $N^3$ comparisons. J. ACM, 19, 423—424, [4.Z].
Zadeh, N. (1972): Theoretical efficiency of the Edmonds–Karp algorithm for computing maximal flows. J. ACM, 19, 184—192, [6.B].
Zadeh, N. (1973): A bad network problem for the simplex method and other minimum cost flow algorithms. Math. Programming, 5, 255—266, [2.B, 6.D].
Zajcev (pozri Dinic).
Zelinka, B. (1975): On certain distance between isomorphism classes of graphs. Časop. pěst. mat., 100, 371—373, [11.Z].
*Zelinka, B., 1977: Rovinné grafy. Praha, Mladá fronta, [predhovor].
Znám (pozri Bosák, Palumbíny).
*Zykov, A. A. (1969): Teorija konečnych grafov. Novosibirsk, Nauka.
Zykov, A. A. (1974): Gipergrafy. Uspechi mat. nauk, 29, 89—154, [2.E].



# PREHĽAD OZNAČENÍ

| | |
|---|---|
| $x := y$ | definitorická rovnosť (symbol $x$ dostáva hodnotu symbolu $y$) |
| $\lfloor x \rfloor$ | dolná celá časť čísla $x$ (najväčšie celé číslo nepresahujúce $x$) |
| $\lceil x \rceil$ | horná celá časť čísla $x$ (najmenšie celé číslo nie menšie ako $x$) |
| $A \cup B$ | zjednotenie množín |
| $A \cap B$ | prienik množín |
| $A - B$ | rozdiel množín |
| $\bar{A}$ | komplement množiny $A$ do základnej množiny (často $\bar{A} = V_G - A$) |
| $A \subseteq B$ | $A$ je podmnožinou množiny $B$ |
| $A \subset B$ | $A$ je vlastnou podmnožinou množiny $B$ |
| $x \in A$ | $x$ patrí do $A$ |
| $x \notin A$ | $x$ nepatrí do $A$ |
| $\|A\|$ | mohutnosť (počet prvkov) množiny $A$, absolútna hodnota čísla $A$ |
| $\{x \mid P\}$ | množina všetkých prvkov $x$ s vlastnosťou $P$ |
| $\varphi: A \to B$ | zobrazenie (funkcia) množiny $A$ do množiny $B$ |
| $\varphi^{-1}$ | inverzné zobrazenie k $\varphi$ |
| $(x, y)$ | usporiadaná dvojica, šíp $xy$ |
| $A \times B$ | $\{(x, y) \mid x \in A, y \in B\}$ |
| $(x_1, x_2, \ldots, x_n)$ | usporiadaná $n$-tica |
| $A \times B \times C$ | $\{(x, y, z) \mid x \in A, y \in B, z \in C\}$ |
| $[x, y]$ | neusporiadaná dvojica, rebro $xy$ |
| $A \circ B$ | $\{[x, y] \mid x \in A, y \in B\}$ |
| $xy$ | súčin čísel $x$ a $y$, rebro s krajmi $x$ a $y$, šíp orientovaný z $x$ do $y$ |
| $x \vee y$ | disjunkcia (výrokov v logike) |
| $x \wedge y$ | konjunkcia |
| $\bar{x}$ | negácia |
| $x \Rightarrow y$ | implikácia |
| $x \Leftrightarrow y$ | ekvivalencia |
| $H \subseteq G$ | $H$ je podgrafom grafu $G$ |
| $H \subset G$ | $H$ je vlastným podgrafom grafu $G$ |
| $H \cong G$ | $H$ a $G$ sú izomorfné |
| $G(S)$ | vrcholovo-indukovaný podgraf na $S$ |

