

9.1.3 Příklad: Letecká bitva o Anglii. V letecké bitvě o Anglii v roce 1940 bojovala na straně Anglie řada pilotů různých národností. Královské letectvo mělo letadla, která vyžadovala dva piloty. Někteří piloti spolu ovšem nemohli letět pro jazykové potíže nebo pro rozdílnost výcviku. Kolik letadel (tj. dvojic pilotů) může za těchto omezení vzlétnout najednou?

Tento problém by bylo možno řešit nalezením maximálního párování v grafu, jehož vrcholy jsou piloti a jehož hrany spojují piloty, kteří mohou letět spolu.

9.1.4 Příklad zahrádkářský. Představme si ovocnou zahradu, ve které je $2n$ stromků a jedna hromada hnoje. Naším úkolem je přivést půl kolečka hnoje ke každému stromku. Abychom šetřili časem, chceme vždy vézt k některému stromku plné kolečko, polovinu jeho obsahu složit a pokračovat s poloprázdným kolečkem k jinému stromku. Kromě času chceme šetřit i silami a chceme, aby např. součet drah vykonaných s plným kolečkem byl co nejmenší.

Tuto úlohu lze převést na hledání nejlevnějšího perfektního párování v úplném grafu o $2n$ vrcholech. Každý vrchol odpovídá jednomu stromku. Cena hrany bude úměrná námaze spojené s jízdou plného kolečka k bližšímu z obou stromků, s následující jízdou poloprázdného kolečka ke druhému z nich a s návratem prázdného kolečka k hromadě hnoje. Je zřejmé, že každý stromek (vrchol) musí být zahrnut do přesně jedné jízdy, hrany odpovídající těmto jízdám musí tedy tvořit perfektní párování.

9.1.5 Cvičení. Jak byste řešili předchozí úlohu pro lichý počet stromků?

9.1.6 Příklad: Taneční. Předpokládejme, že v tanečních má každý hoch přesně k přítelkyň a každá dívka přesně k přátel. Je možno uspořádat tanec, při kterém každý hoch i dívka tančí s některým ze svých přátel? (Zde poněkud nerealisticky předpokládáme, že vztah přátelství je symetrický.)

Vztah přátelství můžeme znázornit bipartitním grafem, v němž každý vrchol má stupeň k . V tomto grafu hledáme perfektní párování. Ukážeme, že za uvedených podmínek lze existenci perfektního párování zaručit.

9.1.7 Přiřazovací úloha. Klasická formulace přiřazovací úlohy, totiž přiřazování pracovních úkolů pracovníkům, byla uvedena v 8.2.5, str. 135. Kromě řady dalších přímých aplikací se přiřazovací úloha vyskytuje i jako podúloha při řešení složitějších úloh, viz např. úloha čínského pošťáka 10.3, str. 182, a problém obchodního cestujícího 12.4.4, str. 206.

Řešením přiřazovací úlohy se budeme zabývat v 9.4, str. 171.

9.1.8 Cvičení. Úlohu o nejlevnějším maximálním párování lze převést na úlohu o nejdražším párování změnou cen hran. Navrhněte způsob, jak to udělat.

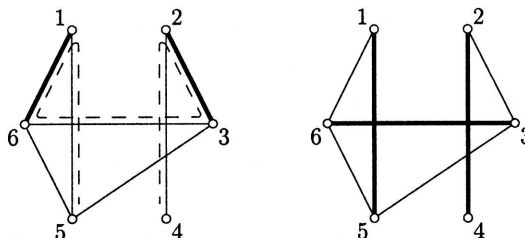
9.1.9 Poznámka. Úlohy o párování můžeme chápat jako hledání soustavy disjunktních dvojic vrcholů. Podobné úlohy týkající se disjunktních trojic jsou podstatně obtížnější (NP-těžké). Znamená to např., že kdybychom v zahrádkářské úloze

9.1.4 snížili dávku hnoje na třetinu kolečka, byl by rozvoz hnoje snazší, ale najít optimální řešení by bylo složitější.

9.2 Párování v obecných grafech

9.2.1 Střídavé cesty a kružnice. Buď dán graf G a v něm párování P . *Střídavá cesta* (někdy též *alternující cesta*) vzhledem k párování P je taková neorientovaná cesta, že její hrany střídavě leží v P a neleží v P a je-li krajní vrchol cesty nasycen v párování P , pak hrana, která jej nasycuje, je částí cesty.

Příklad střídavé cesty je na obr. 9.1 vlevo vyznačen čárkovaně. Hrany, které náleží k párování, jsou nakresleny silně. Poznamenejme, že cesta spojující vrcholy 1, 6, 3 není střídavou cestou, ale její prodloužení 1, 6, 3, 2 již střídavou cestou je.



Obrázek 9.1: Střídavá cesta a změna párování.

Střídavá kružnice vzhledem k párování P je kružnice, jejíž hrany střídavě leží a neleží v párování P . Střídavá kružnice má vždy sudý počet hran.

Pomocí střídavých cest a kružnic lze párování snadno měnit tím, že u hran, které leží na cestě nebo kružnici, změním příslušnost k párování. Přesněji, je-li H množina hran tvořících střídavou cestu nebo kružnici vzhledem k párování P , vytvoříme nové párování P' takto:

$$\begin{aligned} \text{jestliže } e \in H, & \text{ pak } e \in P' \Leftrightarrow e \notin P, \\ \text{jestliže } e \notin H, & \text{ pak } e \in P' \Leftrightarrow e \in P. \end{aligned}$$

Takovou změnu párování budeme nazývat *změnou podél střídavé cesty* nebo *kružnice*. Výsledek změny párování podél střídavé cesty je na obr. 9.1 vpravo.

9.2.2 Věta. Buď dán prostý graf G a v něm libovolné párování P_1 . Pak pro každé párování P_2 v grafu G existuje soustava vrcholově disjunktních střídavých cest a střídavých kružnic taková, že změnami podél všech těchto cest a kružnic lze z párování P_1 získat párování P_2 .

POZNÁMKA: Poněvadž střídavé cesty a kružnice nemají žádný společný vrchol (jsou vrcholově disjunktní), lze změny provádět v libovolném pořadí.

DŮKAZ: Uvažujme faktor G' grafu G s množinou hran $(P_1 \setminus P_2) \cup (P_2 \setminus P_1)$. Graf G' tedy obsahuje právě ty hrany grafu G , které leží přesně v jednom z obou párování (tj. nikoli v obou).

Pro každý vrchol $x \in V(G') = V(G)$ nastane jedna ze čtyř možností:

1. Jestliže x není nasycen ani v P_1 , ani v P_2 , je izolovaným vrcholem v G' .
2. Je-li vrchol x nasycen jen v jednom z párování P_1, P_2 , má v grafu G' stupeň 1.
3. Je-li x nasycen v P_1 i P_2 touž hranou, pak tato hrana neleží v G' , a vrchol x je tedy v G' izolovaným vrcholem.
4. Je-li x nasycen v P_1 i P_2 různými hranami $e_1 \in P_1$ a $e_2 \in P_2$, pak obě tyto hrany leží v G' a stupeň vrcholu x je roven 2.

Každý vrchol má tedy v grafu G' stupeň nejvýše 2. Z toho plyne, že graf G' má komponenty souvislosti tří typů:

1. izolovaný vrchol,
2. kružnice sudé délky, jejíž hrany střídavě leží v P_1 a P_2 ,
3. cesta, jejíž hrany střídavě leží v P_1 a P_2 a jejíž krajní vrcholy jsou různé, přičemž každý z nich je nasycen v jednom z obou párování P_1, P_2 .

Komponenty souvislosti grafu G' přímo určují střídavé cesty a kružnice. Provedením změn párování P_1 podél všech těchto cest a kružnic dostaneme párování P_2 . \square

9.2.3 Věta. Párování P v grafu G je maximální právě tehdy, když v grafu G vzhledem k párování P neexistuje střídavá cesta spojující dva volné vrcholy.

DŮKAZ: Je-li párování P maximální, pak vzhledem k němu nemůže existovat střídavá cesta spojující volné vrcholy. Kdyby totiž existovala, mohli bychom párování podél ní zvětšit.

Jestliže naopak párování P není maximální, vezměme nějaké maximální párování (nějaké určitě existuje) a označme je P_1 . Podle věty 9.2.2 existuje soustava střídavých cest a kružnic taková, že změnami podél nich získáme párování P_1 z párování P . Poněvadž $|P_1| > |P|$, musí některá z těchto změn zvětšovat párování, musí tedy být změnou podél střídavé cesty s volnými krajními vrcholy (jiné změny nezvětšují počet hran v párování). \square

9.2.4 Cena střídavé cesty a kružnice. Uvažujme graf G , jehož hrany jsou ohodnoceny cenami $c : E(G) \rightarrow \mathbb{R}$. Dále mějme v grafu G párování P a vzhledem k němu střídavou cestu, popř. kružnici. Množinu hran této cesty, popř. kružnice, označme H .

Cenu střídavé cesty, popř. kružnice, definujeme jako

$$C = \sum_{e \in H \setminus P} c(e) - \sum_{e \in H \cap P} c(e).$$

Má-li střídavá cesta, popř. kružnice, cenu C a provedeme-li podél ní změnu, pak cena párování vzroste o hodnotu C .

9.2.5 Věta. Párování P v grafu G je nejdražší právě tehdy, když v grafu G vzhledem k párování P neexistuje ani střídavá cesta, ani střídavá kružnice s kladnou cenou.

DŮKAZ je podobný důkazu věty 9.2.3 a je přenechán čtenáři jako cvičení. \square

9.2.6 Algoritmy pro párování v obecných grafech. Všeobecně lze k hledání maximálního i nejdražšího párování využít metody založené na větách 9.2.3 a 9.2.5. Základní schéma algoritmů je toto:

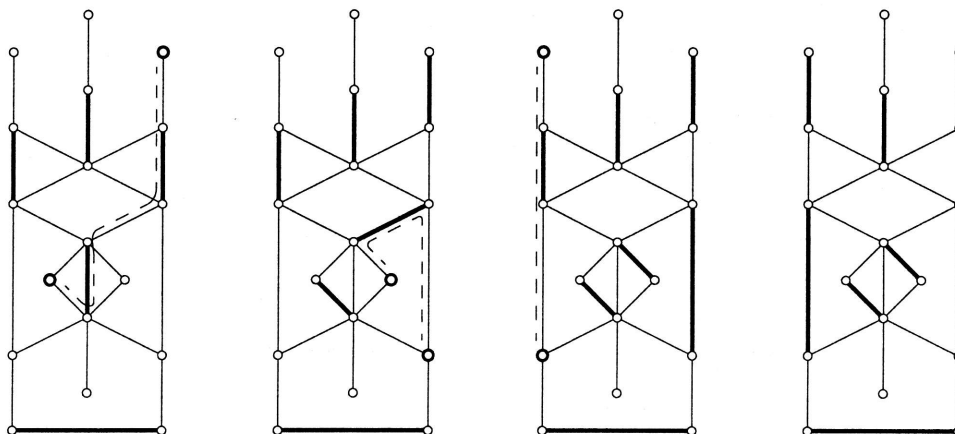
Zvolíme libovolné výchozí párování P a hledáme, zda vzhledem k němu existuje střídatá cesta, popř. kružnice, která by dovolila párování zlepšit. Jestliže neexistuje, pak podle věty 9.2.3, popř. 9.2.5, je párování P již optimální. Jestliže existuje, změníme (tj. zlepšíme) párování a znovu hledáme další střídatou cestu, popř. kružnici.

Bohužel, hledání potřebných střídatých cest a kružnic není tak jednoduché, jak se na první pohled zdá. Běžné metody prohledávání grafů popsané v kap. 3 a založené na značkování vrcholů lze jednoduše upravit pro hledání střídatých cest pouze v bipartitních grafech (viz 9.3).

V obecných grafech sice lze pro hledání střídatých cest použít backtracking (viz 11.2, str. 189), ten však je ve větších grafech neúnosně pomalý.

Rychlé (polynomiální) algoritmy pro párování v obecných grafech existují, pracují v čase $O(n^3)$, ale jsou poměrně komplikované a v případě nejdražšího párování se navíc opírají o tzv. dualitu v lineárním programování. Z tohoto důvodu nastíníme pouze hlavní myšlenky rychlých algoritmů, a to pouze pro maximální párování (viz 9.2.8). Polynomiální algoritmy pro maximální i nejdražší párování jsou popsány např. v knihách [Plesník83], [Chr75] a [Lawler76].

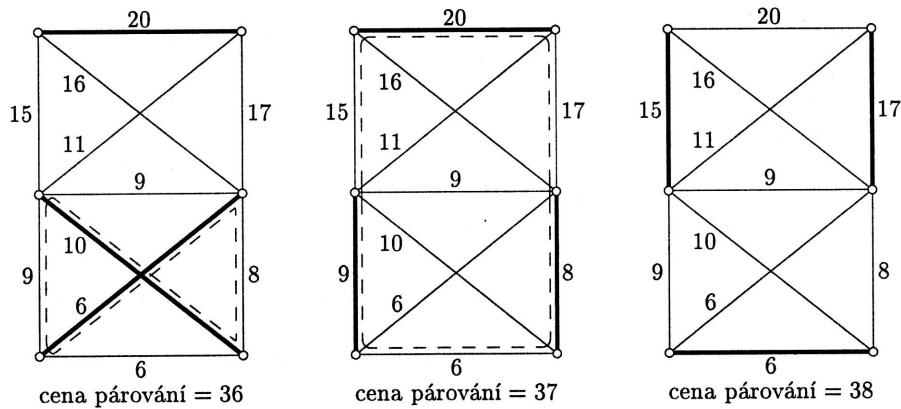
9.2.7 Příklad. Na obr. 9.2 a 9.3 jsou příklady hledání maximálního a nejdražšího párování.



Obrázek 9.2: Výpočet maximálního párování.

Výchozí párování na obr. 9.2 vlevo bylo získáno náhodným přidáváním hran. Všimněte si, že k tomuto párování už nelze přidat žádnou další hranu, přesto má dost daleko k maximálnosti. Střídaté cesty mezi volnými vrcholy jsou vyznačeny čárkovaně. Všimněte si také, že výsledné maximální párování není perfektní.

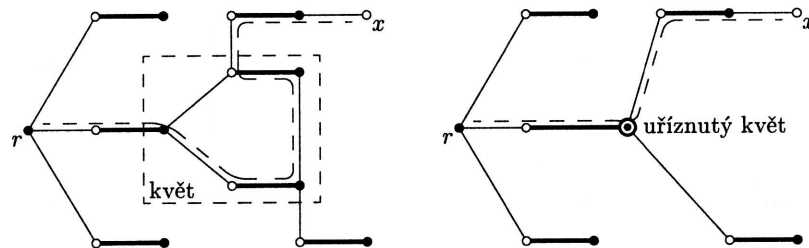
Výchozí párování na obr. 9.3 vlevo bylo získáno „hladovým“ postupem: postupně jsme přidávali vždy nejdražší přijatelnou hranu. Střídavé cesty i kružnice byly hledány ručně s využitím backtrackingu. Všimněte si, že nejdražší párování na obr. 9.3 vpravo jsme mohli získat také přímo z výchozího párování, kdybychom zvolili jinou střídavou kružnici.



Obrázek 9.3: Výpočet nejdražšího párování.

9.2.8 Rychlý algoritmus pro maximální párování. Naznačíme hlavní myšlenky rychlého algoritmu pro hledání střídavé cesty, jejíž krajní vrcholy jsou volné v párování P . Podrobnější popis nebo jiné postupy lze najít v [Kučera83], [Plesník83], [Chr75], [Lawler76], [SwTh81].

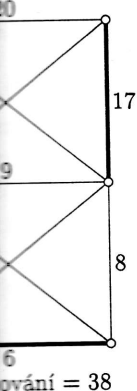
Začínáme v některém volném vrcholu r , který nazveme *kořenem*, a vytváříme tzv. *střídavý strom*, který má tu vlastnost, že každá cesta v tomto stromě, která začíná v r , je střídavou cestou. Strom vytváříme pomocí značkování vrcholů, přičemž používáme dva druhy značek: vrcholy stromu označujeme střídavě jako *vnější* a *vnitřní*, přičemž kořen r je vnější. V příkladě na obr. 9.4 jsou vnější vrcholy kresleny jako plné tečky.



Obrázek 9.4: Uříznutí květu a střídavá cesta procházející květem.

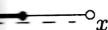
Jestliže najdeme hranu, která spojuje vnější vrchol s nějakým volným vrcholem různým od r , máme střídavou cestu s volnými krajními vrcholy a můžeme zvětšit párování.

tem: postupně kružnici byly párování na obr. bychom zvolili



me hlavní myš- ní vrcholy jsou t v [Kučera83],

m, a vytváříme o stromě, která vrcholů, přičemž avě jako vnější a vnější vrcholy



utý květ



ětem.

volným vrcholem můžeme zvětšit

Jestliže najdeme hranu, která spojuje dva vnější vrcholy (a která tedy neleží v párování), našli jsme kružnici liché délky, která bývá nazývána *květem*. Květ kazí vlastnost střídavého stromu, totiž že cesta začínající v r je střídavou cestou. Proto všechny vrcholy květu nahradíme jediným vrcholem a hrany, které vedly z vnějšku do květu, vedeme do tohoto vrcholu. Tato operace bývá nazývána *uříznutím květu*. Lze dokázat, že střídavá cesta z r do některého volného vrcholu x v původním grafu existuje právě tehdy, když existuje i v upraveném grafu. Dále tedy pokračujeme s upraveným grafem (viz obr. 9.4) a dále se snažíme střídavý strom rozšířit.

Jestliže střídavý strom nelze dále rozšířit a nenastává žádný z předchozích případů, tj. když z vnějších vrcholů vedou hrany pouze do vnitřních vrcholů, odstraníme z grafu celý strom a všechny hrany s ním incidentní. Lze totiž dokázat, že část párování, která ležela ve stromě, je částí některého maximálního párování. To znamená, že vynechanou částí grafu se již není třeba dále zabývat. (Ověřte na příkladě!)

Tento postup opakujeme, dokud buď nezredukujeme graf na prázdný (pak stávající párování je již maximální), nebo dokud nenalezneme střídavou cestu s volnými krajními vrcholy. V tomto druhém případě lze stávající párování zvětšit. Jestliže však střídavá cesta prochází přes vrchol, který vznikl uříznutím květu, musíme nejprve květ obnovit a najít v něm chybějící část střídavé cesty. To se může i několikrát opakovat, neboť uříznutý květ mohl být součástí dalšího květu atd.

Praktické provedení tohoto algoritmu je relativně komplikované. Uřezávání květů se neprovádí změnami grafu, ale pomocí soustavy odkazů ve vhodné datové struktuře. Detaily i důkaz správnosti lze najít v [Plesník83], [Chr75], [Lawler76], poněkud odlišný algoritmus je uveden v [Kučera83]. Časové nároky těchto algoritmů jsou $O(n^3)$, popř. až $O(n^2\sqrt{n})$.

Poznamenejme, že v grafu, který neobsahuje kružnici liché délky (tj. který je bipartitní), se květ nemůže vyskytnout.

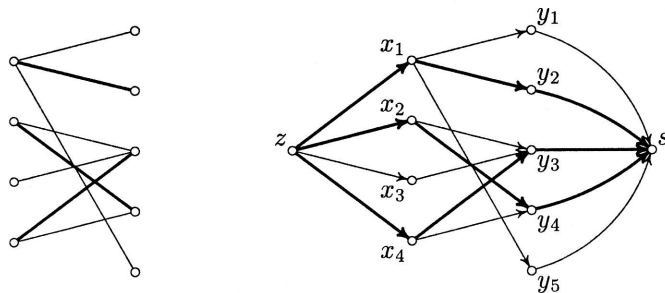
9.3 Párování v bipartitních grafech

Teorie i algoritmy pro párování v bipartitních grafech jsou podstatně jednodušší než v obecném případě, a to díky těsnému vztahu bipartitního párování k tokům v síti.

9.3.1 Převod bipartitního párování na tok v síti. Mějme bipartitní graf G se stranami X, Y . Orientaci hran tohoto grafu zvolme tak, aby všechny hrany vedly z X do Y . K takto vzniklému orientovanému grafu přidejme dva vrcholy, umělý zdroj z a umělý spotřebič s (viz obrázek 9.5 na následující straně). Dále pro každý vrchol $x \in X$ přidejme hranu ze zdroje z do x a pro každý vrchol $y \in Y$ přidejme hranu z y do spotřebiče s . Kapacity těchto přidaných hran jsou jednotkové, kapacity původních hran jsou větší nebo rovny jedné. Dolní omezení toku jsou ve všech hranách nulová.

Párování v původním grafu vzájemně jednoznačně odpovídají celočíselným tokům od zdroje ke spotřebiči: párování je tvořeno těmi hranami původního grafu,

kterými teče nenulový (tj. jednotkový) tok. Maximální párování lze tedy hledat algoritmem pro maximální tok, nejdražší párování algoritmem pro nejlevnější tok (po obrácení znamének cen a po eventuálním přidání návratové hrany).



Obrázek 9.5: Převod úlohy o párování v bipartitním grafu na úlohu o toku v síti.

9.3.2 Speciální algoritmy pro maximální párování v bipartitním grafu jsou ve své podstatě jen přeformulováním a zjednodušením algoritmů určených pro maximální tok. Zlepšující cestě, která se používá ke změnám toku v transportní síti, odpovídá v původním grafu střídatá cesta s volnými krajními vrcholy.