| | |
|---|---|
| $G\langle S\rangle$ | obal (hranovo indukovaný podgraf) na $S$ |
| $G-S$ | maximálny podgraf grafu $G$ neobsahujúci $S$ |
| $\bar{G}$ | komplementárny graf ku $G$ |
| $K_n$ | kompletný graf s $n$ vrcholmi |
| $V(G)$, $V_G$ | vrcholová množina grafu $G$ |
| $E(G)$, $E_G$ | hranová množina grafu $G$ |
| $n(G)$, $n_G$, $n$ | počet vrcholov grafu $G$ |
| $m(G)$, $m_G$, $m$ | počet hrán grafu $G$ |
| $V(v)$ | (presnejšie: $V_G(v)$) vrcholové okolie vrcholu $v$ (množina okolitých vrcholov) |
| $V^-(v)$ | prichádzajúce vrcholové okolie |
| $V^+(v)$ | odchádzajúce vrcholové okolie |
| $V^0(v)$ | neutrálne vrcholové okolie |
| $E(v)$ | (presnejšie: $E_G(v)$) hranové okolie vrcholu $v$ (množina hrán incidentných s vrcholom $v$) |
| $E^-(v)$ | prichádzajúce šípové okolie |
| $E^+(v)$ | odchádzajúce šípové okolie |
| $E^0(v)$ | rebrové okolie |
| $\deg(v)$ | (presnejšie: $\deg_G(v)$) (celkový) stupeň vrcholu $v$ |
| $\deg^-(v)$ | prichádzajúci stupeň |
| $\deg^+(v)$ | odchádzajúci stupeň |
| $\deg^0(v)$ | rebrový stupeň |
| $S^{-1}$ | obrátený sled k sledu $S$ |
| $d(u, v)$ | (presnejšie: $d_G(u, v)$) vzdialenosť vrcholov $u$ a $v$ |
| $\mathrm{ex}(u)$ | (presnejšie: $\mathrm{ex}_G(u)$) excentricita vrcholu $u$ |
| $\mathrm{ex}^-(u)$ | prichádzajúca excentricita |
| $\mathrm{ex}^+(u)$ | odchádzajúca excentricita |
| $\mathrm{diam}(G)$ | diameter |
| $\mathrm{rad}(G)$ | rádius |
| $f(x)$ | veľkosť toku $x$ |
| $f_s^+(x)$ | čistý odtok toku $x$ z $s$ |
| $f_t^-(x)$ | čistý prítok toku $x$ do $t$ |
| $(A, B)_G$ | hranový rez digrafu $G$ |
| $[A, B]_G$ | hranový rez grafu $G$ |
| $\lambda(G)$ | hranová súvislosť grafu $G$ |
| $\varkappa(G)$ | vrcholová súvislosť grafu $G$ |
| $\lambda(u, v)$ | mohutnosť minimového hranového $u$—$v$ rezu |
| $\varkappa(u, v)$ | mohutnosť minimového vrcholového $u$—$v$ rezu |
| $\chi(G)$ | chromatické číslo (farebnosť) grafu $G$ |



# АЛГОРИТМЫ НА ГРАФАХ

*Резюме*

В книге излагаются основные алгоритмы на графах. Большинство из них изложено детально и без требования на специальную математическую подготовку, но определенная математическая зрелость необходима. Материал делится на 11 глав, каждая из которых сопровождается упражнениями и дополнительными замечаниями.

Вводная глава 1 содержит основные понятия теории графов. Глава 2 содержит неформальное введение в алгоритмы. Понятие полиномиального преобразования здесь иллюстрируется многими примерами включая задачу линейного программирования. Затем вводятся NP-трудные и NP-эквивалентные задачи и показывается несколько основных NP-трудных задач, например задача нахождения гамильтонова цикла.

В главе 3 описываются методы поиска в неориентированном графе или лабиринте и затем расширяются для нахождения связных компонент, блоков, точек сочленения, листов и мостов. Рассматривается проблема наименьшего блока или листа содержащего данный граф. Аналогичные методы предложены и для ориентированных графов. Кроме того рассмотрены транзитивное замыкание и редукция. В главе 4 находятся алгоритмы для оптимальных путей в взвешенных графах. Особенно изучаются алгоритмы для кратчайший путей. Тоже рассмотрены кратчайшие пути с ограничениями и все кратчайшие пути.

Деревья и главным образом остовные деревья изучаются в главе 5. Кроме классической задачи минимального остовного дерева, рассматриваются некоторые обобщения: минимальное остовное дерево с заданной степенью одной вершины, или остовное дерево с минимальным отношением двух весов. Тоже дан алгоритм для минимального иерархического дерева. Задача дерева Штейнера решается точным и приближенным алгоритмом.