Přímým přeformulováním značkovací procedury 8.3.5 dostaneme tento návod:

1. [Inicializace.] Označujeme všechny volné vrcholy z množiny X . Ostatní vrcholy jsou beze značek.
2. [Test nalezení cesty.] Je-li označován některý volný vrchol z množiny Y , značkovací procedura končí. Je nalezena střídatá cesta s volnými krajními vrcholy a podél ní lze zvětšit počet hran v párování.
3. [Pokus o značkování vpřed.] Jestliže existuje hrana $e \notin P$ vedoucí z označovaného vrcholu $x \in X$ do neoznačovaného vrcholu $y \in Y$, pak označujeme vrchol y a pokračujeme podle kroku 2. Neexistuje-li taková hrana, pokračujeme podle kroku 4.
4. [Pokus o značkování vzad.] Jestliže existuje hrana $e \in P$ z neoznačovaného vrcholu $x \in X$ do označovaného vrcholu $y \in Y$, pak označujeme vrchol x a pokračujeme podle kroku 3. Neexistuje-li taková hrana, značkovací procedura končí, stávající párování je maximální.

Tento postup lze ještě dále zjednodušit takto:

1. [Inicializace.] Označujeme všechny volné vrcholy z množiny X . Ostatní vrcholy jsou beze značek.
2. [Pokus o značkování vpřed.] Jestliže existuje hrana $e \notin P$ vedoucí z označovaného vrcholu $x \in X$ do neoznačovaného vrcholu $y \in Y$, pak označujeme vrchol y a pokračujeme podle kroku 3. Neexistuje-li taková hrana, značkování končí, střídatá cesta spojující volné vrcholy neexistuje.

lze tedy hledat
nejlevnější tok
(any).

3. [Test nalezení cesty.] Je-li právě označovaný vrchol y volný, značkovácí procedura končí, je nalezena střídavá cesta s volnými krajními vrcholy. Je-li vrchol y párováním P nasycen, pokračujeme podle kroku 4.
4. [Značkování vzad.] Označujeme vrchol z množiny X , který je spárován s vrcholem y , a pokračujeme podle kroku 2.

Jestliže takto upravená značkovácí procedura skončí, aniž našla střídavou cestu s volnými krajními vrcholy, pak stávající párování již je maximální. Důkaz tohoto důležitého tvrzení vyplývá z věty 9.2.3 nebo také z analogie s toky v sítích (věta 8.3.7).

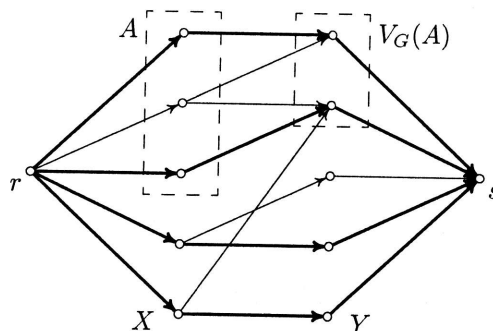
9.3.3 Königova věta. Buď dán bipartitní graf G se stranami X a Y . Počet hran v maximálním párování v grafu G je roven

$$(9.1) \quad \min_{A \subseteq X} (|X \setminus A| + |V_G(A)|).$$

POZNÁMKA: Tato věta má pro párování v bipartitních grafech podobný význam jako Fordova-Fulkersonova věta 8.3.8 pro toky v sítích. Množinu A , pro kterou ve výrazu (9.1) nastává minimum, lze sestrojít pomocí značkovácí procedury.

DŮKAZ: Ke grafu G sestrojme transportní síť T podle 9.3.1.

Vezměme libovolnou množinu $A \subseteq X$ a uvažujme řez $W(Z)$ určený množinou vrcholů $Z = \{z\} \cup A \cup V_G(A)$. (Viz obr. 9.6.) Tento řez obsahuje $|X \setminus A|$ hran z množiny $E^+(z)$ a $|V_G(A)|$ hran z množiny $E^-(s)$. Tyto hrany mají jednotkové kapacity, ostatní hrany kapacitu řezu neovlivňují. Kapacita řezu $W(Z)$ je tedy rovna $|X \setminus A| + |V_G(A)|$. Proto žádné párování nemůže mít více než $|X \setminus A| + |V_G(A)|$ hran.



Obrázek 9.6: K důkazu Königovy věty.

Zbývá dokázat, že existuje množina $A \subseteq X$, pro kterou nastane rovnost. Podle Fordovy-Fulkersonovy věty 8.3.8 je velikost maximálního toku, tj. velikost párování, rovna kapacitě minimálního řezu. Dokážeme, že minimální řez má popsání tvar $W(Z)$, kde $Z = \{z\} \cup A \cup V_G(A)$ pro nějakou množinu $A \subseteq X$.

Uvažujme tedy některý maximální tok f a použijme na něj značkovací proceduru 8.3.5, str. 141. Výslednou množinu označovaných vrcholů označme Z a dále označme $A = X \cap Z$. (Viz obr. 9.6 na předchozí straně.)

Uvažujme libovolný vrchol $x \in A$. Jestliže vrcholem x neprotéká žádný tok, pak vrchol x byl označován po hraně ze zdroje z a určitě jsou označovány všechny vrcholy množiny $V_G(x)$. Jestliže vrcholem x tok protéká, pak vrchol x nemohl být označován po hraně ze zdroje (neboť je nasycena tokem), musel tedy být označován po (jediné) hraně, kterou tok z vrcholu x odtéká. Pak ovšem musel být koncový vrchol této hrany označován dříve než x . Ostatními hranami z $E^+(x)$ tok neteče, proto jejich koncové vrcholy jsou jistě označovány (kdyby nebyly, bylo by možno dále značkovat). Tím je dokázáno, že $V_G(A) \subseteq Y \cap Z$. Žádný vrchol z množiny $Y \setminus V_G(A)$ nemůže být označován, neboť do něj nevede žádná hrana z označovaného vrcholu (tj. z množiny A). Platí tedy

$$V_G(A) = Y \cap Z \quad \text{a} \quad Z = \{z\} \cup A \cup V_G(A),$$

a pro množinu A tedy nastává rovnost. \square

9.3.4 Hallova věta. Nechť G je bipartitní graf se stranami X a Y . V grafu G existuje párování, které nasycuje množinu X právě tehdy, když

$$(9.2) \quad \text{pro každou množinu } A \subseteq X \text{ platí } |A| \leq |V_G(A)|.$$

DŮKAZ: Pokud existuje párování nasycující množinu X , pak podmínka (9.2) triviálně platí. Zajímavá je pouze opačná implikace.

Z podmínky (9.2) plyne, že pro každou množinu $A \subseteq X$ platí $|X \setminus A| + |V_G(A)| = |X| - |A| + |V_G(A)| \geq |X|$. Podle Königovy věty je tedy počet hran v maximálním párování větší nebo roven $|X|$. Větší být ovšem nemůže, proto je roven $|X|$. \square

9.3.5 Věta. Jestliže v bipartitním grafu G se stranami X, Y pro každé $x \in X$ a $y \in Y$ platí $d(x) \geq d(y)$, pak v G existuje párování, které nasycuje všechny vrcholy množiny X .

DŮKAZ: Označme

$$d_X = \min_{x \in X} d(x) \quad \text{a} \quad d_Y = \max_{y \in Y} d(y).$$

Platí tedy $d_X \geq d_Y$. Vezměme nyní libovolnou množinu $A \subseteq X$. O počtu hran v řezu $W(A)$ platí

$$|W(A)| \geq |A| \cdot d_X$$

a také

$$|W(A)| \leq |W(V(A))| \leq |V(A)| \cdot d_Y.$$

Odtud $|A| \cdot d_X \leq |V(A)| \cdot d_Y$, a poněvadž $d_X \geq d_Y$, platí $|A| \leq |V(A)|$. Toto platí pro libovolnou množinu $A \subseteq X$. Podle Hallovy věty tedy existuje párování, které nasycuje množinu X . \square

9.3.6 Cvičení. Dokažte, že z předpokladů předchozí věty plyne, že $|X| \leq |Y|$. Charakterizujte grafy, pro které jsou splněny předpoklady věty a platí $|X| = |Y|$.

9.3.7 Cvičení. Pomocí předchozí věty dokažte, že v příkladu 9.1.6 (taneční) existuje řešení, tj. perfektní párování.

9.3.8 Věta. V každém bipartitním grafu existuje párování, které nasycuje všechny vrcholy s maximálním stupněm.

DŮKAZ: Označme k maximální stupeň vrcholu. Graf doplníme novými hranami a vrcholy tak, aby byl bipartitní a všechny vrcholy měly stupeň k . V takto vzniklém grafu mají obě strany stejný počet vrcholů a podle věty 9.3.5 v něm existuje perfektní párování. Nyní ze zvětšeného grafu vypustíme všechny přidané hrany a vrcholy. Tím získáme párování v původním grafu. Toto párování nasycuje všechny vrcholy stupně k , neboť k nim nebyla přidána žádná hrana. \square

9.3.9 Věta. Nechť v bipartitním grafu G se stranami X, Y existují dvě párování P_1, P_2 . Nechť párování P_1 nasycuje množinu vrcholů $X_1 \subseteq X$ a párování P_2 nasycuje množinu $Y_2 \subseteq Y$. Potom existuje párování $P \subseteq P_1 \cup P_2$, které nasycuje obě množiny $X_1 \subseteq X$ i $Y_2 \subseteq Y$.

DŮKAZ: Označme G' faktor grafu G s množinou hran $P_1 \cup P_2$. V grafu G' mají všechny vrcholy stupeň 0, 1 nebo 2, komponenty souvislosti grafu G' jsou tedy izolované vrcholy, cesty a kružnice.

Položme $P = P_1$. Jestliže P nasycuje všechny vrcholy z Y_2 , pak P je hledané párování. Jestliže P nenasytuje některý vrchol $y \in Y_2$, pak vrchol y je krajním vrcholem cesty, která je komponentou souvislosti grafu G' . Tuto cestu můžeme pokládat za střídavou cestu vzhledem k párování P ; provedme změnu P podél této cesty. Nyní je vrchol y nasycen v P a všechny vnitřní vrcholy cesty také zůstávají nasyceny v P . Označme v druhý krajní vrchol cesty. Jestliže $v \in X$, pak je nyní vrchol y nasycen v P . Jestliže $v \in Y$, pak sice není nasycen v P , ale není toho třeba, neboť v tomto případě byl vrchol v nasycen pouze v P_1 , a tedy $v \notin Y_1$. Opakováním tohoto postupu budeme měnit párování P tak, že bude nasycovat stále větší počet prvků množiny Y_2 a zároveň celou množinu X_1 . \square

9.4 Přiřazovací úloha

Uvedeme tzv. maďarský algoritmus pro řešení přiřazovací úlohy, tj. pro hledání nejlevnějšího perfektního párování v úplném bipartitním grafu $K_{n,n}$, jehož hrany jsou ohodnoceny cenami. Algoritmus je založen na změnách cen, které nemají vliv na výsledné řešení.