Глава 6 имеет дело с потоками и начинает классическим алгоритмом Форда а Фалкерсона. Затем даны полиномиальные алгоритмы для максимального потока. Глава продолжается с вариантами основной задачи. Классическая транспортная задача, проблема потока минимальной стоимости с одним приложением и неориентированная двухпродуктовая задача, тоже включаются. В конце главы некоторые теоретические обсуждения других потоковых задач. Две части главы 7 содержат специальные потоковые вопросы: вершинная или реберная связность. Многополюсные задачи анализа и синтеза основаны на Гомори и Ху. Как мера связности графа, выступает и максимальное число непересекающихся остовных деревев.

Паросочетания обсуждаются в главе 8. Употребление чередующихся цепей демонстрировано на двух алгоритмах Эдмондса (паросочетание максимальной мощности и паросочетание максимальной цены). В одной части этой главы исследуется проблема

нахождения подграфа с заданными степеньями вершин. Глава 9 включает проблему эйлеровых циклов, китайского почтальона и задачу коммивояжера.

Последние две главы относятся к плоским графам: Глава 10 содержит некоторые классические теоремы о плоских графах и в общих чертах один алгоритм для расположения графа в плоскости. Изоморфное вхождение одного дерева в другое и изоморфизм плоских графов изучаются в главе 11.

# GRAPH ALGORITHMS

*Summary*

The book provides main graph algorithms. Most of them are discussed in details and without prerequisities to a special mathematical backround, but certain mathematical maturity is necessary. The material is divided into eleven chapters each of which is supplemented by exercises and additional remarks.

The introductory Chapter 1 contains basic graph definitions. Chapter 2 contains an informal introduction to algorithms. The notion of polynomial transformation is illustrated by several examples including the problem of linear programming. Then NP-hard and NP-equivalent problems are introduced and some basic problems are shown to be NP-hard, for example the hamiltonian cycle problem.

In Chapter 3 methods of exploring an undirected graph or a maze are described and these are also extended for finding components, blocks, cutvertices, leaves and bridges. The problem of smallest augmentation of a given graph to a block or a leaf is studied. Analogous methods are developed for directed graphs. In addition, the transitive closure and reduction are considered. Chapter 4 deals with optimal paths in weighted graphs. Especially shortest--path algorithms are presented. Constrained shortest paths and all shortest paths can be found there.

Trees and mainly spanning trees are studied in Chapter 5. In addition to the classical minimum spanning tree problem, some generalizations are considered: a minimum spanning tree with prescribed degree of a vertex, or a minimum-ratio spanning tree (each edge has two weights). An algorithm for a minimal directed spanning tree is also given. The problem of finding Steiner minimum tree is studied and an approximation algorithm, in addition to an exact method, is presented.

Chapter 6 deals with flows and begins with the classical algorithm of Ford and Fulkerson. Then some polynomial algorithms for a maximal flow are shown. The chapter continues with variations of the basic problem. The classical transportation problem, the problem of a minimum cost flow with an application, and the two-commodity (undirected) flow problem are included too. Finally, some theoretical considerations about other flow problems are presented. Two parts of Chapter 7 contain so special flow problems as finding the vertex, or edge connectivities of a graph. The multiterminal analysis and synthesis are based on Gomory and Hu. As a measure of connectivity of a graph, the maximal number of disjoint spanning trees is used.

Matchings are discussed in Chapter 8. The usefulness of alternating paths is demonstrated by two Edmonds' algorithms (one for the cardinality matching problem and the second for the weighted matching problem). A part of the chapter is devoted to spanning subgraphs

with prescribed degrees. Chapter 9 includes eulerian trails, the Chinese postman's problem, and the traveling salesman problem.

Two last chapters deal with planar graphs. Chapter 10 contains some classical theorems about planar graphs and an outline of a planarity algorithm. The subtree problem and the isomorphism problem for planar graphs are treated in Chapter 11.

# REGISTER


algoritmus 40
—, aproximačný 53
—, Dijkstrov 117
—, Edmondsov 243, 249
—, Fordov 119
— Fordov a Fulkersonov 178
—, heuristický 153
— Hopcroftov a Tarjanov pre planaritu 289
— labytrintový 262
—, Moorov 116
— najširších polociest 182
— polocyklov a polociest 193
—, polynomiálny 43
—, štiepiaci 271
—, Warshallov 102
— zjemňovania stupnice 183
analýza grafu 224
— projektu 113
aproximácia 53
artikulácia 79
automorfizmus 22
—, silný 300

báza kvetu 245
blok 79
—, izolovaný 81
—, koncový 81

centrum grafu 39
— stromu 39
(*p*—centrum) 70
cesta 24
—, alternujúca 240
—, hamiltonovská 38, 61, 91
—, kritická 136
—, najdlhšia 115
—, najkratšia 111
—, v grafe 124
—, najširšia 131
— s najväčšou očakávanou šírkou 132
—, zväčšujúca 240
cesty, hranovo disjunktné 220
—, *M* najkratších 129
—, vnútorne disjunktné 220
cirkulácia 193, 216
CPM 114, 136
cyklus 24
(p-cyklus) 24
—, hamiltonovský 38, 273
—, — najkratší 273
— s maximálnym ziskom 133
—, záporný 123

činnosť, kritická 136
číslo, cyklomatické 138
—, chromatické 58

dezorientácia pseudomigrafu 22, 108
diagram 20
diagram, duálny 294
—, rovinný 20, 281
—, sférický 284
—, úsečkový 286
diagramy, ekvivalentné 284
—, izomorfné 300
—, silne ekvivalentné 284
—, — izomorfné 300
digraf 18
—, acyklický 31
—, hranový 279
—, jednostranne súvislý 91

—, kompletný 30
— projektu 114
—, silne súvislý 91
—, slabo súvislý 91
—, súvislý 91
diameter (= priemer) 28
dĺžka hranice steny 282
— polosledu 24, 34
— sledu 24, 34
doplnenie grafu na blok 88
— — — list 86
— digrafu na silne súvislý 99
dvojartikulácia, koncová 307

elevácia 109
excentricita 27
—, odchádzajúca 27
—, prichádzajúca 27

faktor 22
—, lineárny (1-faktor) 256
—, pravidelný 256
— s predpísanými stupňami 256
farbenie vrcholov grafu 58
— — hypergrafu 57
farebnosť grafu 58
forma, konjunktívna normálna 50
funkcia, účelová 45, 53

graf 18
—, bipartitný 30
—, blokovo-artikulačný 81
—, eulerovský 261
—, hamiltonovský 273
—, komplementárny 30
—, kompletný 30
—, *k*-partitný 30
—, kubický 29
—, kvadratický 29
—, lineárny 29
—, nekonečný 23
—, nesúvislý 27
—, netriviálny 30
—, nulový 30
—, ohodnotený 34
—, Petersenov 294
—, planárny 20, 281
— —, maximálny 283
—, pravidelný (stupňa k) 29
—, prázdny 30
—, prienikový 39
—, sieťový 114
—, súvislý 26
—, triviálny 30
— vnoriteľný do guľovej plochy 284
—, vonkajškovo-planárny 294
grafy homeomorfné 288
—, izomorfné 22

heuristika 54
hodnota toku 174
hrana 18
— incidentná s vrcholom 21
—, krížová 95
—, nasýtená 175
—, neorientovaná 21
—, nepáriaca 240
—, odchádzajúca 21
—, orientovaná 21
—, páriaca 240
—, priama 95
—, prichádzajúca 21
—, spätná 76, 95
—, stromová 76, 95
—, tranzitívna 102
hrany, násobné 18, 21
—, nezávislé 240
— polosledu, nesúhlasné 24
— —, súhlasné 24
—, rovnobežné (= paralelné) 21
—, susedné 21
hypergraf 57