9.4.1 Matice cen a její transformace. Buď G úplný bipartitní graf se stranami X, Y takový, že $|X| = |Y| = n$. Předpokládejme, že vrcholy obou stran X, Y jsou očíslovány čísly $1, 2, \dots, n$. Hrany grafu G můžeme bez újmy na obecnosti

ztotožnit s uspořádanými dvojicemi vrcholů $(x, y) \in X \times Y$. Ceny hran pak můžeme přirozeným způsobem uspořádat do tvaru matice C , jejíž libovolný prvek $c(i, j)$ je cenou hrany (i, j) .

Vezměme libovolné ohodnocení vrcholů grafu G reálnými čísly $p(v)$ pro $v \in V(G)$ a na základě tohoto ohodnocení definujeme matici transformovaných cen C_p předpisem

$$c_p(i, j) = c(i, j) - p(i) - p(j).$$

Transformací cen se optimální řešení přiřazovací úlohy nezmění, neboť cena každého perfektního párování se sníží o tutéž hodnotu

$$\sum_{i \in X} p(i) + \sum_{j \in Y} p(j) = \sum_{v \in V(G)} p(v).$$

(Ověřte na příkladě!)

Ohodnocení vrcholů p nazveme *přípustným ohodnocením*, jsou-li všechny transformované ceny nezáporné, tj. $c_p(i, j) = c(i, j) - p(i) - p(j) \geq 0$.

Je-li p přípustné ohodnocení vrcholů, pak *grafem rovnosti* G_p nazveme faktor grafu G , který obsahuje právě ty hrany grafu G , jejichž transformovaná cena je nulová, tj. $c(i, j) = p(i) + p(j)$. Poznamenejme, že všechny ostatní hrany grafu G mají transformované ceny kladné. Platí tedy tato věta:

9.4.2 Věta. Jestliže graf rovnosti G_p určený ohodnocením vrcholů p obsahuje perfektní párování P , pak párování P je optimálním řešením přiřazovací úlohy.

DŮKAZ: Párování P má v transformovaných cenách C_p nulovou cenu. Žádné jiné párování nemůže být levnější, protože matice C_p neobsahuje záporné prvky. P je tedy nejlevnějším párováním vzhledem k transformovaným cenám C_p . Pak je ovšem nejlevnější i vzhledem k původním cenám C . \square

POZNÁMKA: Při notné dávce štěstí bychom mohli párování P a ohodnocení p uhodnout a pomocí této věty pak už jen ověřit, že jsme hádali správně.

9.4.3 Maďarský algoritmus pro přiřazovací úlohu. Hlavní myšlenka tzv. maďarského algoritmu spočívá v nalezení takového přípustného ohodnocení vrcholů, že v příslušném grafu rovnosti existuje perfektní párování.

Výchozí transformovanou matici cen a přípustné ohodnocení vrcholů získáme tím, že od každého řádku původní matice cen odečteme minimální cenu. Tím získáme v každém řádku alespoň jednu nulu. Tutéž operaci můžeme provést také se sloupci, pak i každý sloupec obsahuje alespoň jednu nulu. K takto získané transformované matici cen C_p sestrojíme graf rovnosti G_p a v něm najdeme maximální párování (viz 9.3.2, str. 168). Získáme-li takto perfektní párování, je úloha vyřešena.

Jestliže v grafu G_p perfektní párování neexistuje, pak v něm podle Hallovy věty existuje množina $A \subseteq X$ taková, že $|A| > |V_{G_p}(A)|$. Tuto množinu najde značkovací procedura 9.3.2 při neúspěšném pokusu o nalezení střídavé cesty s volnými krajními vrcholy: $A = Z \cap X$, kde Z je množina všech označovaných vrcholů.

Abychom mohli v grafu rovnosti G_p zvětšit párování, potřebujeme označkovat další vrchol. K tomu je třeba přidat ke grafu G_p některou hranu, která vede z množiny A do neoznačkováného vrcholu v množině Y . Potřebujeme tedy změnit ceny, ale chceme to udělat tak, abychom tuto změnu mohli pokládat za transformaci cen ve smyslu 9.4.1.

Zvýšíme tedy $p(i)$ na množině A a zároveň snížíme $p(j)$ na množině $V_{G_p}(A)$ o tutéž hodnotu d takovou, aby se vynulovala transformovaná cena na některé hraně vedoucí z A do $Y \setminus V_{G_p}(A)$ a přitom aby všechny transformované ceny zůstaly nezáporné. Prakticky to vypadá tak, že hodnota d bude rovna minimu (ze současných transformovaných) cen v průsečíku označkových řádků a neoznačkových sloupců.

Takto provedená změna ohodnocení zaručuje, že v grafu rovnosti zůstanou všechny hrany tvořící párování a také všechny hrany, které byly použity k označování některého vrcholu. Navíc však ke grafu rovnosti přibude alespoň jedna hrana, po které bude možno označkovat alespoň jeden další vrchol. (Přitom ovšem může některá nepotřebná hrana z grafu rovnosti také zmizet.)

V takto upraveném grafu rovnosti opět hledáme maximální párování, tj. znovu použijeme značkovací proceduru a buď zvětšíme párování, nebo znovu měníme ohodnocení vrcholů, abychom mohli označkovat další vrchol. Po konečném počtu kroků nutně získáme perfektní párování, které je podle věty 9.4.2 optimálním řešením přiřazovací úlohy.

Maďarský algoritmus pro řešení přiřazovací úlohy lze shrnout takto:

1. [Počáteční přípustné ohodnocení vrcholů.] Pro každé $i \in X$ položíme $p(i) := \min_{j \in Y} c(i, j)$. Pro každé $j \in Y$ dále položíme $p(j) := \min_{i \in X} (c(i, j) - p(i))$.

2. [Graf rovnosti.] Sestrojíme faktor G_p grafu G takový, že

$$E(G_p) = \{(i, j) \in E(G) \mid c(i, j) - p(i) - p(j) = 0\}.$$

3. [Maximální párování.] Sestrojíme maximální párování P v grafu G_p . Je-li toto párování perfektní, výpočet končí, výsledné perfektní párování je nejlevnější.

4. [Změna přípustného ohodnocení vrcholů.] Není-li párování P perfektní, nalezneme množinu $A \subseteq X$ takovou, že $|A| > |V_{G_p}(A)|$, vypočteme

$$d = \min\{c(i, j) - p(i) - p(j) \mid i \in A, j \notin V_{G_p}(A)\}$$

a změníme přípustné ohodnocení vrcholů takto:

$$\begin{aligned} p(i) &:= p(i) + d && \text{pro všechna } i \in A, \\ p(j) &:= p(j) - d && \text{pro všechna } j \in V_{G_p}(A). \end{aligned}$$

Pokračujeme podle kroku 2.

ČASOVÉ NÁROKY: $O(n^4)$.

9.4.4 Příklad. Řešme přiřazovací úlohu, která je dána touto maticí cen:

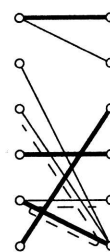
5	3	7	4	5	4
10	11	10	7	8	3
18	7	6	6	6	2
6	12	2	1	9	8
8	4	4	4	1	1
4	8	1	3	7	4

Odečtením řádkových minim od jednotlivých řádků získáme tuto transformovanou matici cen a ohodnocení $p(i)$ vrcholů strany X :

							$p(i)$
2	0	4	1	2	1	3	
7	8	7	4	5	0	3	
16	5	4	4	4	0	2	
5	11	1	0	8	7	1	
7	3	3	3	0	0	1	
3	7	0	2	6	3	1	

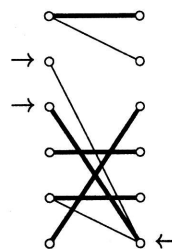
Podobně odečteme od každého sloupce sloupcové minimum. Tím získáme výchozí ohodnocení $p(j)$ vrcholů strany Y a transformovanou matici cen, která již obsahuje v každém řádku a v každém sloupci alespoň jednu nulu. Sestrojíme příslušný graf rovnosti a v něm zvolíme (v podstatě libovolným způsobem) výchozí párování.

							$p(i)$
	0	0	4	1	2	1	3
	5	8	7	4	5	0	3
14	5	4	4	4	0	0	2
3	11	1	0	8	7	1	1
5	3	3	3	0	0	0	1
1	7	0	2	6	3	1	1
$p(j)$	2	0	0	0	0	0	



Značkováním jsme našli střídavou cestu s volnými krajními vrcholy a podél této cesty změníme (tj. zvětšíme) párování. Matice transformovaných cen ani samotný graf rovnosti se tím nemění. Poněvadž výsledné párování ještě není perfektní, pokračujeme dalším značkováním z volného vrcholu:

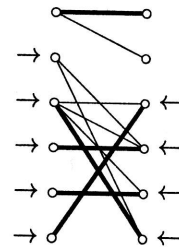
							$p(i)$
	0	0	4	1	2	1	3
→	5	8	7	4	5	0	3
→	14	5	4	4	4	0	2
	3	11	1	0	8	7	1
	5	3	3	3	0	0	1
	1	7	0	2	6	3	1
$p(j)$	2	0	0	0	0	0	



ící cen:

Označované vrcholy a jim odpovídající řádky a sloupce jsou označeny šipkami. Velikost změny přípustného ohodnocení vrcholů $d = 4$ získáme jako minimum z transformovaných cen v průsečíku označených řádků a neoznačených sloupců, tj. zde druhého a třetího řádku a prvního až pátého sloupce. Po transformaci matice dostaneme:

			↓	↓	↓	↓	$p(i)$
	0	0	4	1	2	5	3
→	1	4	3	0	1	0	7
→	10	1	0	0	0	0	6
→	3	11	1	0	8	11	1
→	5	3	3	3	0	4	1
→	1	7	0	2	6	7	1
$p(j)$	2	0	0	0	0	-4	



ansformovanou

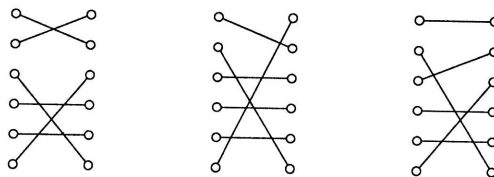
V transformované matici přibylo několik nul a v grafu rovnosti přibylo několik hran. Hrana (5,6) naopak z grafu rovnosti ubyla. Párování stále ještě nelze zvětšit, ale zvětšil se alespoň počet označovaných vrcholů. Opět vybíráme minimum z průsečíku označovaných řádků a neoznačovaných sloupců, tentokrát $d = 1$:

							$p(i)$
	0	0	5	2	3	6	3
	0	3	3	0	1	0	7
	9	0	0	0	0	0	6
	2	10	1	0	8	11	1
	4	2	3	3	0	4	1
	0	6	0	2	6	7	1
$p(j)$	2	0	-1	-1	-1	-4	



skáme výchozí
rá již obsahuje
příslušný graf
í párování.

Nyní již v grafu rovnosti existuje střídavá cesta, podél níž lze párování zvětšit. Dokonce jsou zde tři takové cesty, nakreslena je pro přehlednost pouze jedna z nich. Tři výsledná perfektní párování jsou na obr. 9.7. Cena každého z nich je rovna součtu ohodnocení vrcholů, tj. 18.