chorda 76, 137, 289





izomorfizmus 22, 296, 300
—, silný 300

jadro 109

kapacita hrany 173
— rezu 176
— vrcholu 186
kapacity šípov (najlacnejšie zväčšenie) 201
klauzula 51
kocka 39
komplement grafu 30
komponent pseudomigrafu 26
(3-komponent) 306
(—, koncový) 307
— —, izolovaný 88
— digrafu, jednostranný 92
— —, silný 92
— —, slabý 92
koniec hrany 21
kondenzácia, listová 86
—, digrafu 92
koreň 77, 96, 245
kostra 27, 137
—, najlacnejšia 139
—, —, zdrojová 157
— s minimálnym pomerom dvoch ohodnotení 150
—, susedná 140
kostry, disjunktné 235
—, — zdrojové 232
kraj hrany 21
— polosledu 24
krivka, nákladová, projektu 218
kúsok 291
kvet 245

labyrint 72
les 31
—, maďarský 247
list 79
—, izolovaný 86
—, koncový 86
matica incidenčná 37
—, susedská 35
matroidy 69
medián (p-medián) 70
medza na tok, dolná 186
metóda bisekcie 48, 134
— dynamického programovania 67
—, Edmonosova 243, 251
—, elipsoidová 46
—, Christofidesova 277
—, maďarská 200, 249
—, pažravá 68
—, simplexová 45, 196
— vetvenia a ohraničovania 67
—, zdvojenia kostry 275
metódy, pravdepodobnostné 68
migraf 18
mnohosten (= polyéder) 286
—, pravidelný 294
množina, absorbujúca 109
—, dominujúca 70
—, nezávislá 59, 109
multidigraf 18
multigraf 18
multimigraf 18
most

nasledovník 21
násobnosť dvojice 21
— hrany 21
neslučka 21

obal 22
očíslovanie vrcholov digrafu, monotónne 97
odtok, čistý 174
ohodnotenie pseudomigrafu 34
ohraničenia, kapacitné 173
okolie, hranové 21
—, neutrálne vrcholové 21
—, odchádzajúce šípové 21
—, — vrcholové 21
—, prichádzajúce šípové 21
—, — vrcholové 21

—, rebrové 21
—, vrcholové 21
orientácia pseudomigrafu 22, 108, 239
otec 76

palma 76
párenie 240
—, maximálne 240
—, najcennejšie 240
—, najpočetnejšie 240
—, úplné 240
—, 3-rozmerné 55
partia vrcholov 30
PERT 114, 136
plocha rodu *g*
počítač, nedeterministický 50
— RAM 42
—, Turingov 41
podgraf 22
—, hranovo-indukovaný 22
—, indukovaný (vrcholovo-indukovaný) 22
—, maximový kompletný 59
—, kubický 260
—, steinerovský 164
—, vlastný 22
podgrafy, hranovo disjunktné 22
—, (vrcholovo) disjunktné 22
podmienky kontinuity 173
— ortogonality 45
podstrom, izomorfný 298
podkrytie, hranové (vrcholov) 260
—, presné 3-množinami 57
—, vrcholové (hrán) 59, 243, 260
polocesta 24
—, rezervná 176
—, zväčšujúca 176
polocyklus 24
(*p*-polocyklus) 26
polocykly, nezávislé 137
polomer (= rádius) 28
polosled 24
—, neorientovaný 24
—, netriviálny 24
—, obrátený 24
—, orientovaný 24
—, otvorený 24
—, triviálny 24
—, uzavretý 24
polosledy, nezávislé 137
poloťah 24
polyéder 286
postup do hĺbky 76, 95
— — šírky 77
prameň 29
predchodca 21
priemer (= diameter) 28
priepustnosť hrany 173
priesečník 20, 294
prieskum digrafov 93
— grafov 72
—, Tarryho 73, 93
—, Trémauxov 75, 95
princíp optimality 67
problém, algotitmicky nerozhodnuteľný 41
— mostov 12
— multicentra 70
— multimediánu 70
—, NP-ekvivalentný 50
—, NP-ťažký 50
—, NP-úplný 50
—, optimalizačný 50, 53
—, priraďovací 249
— pochôdzky, všeobecný 280
—, rozhodovací 50
— splniteľnosti 50
—, Steinerov 163
— štyroch farieb 14
problémy, polynomiálne ekvivalentné 44
programovanie, dynamické 67
protišípy (dvojica) 18
prút 88
prvok pseudomigrafu 21
prvky, incidentné 21
pseudodigraf 18
—, rovnovážny (= rovnovážne orientovaný) 29