Obrázek 9.7: Tři rovnocenná optimální řešení příkladu 9.4.4.

9.4.5 Cvičení. Řešte tutéž přiřazovací úlohu, ale volte jinak výchozí přípustné ohodnocení vrcholů a výchozí graf rovnosti. Výsledné optimální přiřazení musí mít samozřejmě stejnou cenu, ale postup výpočtu může být hodně odlišný.

ly a podél této
n ani samotný
nění perfektní,

9.4.6 Poznámka. Při výpočtu na počítači se obvykle graf rovnosti explicitě nesestrojuje, maximální párování lze hledat pomocí nulových prvků matice C_p , značkují se přitom řádky a sloupce matice.

9.4.7 Jiné algoritmy pro přiřazovací úlohu. Přiřazovací úlohu lze pokládat za speciální případ klasické dopravní úlohy 8.2.4, str. 135, kde X je množina dodavatelů, Y je množina spotřebitelů a kapacity všech dodavatelů i spotřebitelů jsou jednotkové. Dopravní úlohu pak lze řešit buď speciálním algoritmem, nebo jako úlohu o nejlevnější cirkulaci.

Jinou možnost představuje převod přiřazovací úlohy na úlohu o nejdražším párování v témže úplném bipartitním grafu. Tento převod spočívá ve změně cen hran (viz cvičení 9.1.8). Výslednou úlohu o nejdražším párování v bipartitním grafu pak lze řešit tak, že vyjdeme z prázdného párování, které postupně zvětšujeme, a to vždy podél střídavé cesty, která spojuje dva nenasycené vrcholy a která má vzhledem ke stávajícímu párování největší cenu (viz 9.2.4). Lze dokázat, že takto dostaneme nejdražší párování, které je zároveň optimálním řešením přiřazovací úlohy.

vnosti explicitě
vků matice C_p ,

hu lze pokládat
 X je množina
ů i spotřebitelů
mem, nebo jako

u o nejdražším
á ve změně cen
partitním grafu
většujeme, a to
rá má vzhledem
akto dostaneme
í úlohy.

Kapitola 10

Eulerovské tahy

10.1 Základní pojmy a aplikace

10.1.1 Eulerovský tah, pokrývání hran. Připomeňme, že *tah* je sled, který neobsahuje žádnou hranu dvakrát (viz 1.5.2, str. 20). *Eulerovský tah* je takový tah, který obsahuje všechny hrany grafu (tedy každou hranu přesně jedenkrát). Eulerovské tahy dělíme na *orientované* a *neorientované* a na *uzavřené* a *neuzavřené*.

Řekneme, že soustava tahů *pokrývá hrany grafu*, jestliže každá hrana grafu leží přesně v jednom z nich. Eulerovský tah je tedy takový tah, který sám pokrývá všechny hrany grafu.

V souvislosti s eulerovskými tahy lze formulovat a řešit tyto úlohy:

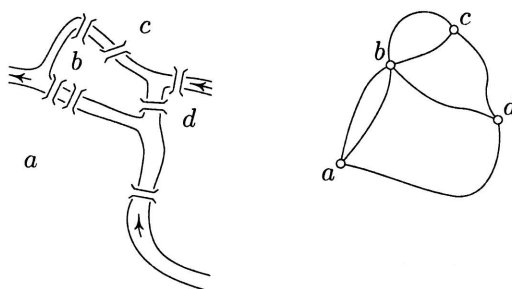
1. Rozhodnout, zda v grafu existuje eulerovský tah.
2. Sestrojit eulerovský tah.
3. Najít nejmenší počet tahů, které pokrývají hrany daného grafu.
4. V daném souvislém grafu, jehož hrany jsou ohodnoceny kladnými čísly, najít nejkratší uzavřený sled, který obsahuje všechny hrany grafu (každou hranu aspoň jedenkrát).

Pro řešení těchto úloh jsou známy rychlé (polynomiální nebo dokonce lineární) algoritmy.

Snadnost řešení těchto úloh je v pozoruhodném kontrastu s obtížností tzv. hamiltonovských úloh, které uvedeme v kapitole 12 (viz 12.1.4, str. 198).

10.1.2 Sedm mostů města Královce. Město Královec (Königsberg, dnešní Kaliningrad) leží na březích řeky Pregel a na dvou ostrovech. Břehy a ostrovy byly v 18. století spojeny sedmi mosty zhruba podle obr. 10.1 na následující straně. Obyvatelé města se tehdy bavili otázkou, zda je možno vykonat procházku, která by vedla přes každý most přesně jedenkrát. Samozřejmě se při tom nesmělo plavat přes řeku. Dokonce se o tom uzavíraly a prohrávaly sázky, dokud Leonard Euler v roce 1736 nedokázal, že taková procházka není možná. Eulerův způsob řešení této hříčky bývá označován za počátek teorie grafů (historické podrobnosti viz [Šišma97]).

Úloha o sedmi mostech je ekvivalentní úloze najít v grafu, který je nakreslen na obr. 10.1 vpravo, eulerovský tah.



Obrázek 10.1: Sedm mostů města Královce a odpovídající graf.

10.1.3 Kreslení co nejmenším počtem tahů. Na hledání eulerovského tahu lze zřejmým způsobem převést známou úlohu, nakreslit nějaký daný obrázek (třeba „domeček“) jedním tahem, aniž bychom kreslili některou čáru dvakrát a aniž bychom zvedli tužku z papíru. Od této úlohy je zřejmě odvozen termín *tah*. Každý asi ví, že na některé obrázky jeden tah nestačí.

Vzniká tedy přirozená otázka, jaký nejmenší počet tahů je k nakreslení nutný.

Nemusí přitom jít jen o zábavnou úlohu. Např. při kreslení pomocí počítače je účelné minimalizovat počet a délku přejezdů pisátka bez kreslení. Poněvadž každá čára by se měla kreslit přesně jedenkrát, je tato úloha příbuzná úloze 4. Pozor, nikoli totožná, a to ze dvou důvodů: obrázek nemusí být souvislý a navíc mohou být přejezdy pisátka přímočaré a nemusí probíhat podél kreslených čar. Je-li však kresba souvislá, lze k řešení použít postup podobný algoritmu 10.3.2.

10.1.4 Úloha čínského pošťáka. Pošťák musí při roznášce pošty projít všechny ulice svého rajónu. Jak má postupovat, aby ušel co nejméně kilometrů?

Tato úloha je pouze slovním vyjádřením úlohy 4, uvedené v 10.1.1. Jiným příkladem ze stejného soudku je optimalizace jízdy kropicího vozu, který má pokropit všechny ulice ve městě. Řešením se budeme zabývat v 10.3.

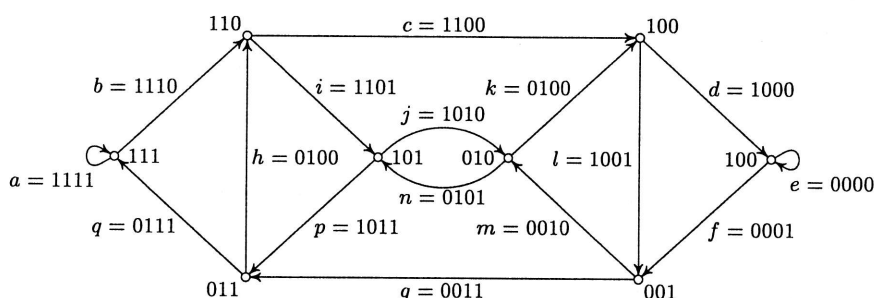
10.1.5 De Bruijnova posloupnost je příkladem důmyslné aplikace eulerovských tahů:

Máme dáno kladné číslo k . Úkolem je najít co nejdelší cyklickou posloupnost nul a jedniček takovou, že žádné dvě k -tice po sobě jdoucích cifer nejsou stejné. Taková posloupnost se nazývá *de Bruijnova posloupnost*.

Tato úloha souvisí s kódováním zhruba takto: obvod otáčejícího se kotouče má být označen nulami a jedničkami tak, aby bylo možno určit přesnou pozici kotouče přečtením pouze k po sobě jdoucích číslic.

ŘEŠENÍ: Počet posloupností tvořených k nulami a jedničkami je 2^k . Proto délka hledané de Bruijnovy posloupnosti nemůže být větší než 2^k . Pomocí eulerovských tahů dokážeme, že posloupnost o délce 2^k existuje, a získáme návod, jak ji sestrojít.

Veźměme graf, který má 2^{k-1} vrcholů, viz příklad na obr. 10.2 pro $k = 4$. Vrcholy označíme všemi $(k-1)$ -ticemi nul a jedniček. Graf bude mít 2^k hran označených všemi k -ticemi nul a jedniček, a to tak, že hrana označená $(x_1, x_2, \dots, x_{k-1}, x_k)$ povede z vrcholu $(x_1, x_2, \dots, x_{k-1})$ do vrcholu $(x_2, \dots, x_{k-1}, x_k)$. To znamená, že z každého vrcholu vychází přesně dvě hrany; jejich označení se liší pouze na posledním k -tém místě, jedna hrana tam má jedničku a druhá nulu. Podobně do každého vrcholu vchází dvě hrany, jejichž označení se liší v první číslici.



Obrázek 10.2: Graf pro konstrukci de Bruijnovy posloupnosti pro $k = 4$. Například eulerovskému tahu $abcdefghijklmnpq$ odpovídá de Bruijnova posloupnost 1111000011010010 .

Libovolný sled v takovémto grafu můžeme snadno zakódovat pomocí posloupnosti nul a jedniček: každá číslice vyjadřuje jednu ze dvou možností jak pokračovat z daného vrcholu, který je označen předcházející $(k-1)$ -ticí.

Původní úloha o de Bruijnově posloupnosti je tedy převedena na hledání uzavřeného orientovaného eulerovského tahu v popsaném grafu. Tento graf je souvislý a pro každý jeho vrchol x platí $d^+(x) = d^-(x)$, proto podle věty 10.2.2 požadovaný eulerovský tah existuje. Dokonce je zřejmé, že eulerovských tahů (a tedy i de Bruijnových posloupností) může existovat několik.

Ve výše uvedené úloze jsme použili abecedu o dvou symbolech, totiž $\{0,1\}$. Výše popsanou konstrukci lze snadno zobecnit i pro libovolnou větší abecedu. Má-li abeceda n prvků, bude mít de Bruijnova posloupnost délku n^k .

10.1.6 Cvičení. Najděte de Bruijnovu posloupnost pro $k = 3$ a $n = 2$. Kolik takových posloupností je? Řešte tutěž úlohu pro $k = 2$ a $n = 3$.

10.2 Existence a hledání eulerovských tahů

10.2.1 Věta. (Euler 1736.) Nechť graf G je souvislý. Pak v grafu G existuje neorientovaný uzavřený eulerovský tah právě tehdy, když každý vrchol grafu G má sudý stupeň.

DŮKAZ je téměř shodný s důkazem následující věty 10.2.2. Proto obě věty dokážeme společně. \square

10.2.2 Věta. Nechť graf G je souvislý. Pak v grafu G existuje orientovaný uzavřený eulerovský tah právě tehdy, když pro každý vrchol v platí $d^+(v) = d^-(v)$.

DŮKAZ: Důkaz jedné z implikací je velmi snadný: Předpokládejme, že v grafu existuje uzavřený eulerovský tah (orientovaný nebo neorientovaný). Představme si cestovatele, který prochází grafem podél tohoto tahu. Zřejmě platí, že kolikrát přišel do nějakého vrcholu v , tolikrát z něj musel i odejít. Poněvadž při tom projde každou hranou přesně jedenkrát, plynou odtud okamžitě tvrzení o stupních vrcholů: $d(v)$ je sudý pro neorientovaný tah a $d^+(v) = d^-(v)$ pro tah orientovaný.

Opačná implikace je těžší, dokážeme ji tím, že popíšeme (a dokážeme) postup, jak eulerovský tah sestrojít, jsou-li splněny předpoklady jeho existence, tj. souvislost a vlastnosti stupňů.

Začneme z libovolného vrcholu x_0 a procházejme hranami grafu libovolně, ale tak, abychom žádnou hranou nešli dvakrát. Takto pokračujeme, dokud je to možné, tj. dokud z vrcholu, kam jsme právě přišli, vychází ještě nějaká dosud nepoužitá hrana. Tato procházka zcela jistě skončí v x_0 , protože díky vlastnostem stupňů nemůže skončit nikde jinde a díky konečnosti grafu někde skončit musí. Nalezli jsme tedy nějaký uzavřený tah.

Nyní jsou dvě možnosti. Je-li tento tah eulerovský, jsme hotovi. Není-li tah eulerovský, pak někde v grafu existuje hrana, která v tahu neleží. Mezi x_0 a touto hranou existuje cesta (neboť graf je souvislý) a někde na této cestě musí existovat vrchol x_1 , kterým prochází tah a z něhož ještě vychází nějaká nepoužitá hrana. Ve vrcholu x_1 tah přerušíme (rozpojíme) a začneme opět procházet nepoužitými hranami a tím tah (který je nyní otevřený) prodlužovat. Díky vlastnostem stupňů toto prodlužování skončí ve vrcholu x_1 . V tomto vrcholu navážeme novou část tahu na původní, čímž získáme uzavřený tah, který má více hran.

Tento postup (přerušování a prodlužování tahu) opakujeme, dokud existuje hrana, která v tahu neleží. Díky konečnému počtu hran v grafu G nakonec jistě dostaneme tah, který je eulerovský. \square

10.2.3 Poznámka. Předpoklad souvislosti v předchozích dvou větách je zbytečně silný: izolované vrcholy by nevadily.

10.2.4 Cvičení. Dokažte, že (slabě) souvislý orientovaný graf, jehož každý vrchol v splňuje $d^+(v) = d^-(v)$, je také silně souvislý.

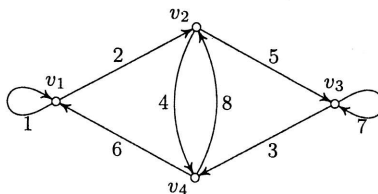
10.2.5 Algoritmus pro hledání uzavřeného eulerovského tahu. Postup byl již vysvětlen v důkaze věty 10.2.2. Zaměříme se na jeho efektivní provedení. V algoritmu se střídavě provádějí dvě fáze:

1. Existující tah prodlužujeme, dokud se nestane uzavřeným.
2. Uzavřený tah kontrolujeme, zda je eulerovský.

Při kontrole procházíme podél tahu a v každém vrcholu x testujeme, zda v množině $E(x)$, popř. $E^+(x)$ existuje hrana h , která dosud není obsažena v tahu. Pokud takovou hranu h najdeme, přerušíme kontrolu, tah ve vrcholu x rozpojíme a začneme

jej prodlužovat počínaje hranou h . Prodlužování skončí ve vrcholu x . Po propojení staré a nové části tahu pokračujeme v kontrole počínaje vrcholem x (předcházející část tahu je již zkontrolována) a postupujeme podél nové části tahu (ta doud není zkontrolována). Tím je zajištěno, že nejen při prodlužování, ale i při kontrole postupujeme podél každé hrany pouze jedenkrát. Celý postup tedy vyžaduje čas $O(n+m)$.

10.2.6 Příklad. Sestrojme uzavřený orientovaný eulerovský tah v grafu, který je nakreslen na obr. 10.3. Použijeme postup 10.2.5. Abychom se vyhnuli nejednoznačnosti, budeme k prodlužování tahu volit vždy hranu s nejnižším číslem a začneme ve vrcholu v_1 .



Obrázek 10.3: Graf k příkladu 10.2.6.

V první fázi získáme tah tvořený hranami 1, 2, 4, 6. Při kontrole zjistíme, že ve vrcholu v_1 jsou všechny hrany obsaženy v tahu, ale z vrcholu v_2 vychází dosud nepoužitá hrana 5. Tah tedy rozpojme mezi hranami 2 a 4, prodloužíme jej o hrany 5, 3, 8 a znovu spojíme. Tím získáme uzavřený tah 1, 2, 5, 3, 8, 4, 6. Pokračujeme v kontrole tam, kde jsme ji přerušili, tj. ve vrcholu v_2 (ten je již v pořádku) a dále ve v_3 , kde zjistíme, že z v_3 vychází nepoužitá hrana 7. Tah proto rozpojme mezi hranami 5 a 3 a prodloužením získáme tah 1, 2, 5, 7, 3, 8, 4, 6. Další kontrola ve vrcholech v_3, v_3, v_4, v_2, v_4 již ukáže, že tento tah je eulerovský.

10.2.7 Věta. Necht' graf G je souvislý a obsahuje k vrcholů lichého stupně. Pak nejmenší počet neorientovaných tahů pokrývajících hrany grafu G je roven $k/2$.

DŮKAZ: Nejprve si uvědomme, že číslo k je sudé: Součet všech stupňů je roven $2|E(G)|$, a je tedy sudý. Kdyby k bylo liché, musel by součet všech stupňů být lichý, což by byl spor (součet lichého počtu lichých čísel je vždy lichý).

Přidejme nyní ke grafu G celkem $k/2$ hran, které budou spojovat vždy dva vrcholy lichého stupně. Tím dostaneme souvislý graf G' , v němž všechny vrcholy budou mít stupně sudé. Podle věty 10.2.1 tedy v grafu G' existuje uzavřený neorientovaný eulerovský tah.

Nyní odstraňme z grafu G' přidané hrany. Tím jednak dostaneme původní graf G , jednak se nám uzavřený eulerovský tah v G' rozpadne na celkem $k/2$ neuzavřených tahů, které dohromady pokrývají všechny hrany grafu G . \square

10.2.8 Cvičení. Kolik tahů je nutných k pokrytí všech hran neorientovaného grafu, o kterém víme, že má 3 komponenty souvislosti, 2 vrcholy lichého stupně a žádný izolovaný vrchol? Bylo by možno určit potřebný počet tahů, kdyby počet vrcholů lichého stupně nebyl 2, ale 4?

10.2.9 Věta. V orientovaném grafu G bez izolovaných vrcholů existuje orientovaný neuzavřený eulerovský tah právě tehdy, když graf G je souvislý a existují v něm dva vrcholy x, y takové, že $d^+(x) = d^-(x) + 1$ a $d^+(y) = d^-(y) - 1$ a pro všechny ostatní vrcholy v platí $d^+(v) = d^-(v)$.

DŮKAZ je podobný důkazu věty 10.2.7. □

10.2.10 Cvičení. Najděte způsob, jak pro (slabě) souvislý orientovaný graf zjistit na základě stupňů vrcholů nejmenší počet orientovaných tahů, které pokrývají jeho hrany. Návod: pro každý vrchol vypočtete číslo $R(v) = d^+(v) - d^-(v)$.

10.3 Úloha čínského pošťáka

10.3.1 Úloha čínského pošťáka. Pošťák má projít všechny ulice města a vrátit se do výchozího místa, a to tak, aby ušel co nejméně kilometrů. Přeloženo do řeči grafů, jde o tuto úlohu:

Je dán neorientovaný souvislý graf, jehož hrany jsou ohodnoceny kladnými čísly. Úkolem je najít nejkratší uzavřený sled, který obsahuje všechny hrany grafu.

10.3.2 Řešení úlohy čínského pošťáka. Je zřejmé, že existuje-li v grafu eulerovský tah, je tento tah hledaným optimálním řešením.

Pokud v grafu eulerovský tah neexistuje, pak sled, který obsahuje všechny hrany, musí některými hranami procházet dvakrát nebo i vícekrát. Lze však dokázat (viz cvičení 10.3.4), že nejkratší sled prochází každou hranou pouze jedenkrát nebo dvakrát. Hledáme tedy sled, v němž je nejmenší součet délek hran, které jsou procházeny opakovaně (tj. dvakrát).

V nejkratším sledu, který obsahuje všechny hrany grafu, tvoří opakovaně procházené hrany soustavu cest, které spojují vždy dva vrcholy lichého stupně. Lze dokázat, že tyto cesty jsou navzájem hranově disjunktní, tj. že žádná hrana grafu neleží ve dvou různých takových cestách (viz cvičení 10.3.5).

Klíčem k řešení úlohy tedy je rozdělit vrcholy s lichým stupněm do dvojic, a to tak, aby součet délek nejkratších cest mezi vrcholy ve dvojicích byl nejmenší. Toto je vlastně úloha o párování, v níž je však třeba spárovat pouze vrcholy lichého stupně. Můžeme se na to dívat také tak, že hledáme nejlevnější perfektní párování v pomocném úplném grafu K , jehož vrcholy jsou všechny vrcholy lichého stupně původního grafu G a délky hran grafu K jsou délky nejkratších cest v grafu G .

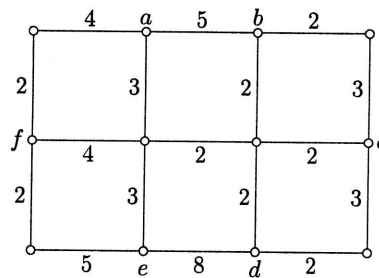
Celý postup řešení lze shrnout takto:

1. V daném grafu G najdeme množinu L vrcholů s lichým stupněm.
2. Pro všechny dvojice (x, y) vrcholů množiny L vypočteme délku $u(x, y)$ nejkratší (neorientované) cesty z x do y .
3. Na množině vrcholů L definujeme úplný graf K , jehož hrany jsou ohodnoceny délkami nejkratších cest $u(x, y)$.
4. V grafu K najdeme nejlevnější perfektní párování P (viz kapitola 9).