pseudograf 18
pseudografy, homeomorfné 288
pseudomigraf 18
—, hranovo *k*-súvislý 222
—, kubický 29
—, kvadratický 29
—, lineárny 29
—, neohodnotený 34
—, nesúvislý 27
—, ohodnotený 34
—, pravidelný 29
—, súvislý 27
—, vrcholovo *k*-súvislý 222
pseudmigrafy, izomorfné 22
pseudovrchol 245
—, vonkajší 251

rádius (= polomer) 28
rebro 18, 21
—, slučkové 21
redukcia, polynomiálna 44
—, tranzitívna 102
rekurzia 67
reprezentant 216
rez 141, 176, 220
—, hranový 141, 176, 220
—, maximový 70
—, minimový 176, 220
—, vrcholový 220
rezy, nepretínajúce sa 224
riešenie, bázické 45
—, prípustné 45
—, optimálne 45
— rovnice, binárne 48
rovnica, polynomická 41
rozdelenie multimnožiny čísel 48
rozklad grafu na faktory 235, 239, 259
—, distančný 116
rozštiepenie množín 57
roztriedenie vrcholov acyklického digrafu 97

separátor 219
sieť 35
skelet polyédra 286
sled 24
—, eulerovský 264
—, —, najkratší 264
—, hamiltonovský 273
—, —, najkratší 273
—, hraničný 302
—, najkratší 111
—, — s ohraničeniami 124
—, najspoľahlivejší 113
slučka 21
splniteľnosť (boolovskej formy) 50, 190
3-splniteľnosť 54
stena diagramu 281
—, vnútorná 281
—, vonkajšia 281
stok 33
stopka kvetu 245
strom 27, 32
—, kanonický 296
—, neorientovaný 33
—, orientovaný 33
—, označkovaný 296
—, steinerovský 164
—, —, najlacnejší 164
—, stokový 33
—, zdrojový 33
—, —, najlacnejší 157
stupeň vrcholu 28, 38
— —, celkový 28
— —, odchádzajúci 28
— —, prichádzajúci 28
— —, rebrový 28
sused kostry 140
súvislosť digrafu 91
— —, jednostranná 91
— —, silná 91
— —, slabá 91
—, hranová 220, 221
—, vrcholová 220, 221
symbol (boolovský) 51
syn 76

syntéza grafu 229
systém kostier, najobjemnejší 235
— rozličných reprezentantov 216

šíp 18, 21
—, slučkový 21
šípy, nesúhlasne rovnobežné 18
—, rozlíšiteľné 302
šírka cesty 131, 172
— —, očakávaná 132
— grafu, pásová 110
— hrany 131, 172
štruktúra, grafová 18

teleso, platónske 294
tok 173
—, acyklický 216
—, blokujúci 175
—, celočíselný 179, 192, 195, 206
—, dvojproduktový 202, 205, 212, 213, 215
—, — s dolnými medzami 212, 215
—, homologický 209, 215
—, lexikograficky maximálny 186
—, maximálny 174
—, —, najlacnejší 192
—, minimálny 187, 189
— po rebre 189
—, prípustný 187
— s násobičmi 217
— s požiadavkami 213
—, selektívny 210
—, viacproduktový 201, 210
transformácia 44
—, polynomiálna 44
triangulácia, rovinná 283
trieda homologických hrán 209
— NP 50
triedy, základné 302
triedenie 43
trvanie činnosti 113, 136, 217
— projektu 115, 136, 218
turnaj 16
ťah 24