5. Pro každou hranu (x, y) grafu K , která leží v párování P , vezmeme všechny hrany původního grafu G , které tvoří nejkratší cestu z x do y , a ke grafu G přidáme kopie všech těchto hran. V grafu G' , který takto získáme, budou mít všechny vrcholy sudý stupeň.
6. V grafu G' sestrojíme eulerovský tah. Tento tah prochází všemi přidanými hranami, což odpovídá opakovaným průchodům hranami původního grafu.
7. V eulerovském tahu nahradíme všechny přidané hrany jim odpovídajícími hranami grafu původního. Tím získáme hledaný nejkratší sled, který prochází všemi hranami grafu G .

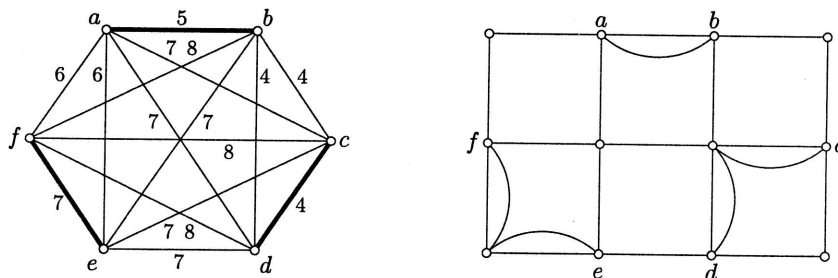
Důkaz správnosti tohoto algoritmu se opírá o fakt, že součet délek opakovaně procházených hran je roven ceně nejlevnějšího perfektního párování. Výpočet lze provést v čase $O(m + n^3)$.

10.3.3 Příklad. Řešme úlohu čínského poštáka pro graf z obrázku 10.4.



Obrázek 10.4: Graf k úloze čínského poštáka 10.3.3.

V grafu je šest vrcholů lichého stupně, $L = \{a, b, c, d, e, f\}$. Úplný graf K na množině vrcholů L je na obr. 10.5 vlevo. Hraný tvořící nejlevnější perfektní párování jsou v něm nakresleny silně. Na téže obrázku vpravo je nakreslen původní graf s přidanými hranami. Stupně všech vrcholů jsou v něm již sudé, není tedy problém najít v něm eulerovský tah.



Obrázek 10.5: Řešení úlohy čínského poštáka 10.3.3.

10.3.4 Cvičení. Dokažte, že sled, který je řešením úlohy čínského pošťáka, nemůže procházet žádnou hranou více než dvakrát.

10.3.5 Cvičení. Dokažte, že žádné dvě cesty, které ke grafu přidáváme v kroku 5 algoritmu 10.3.2, nemají společnou hranu. Jinými slovy, dokažte, že v grafu G' bude ke každé hraně grafu G přidána nejvýše jedna kopie. Návod: využijte faktu, že párování P je nejlevnější.

10.3.6 Poznámka. Úlohu čínského pošťáka jsme pro jednoduchost formulovali a řešili pouze pro kladné délky hran. Algoritmus 10.3.2 lze použít i na graf, který obsahuje hrany o nulové délce. Důkaz správnosti je v tomto obecnějším případě složitější, neboť je třeba počítat s tím, že hrana o nulové délce se může ve výsledném sledu vyskytovat i více než dvakrát. To však nevadí, na délku sledu to nemá vliv.

10.3.7 Cvičení. Lze algoritmus 10.3.2 použít i na graf, který obsahuje hrany o záporné délce?

K

N

V p
kte
věna p
-úpl
troc
s výmec
vyjárůzn
NP-
mož
v 11

11

Teor
zabý11.1
význ

uspo

mezi

Z
o alg

ého pošťáka,

áme v kroku
e v grafu G'
ružijte faktu,

t formulovali
a graf, který
šším případě
ve výsledném
o nemá vliv.

sahuje hrany

Kapitola 11

NP-těžké úlohy

V předchozích kapitolách jsme se (až na několik výjimek) zabývali úlohami, pro které jsou známy uspokojivě rychlé algoritmy. Následující dvě kapitoly se naopak věnují úlohám, které jsou (až na výjimky) „těžké“. Co to však jsou „těžké“ úlohy?

Prvý oddíl této kapitoly obsahuje velmi stručné shrnutí základních pojmů a poznatků teorie složitosti (míněno: složitosti výpočtů), zejména teorie tzv. NP-úplnosti. Uvádíme je jednak proto, aby naše výroky o obtížnosti úloh měly alespoň trochu solidní základ, jednak proto, že se v literatuře lze poměrně často setkat s výroky o NP-úplnosti některých úloh.

Výklad v tomto prvním oddíle je poněkud abstraktnější a vybočuje mimo rámec teorie grafů. Studium tohoto oddílu můžete přeskocit, smíříte-li se s vágním vyjádřením, že „NP-těžké úlohy jsou obtížně řešitelné“.

V oddílech 11.2 a 11.3 uvedeme dvě metody, které lze použít k řešení poměrně různorodých úloh (i negrafových). Obě metody se používají (zejména) k řešení NP-těžkých úloh, obě lze pokládat za „inteligentní průzkum všech myslitelných možností“ a v obou se prohledává tzv. strom řešení. Srovnání obou metod je uvedeno v 11.3.8. Obě metody využijeme v kapitolách 12 a 13.

11.1 Časová složitost

Teorie složitosti je část informatiky, která se (zjednodušeně řečeno a mimo jiné) zabývá závislostí doby řešení úlohy na velikosti instance úlohy.

11.1.1 Typ úlohy a instance úlohy. Slovo „úloha“ se používá ve dvou různých významech, mezi nimiž zde musíme pečlivě rozlišovat:

Typ úlohy určuje, jakým způsobem je úloha zadána, tj. jaká data a v jakém uspořádání budou tvořit zadání úlohy, co má být výsledkem a jaký má být vztah mezi výsledkem a zadáním. *Instance úlohy* je konkrétní případ úlohy daného typu.

Známe-li pouze typ úlohy, nemáme ještě co počítat, můžeme však přemýšlet o algoritmu, kterým bychom řešili instance tohoto typu úlohy, kdybychom nějaké

měli. Počítat má smysl, teprve když máme konkrétní zadání, tj. instanci úlohy. Viz též 1.11.3, str. 32, kde jsme tyto pojmy zavedli pro optimalizační úlohy.

11.1.2 Velikost instance úlohy. Data, která popisují instanci daného typu úlohy, tvoří vstup do algoritmu, který úlohy daného typu řeší, proto o těchto datech mluvíme jako o vstupních datech. Velikost instance pak měříme jako objem těchto vstupních dat. Obecně se používá počet bitů, ale v grafových úlohách, kde zadáním úlohy je obvykle graf (a možná nějaké jeho ohodnocení), je zvykem vyjadřovat velikost vstupních dat počtem vrcholů nebo počtem vrcholů a počtem hran (tj. dvěma čísly).

11.1.3 Doba práce algoritmu. To je tak trochu potíže. Doba výpočtu na konkrétním typu počítače se pro teoretické úvahy nehodí, protože počítače mívají různě rychlé procesory a jakékoli tvrzení by mělo velice omezenou platnost. Počet instrukcí, které je třeba při výpočtu vykonat, je trochu lepší, ale i on závisí na typu procesoru (procesory mívají různé soubory instrukcí).

Proto se v teoretických úvahách často používá tzv. *asymptotické vyjádření* doby práce algoritmu pomocí symbolu O (velké písmeno O). Připomeňme (viz 1.10.6, str. 30) jeho definici:

Nechť f, g jsou dvě nezáporné funkce reálné proměnné. Existují-li reálné konstanty k, m takové, že pro všechna $x > m$ platí $g(x) \leq kf(x)$, pak píšeme $g = O(f)$.

Zápis $g = O(f)$ se vyslovuje jako „ g je o f “. Vztah $g = O(f)$ je vlastně odhadem funkce g shora. Nezávisí na chování obou funkcí g a f pro „malé“ hodnoty argumentů (přesněji, pro $x \leq m$) a nezávisí také na násobení jedné z funkcí nějakou konstantou.

Řekneme-li, že nějaký algoritmus pracuje v čase $O(f(n))$, kde n je velikost instance úlohy, znamená to, že doba jeho práce na instancích od určité velikosti výše nepřesahuje nějaký násobek funkce f . Přitom je však možné, že pro mnoho konkrétních případů vstupních dat (třeba i pro všechny) bude tentýž algoritmus pracovat rychleji. Dokonce i kdyby platil lepší odhad, např. $O(n^2)$, byl by odhad $O(n^3)$ také pravdivý.

Vyjádření doby práce algoritmu symbolem O se nezískává měřením doby skutečných výpočtů na počítači, ale teoretickým rozborem činnosti algoritmu.

Dodejme, že je třeba rozlišovat, zda máme na mysli dobu práce v nejhorším případě nebo dobu práce průměrnou. Není výjimkou, když nějaký algoritmus má odhad průměrné doby práce řádově lepší než odhad pro nejhorší případ. Odhady pro nejhorší případ jsou jednodušší (snáze odvoditelné). V této knize se téměř výhradně zabýváme jen odhady pro nejhorší případ.

Asymptotické vyjádření doby práce pomocí symbolu O má své výhody i nevýhody. Výhodou je to, že se zaměřuje na chování algoritmu na velkých instancích úloh. S instancemi malého rozsahu totiž zpravidla nejsou problémy. Další výhodou je nezávislost na typu a rychlosti počítače, na programovacím jazyce, překladači, schopnostech programátora apod.

Nevýhoda je v tom, že z asymptotického odhadu není jasné, která instance úlohy je velká. Může se snadno stát, že algoritmus s horším asymptotickým odhadem

bude na grafu o řekněme 5000 vrcholech rychlejší než jiný algoritmus, který je sice asymptoticky lepší, ale jeho rychlost se projeví až na grafech mnohem větších.

11.1.4 Rozhodovací úlohy jsou takové úlohy, jejichž výsledek je buď „ano“, nebo „ne“.

Mnohé (ovšem zdaleka ne všechny) rozhodovací úlohy jsou odvozeny z optimalizačních úloh jako jejich rozhodovací verze: k zadání úlohy se přidá konstanta K a ptáme se, zda existuje přípustné řešení s hodnotou účelové funkce, která je lepší nebo rovna K . Zvláštním případem rozhodovací verze pak je dotaz, zda existuje vůbec nějaké přípustné řešení dané úlohy (stačí vhodně zvolit konstantu K). Rozhodovací verze úlohy určitě není těžší než sama optimalizační úloha.

11.1.5 Polynomiálně řešitelné úlohy jsou úlohy, pro něž existuje algoritmus, který danou úlohu řeší, přičemž doba práce je pro nejhorší případ shora omezena asymptoticky nějakým polynomem, tj. doba práce je $O(p(n))$, kde n je velikost vstupních dat a p je nějaký polynom. Říkáme, že takový algoritmus úlohu řeší v *polynomiálním čase* nebo že je *polynomiální*.

Třída úloh P je tvořena polynomiálně řešitelnými rozhodovacími úlohami.