—, eulerovský 261

uholník (p-uholník) 26
úloha, aproximačná 53
— autožeriavu 279
— binárneho lineárneho programovania 47
— celočíselného lineárneho programovania 47
— čínskeho poštára 265
—, dopravná 195
—, —, klasická (Hitchockova) 196
—, hamiltonovská 61
— *k* čínskych poštárov 270
— kvadratického programovania 71
— lineárneho programovania 44
— — —, duálna 45
—, NP-ťažká a pod. (pozri problém) 50
— o binárnom batohu 48
— — celočíselnom batohu 48
— — maximálnom toku 173
— obchodného cestujúceho 273
—, optimalizačná 50, 53
— triedenia 43
— vidieckeho poštára 280
úroveň 97
úsek polosledu 24
— —, vlastný 24
usporiadanie (triedenie) 43
—, monotónne (topologické) 97
ústie 29
uzáver, tranzitívny 101

veľkosť toku 174
veta Königova-Hallova 242
—, Mengerova 220
— o dualite, silná 45
— — —, slabá 45
vetva 88
vnorenie grafu 284
vrchol 18
—, centrálny 28





— dosiahnuteľný (z vrcholu) 27
—, koncový (šípa, polosledu, stromu) 21, 24, 33
—, krajný 21
—, nulový 29
—, okolitý 21
—, párený 240
—, pokrytý 240
—, prechodový 173
—, rovnovážny 29
—, voľný 240
—, začiatočný (šípa, polosledu) 21, 24
vrcholy, steinerovské 164
—, susedné 21
—, súvisiace 26
—, základné 164
vstup 173, 184
výstup 173, 184
výška vrcholu 98

vzdialenosť bodov 35
— vrcholov 27, 34
vzorec Eulerov 281

začiatok hrany 21
zafarbenie mapy 14
— (2-zafarbenie) hypergrafu 57
— (3-zafarbenie) grafu 58
zápis, zátvorkový 298
zdroj 33
zložitosť algoritmu 43
— úlohy 43
zoznam hrán 37
— okolitých vrcholov 37
— *K* najlacnejších kostier 153
— *M* najkratších ciest 129
— — — sledov 127
zreťazenie polosledov 24
zväčšenie kapacít šípov, najlacnejšie 201

# GRAFOVÉ ALGORITMY

**RNDr. Ján Plesník, CSc.**

Obálku a väzbu navrhol Milan Chvíla
Zodpovedná redaktorka publikácie Katarína Kenížová
Technická redaktorka publikácie Gabriela Szabóová
Korektorka Eva Kanská

Vydanie prvé. Vydala VEDA, vydavateľstvo Slovenskej akadémie vied, v Bratislave r. 1983 ako 2408. publikáciu. Strán 344. Náklad 1500 výtlačkov. AH 21,91 (text 18,50, ilustrácie 3,41), VH 22,52. Vytlačili ZT, n. p., závod Svornosť, Bratislava.

1197/I-1973

71–023–83
509/58 03/2
Kčs 48,— I

MORAVSKÁ ZEMSKÁ
BRNO
KNIHOVNA

# GRAFOVÉ ALGORITMY

Ján Plesník

Pomocou grafových štruktúr možno jednoducho sformulovať mnohé úlohy z praxe. Takéto modely sú veľmi prehľadné a dostupné širokému okruhu záujemcov. Pritom sú grafové štruktúry exaktné matematické objekty skúmané v teórii grafov. Pre viaceré úlohy na grafových štruktúrach boli vypracované postupy — grafové algoritmy — umožňujúce ich aj prakticky vyriešiť. Pre mnohé úlohy však takéto algoritmy zatiaľ chýbajú.

Kniha podáva výsledky výskumov v uvedenej problematike. Študujú sa viaceré základné grafové pojmy a s nimi súvisiace úlohy, ako sú optimálne cesty, kostry, toky, párenia, pochôdzky, planarita a izomorfizmus. Hlavný dôraz sa kladie na princípy algoritmov a ich zdôvodnenie, čitateľ je však informovaný aj o výpočtovej zložitosti, a najmä o tzv. NP-ťažkých úlohách. Uvádzajú sa aj niektoré výsledky nealgoritmickej povahy umožňujúce lepšie pochopiť problematiku. Na tento cieľ slúži aj veľké množstvo cvičení, ktoré vedú čitateľa k samostatnej práci.

Publikácia je nenáročná na matematické znalosti čitateľa. Väčšina potrebných pojmov je v knihe vysvetlená a jej štúdium je uľahčené aj množstvom ilustrácií. Kniha je určená širokému okruhu čitateľov: matematikom, technikom, ekonómom a poslucháčom príslušných vysokých škôl.