Polynomiálně řešitelné úlohy jsou obvykle pokládány za „snadno řešitelné“. Pro mnoho úloh však polynomiální algoritmus není znám, a to navzdory dlouhému úsilí nejlepších odborníků.

11.1.6 Třída úloh NP. Rozhodovací úloha patří do třídy NP, když pro ni existuje tzv. *ověřovací algoritmus* \mathcal{V} , který má tyto vlastnosti:

- jeho vstupem jsou data d popisující instanci úlohy a tzv. *certifikát*, což je jakýsi blíže neurčený kus dat, jehož velikost je shora omezena nějakým pevně daným polynomem vzhledem k délce vstupních dat d ,
- pracuje v polynomiálním čase vzhledem k velikosti vstupních dat d a dává vždy odpověď „ano“ nebo „nevím“,
- pro instanci dané úlohy d , pro kterou je správná odpověď „ano“, existuje certifikát c takový, že ověřovací algoritmus $\mathcal{V}(d, c)$ dá odpověď „ano“,
- pro instanci úlohy d , pro kterou je správná odpověď „ne“, dává ověřovací algoritmus vždy (pro všechny možné certifikáty) odpověď „nevím“.

Lidově řečeno, rozhodovací úloha patří do třídy NP, když kladnou odpověď lze potvrdit tím, že uhadneme vhodný certifikát a pak v polynomiálním čase ověříme, že jsme hádali správně. Záporná odpověď se nepotvrzuje, pouze se požaduje, aby v tomto případě neexistoval žádný falešný certifikát, který by potvrdil nesprávnou odpověď.

V typických NP-úlohách se ptáme, zda existuje něco, co se možná i těžko hledá, ale snadno se ověřuje, že jsme to našli.

Konkrétním příkladem je úloha, zda v daném grafu existuje kružnice, která prochází všemi vrcholy grafu (tzv. hamiltonovská kružnice, viz kapitola 12). Certifikátem zde je ona kružnice. Ověřovací algoritmus ověřuje, že to je opravdu kružnice a že opravdu prochází přes všechny vrcholy grafu.

Poznamenejme, že u NP-úloh je velmi důležité, jak je položena otázka, která se v úloze řeší. Je to proto, že certifikát a potvrzení požadujeme pouze pro kladnou odpověď. Obrácením otázky u nějaké NP-úlohy tedy dostaneme úlohu zcela jinou, která již do třídy NP nemusí patřit (může být těžší).

Každá úloha ze třídy P je zároveň ve třídě NP, neboť polynomiální algoritmus, který úlohu řeší, lze pokládat za algoritmus ověřovací, u něhož na certifikátu nezáleží. Platí tedy $P \subseteq NP$. V současné době není známo (nikdo neumí matematicky dokázat), zda platí rovnost, ale všeobecně se věří, že $P \neq NP$.

Ve třídě NP jsou úlohy různé obtížnosti. Než si řekneme, které jsou nejtěžší, musíme umět porovnávat obtížnost úloh.

11.1.7 Porovnávání obtížnosti, P-redukce. Úloha A je *P-redukovatelná* (též *polynomiálně redukovatelná*) na úlohu B , existuje-li polynomiální algoritmus, který ze vstupních dat úlohy A vyrobí vstupní data pro úlohu B , a to tak, že odpovědi na obě instance jsou stejné. Doba řešení úlohy B při tom nehraje žádnou roli.

Existuje-li P-redukce úlohy A na úlohu B , chápeme to jako doklad, že úloha A je lehčí nebo stejně obtížná jako úloha B . Pokud existují P-redukce v obou směrech (A na B a také B na A), pokládáme obě úlohy za stejně obtížné. Díky P-redukčním existují ve třídě NP skupiny stejně obtížných úloh. Třída P je jednou z nich.

Jestliže úloha A je P-redukovatelná na úlohu B a $B \in P$, pak také $A \in P$.

Několik příkladů P-redukcí je uvedeno v 12.2, str. 199.

11.1.8 NP-úplná úloha (anglicky NP-complete) je taková úloha, která patří do NP a je ve třídě NP nejtěžší v tom smyslu, že jakákoli jiná NP-úloha je na ni P-redukovatelná. Třidu všech NP-úplných úloh značíme NP-C.

NP-úplných úloh je známo mnoho (kniha [GJ79] jich uvádí přes 300), řada z nich jsou grafové úlohy. V naší knize se s NP-úplnými úlohami setkáme zejména v kapitolách 12 a 13.

Všechny NP-úplné úlohy jsou z hlediska P-redukce stejně těžké. Kdyby se podařilo najít polynomiální algoritmus pro kteroukoli z nich, měli bychom ihned polynomiální algoritmy pro všechny NP-úlohy a platilo by $P = NP$. V podstatě nikdo nevěří, že se takový algoritmus najde, ale dokázat to prozatím nikdo nedovede.

11.1.9 NP-těžká úloha (anglicky NP-hard) je taková úloha B , na kterou lze P-redukovat nějakou NP-úplnou úlohu A .

Na rozdíl od NP úplnosti zde nepožadujeme příslušnost úlohy B ke třídě NP. Díky tomu do třídy NP-těžkých úloh patří i optimalizační úlohy, jejichž rozhodovací verze jsou NP-úplné. Mezi NP-těžké však patří i mnohé další, daleko obtížnější úlohy.

11.1.10 Pozor na speciální případy. Je běžné, že nějaká úloha je NP-těžká, ale některé její speciální případy jsou snadno (polynomiálně) řešitelné.

Například úloha najít v daném grafu kružnici, která prochází všemi vrcholy grafu (tzv. hamiltonovská kružnice, viz kapitola 12), je notoricky známa jako NP-těžká,

ale kdybychom se omezili pouze na grafy, které mají stejný počet vrcholů i hran, dostali bychom úlohu (typ úlohy), která je triviálně řešitelná v lineárním čase.

Na tuto záludnost byste měli dát pozor, až se budete vmlouvat, že nějakou úlohu neumíte řešit, protože je NP-těžká.

11.1.11 Praktický význam teorie NP-úplnosti. Celá teorie NP-úplnosti působí dost pesimistickým dojmem: pokud věříme, že $P \neq NP$, pak není naděje, že by existovaly polynomiální (a tedy rychlé) algoritmy pro celou řadu velmi praktických úloh.

Reálné využití této teorie je tedy v podstatě alibistické: pokud o nějaké úloze umíme dokázat, že je NP-těžká, a věříme-li, že $P \neq NP$, pak to, že pro naši úlohu nedovedeme najít polynomiální algoritmus, není žádná ostuda, protože to neumí nikdo na světě včetně těch nejchytřejších odborníků.

To ovšem vůbec neznamená, že NP-úplné a NP-těžké úlohy nelze řešit. Musíme však buď slevit z požadavků na rychlost (algoritmus nebude polynomiální, ale stále ještě může být prakticky použitelný), nebo použijeme nějaký heuristický algoritmus, který sice bývá rychlý, ale musíme se smířit s rizikem, že nedosáhneme vždy požadovaného výsledku (např. najdeme řešení, které není optimální, nebo nenajdeme přípustné řešení, i když ve skutečnosti existuje).

Úlohy (instance) malého rozsahu lze téměř vždy řešit tzv. *metodou hrubé síly*, což je vznešený název pro jednoduché vyzkoušení všech možností, které přicházejí v úvahu.

Pro mnoho úloh lze použít backtracking 11.2 nebo metodu větví a mezí 11.3. Tyto metody lze pokládat za inteligentní vyzkoušení všech možností, kdy se dosahuje někdy i značného zrychlení tím, že se nezkoumají možnosti, které nemohou přispět k řešení.

11.2 Backtracking

11.2.1 Posloupnosti a strom řešení. Backtracking lze použít v těch situacích, kdy každé řešení úlohy (tj. každý prvek prostoru řešení) můžeme považovat za konečnou posloupnost, přičemž každý prvek posloupnosti musí být vybrán z nějaké předem dané konečné množiny. Při troše fantazie se na většinu kombinatorických úloh můžeme dívat tímto způsobem.

Kromě posloupností, které představují (úplné) řešení úlohy budeme uvažovat také všechny jejich počáteční úseky. Tyto počáteční úseky odpovídají částečným (neúplně určeným) řešením. Všechny takovéto posloupnosti můžeme pokládat za vrcholy orientovaného grafu, v němž každá částečná posloupnost má jako své následníky všechna svá prodloužení o jeden prvek. Tento graf je kořenovým stromem (kořenem je prázdná posloupnost) a nazýváme jej *stromem řešení*.

11.2.2 Backtracking. Nejjednodušší verze backtrackingu spočívá v prohledávání stromu řešení do hloubky. Vždy, když se při prohledávání dostaneme k posloup-

nosti, která odpovídá úplnému řešení, provedeme test, zda toto řešení je přípustné. U optimalizační úlohy navíc evidujeme nejlepší dosud nalezené přípustné řešení.

Takto jednoduchý backtracking bez použití dalších triků ovšem není v podstatě ničím jiným než systematicky provedenou metodou hrubé síly. Triky, kterými lze výpočet mnohdy i značně urychlit, spočívají v tom, že vynecháme prohledávání některých podstromů.

Má smysl prodlužovat jen ty posloupnosti, u nichž je naděje, že mohou být prodlouženy na přípustné řešení. Pokud jsme tedy schopni s jistotou poznat, že žádné prodloužení už nemůže být přípustné, můžeme ušetřit čas tím, že prohledávání příslušného podstromu přeskočíme (provedením návratu).

Řešíme-li optimalizační úlohu, můžeme dosáhnout další úspory času, budeme-li schopni pro každou posloupnost x_1, \dots, x_i určit číslo D , které nazveme dolním, popř. horním *odhadem* (pro minimalizační, popř. maximalizační úlohu) a které má tu vlastnost, že prodlužováním posloupnosti x_1, \dots, x_i nelze dosáhnout přípustného řešení lepšího než D . Bude-li tedy pro posloupnost x_1, \dots, x_i tento odhad horší než nějaké přípustné řešení, které už v té chvíli známe, nemá další prodlužování posloupnosti x_1, \dots, x_i smysl a prohledávání příslušného podstromu můžeme opět přeskočít.

I přes uvedené možnosti úspor bývá backtracking časově hodně náročný.

Příkladem jednoduchého backtrackingu je algoritmus 3.3.10, str. 56, další příklady uvedeme v kapitolách 12 a 13. Backtracking se však běžně používá i k řešení negrafových úloh.

11.2.3 Příklad. Metodou backtrackingu řešme tuto úlohu: Najděte maximum funkce $4x_1 + 3x_2 + 2x_3 + x_4$ za podmínek $2x_1 + 3x_2 + x_3 + 3x_4 \leq 6$, $x_1, x_2 \in \{0, 1, 2\}$, $x_3, x_4 \in \{0, 1\}$.

Tuto úlohu lze samozřejmě řešit i mnoha jinými (a lepšími) postupy. Zde nám však jde o ukázkou postupu, který je „ve stylu backtrackingu“.

Řešení úlohy, tj. vektor (x_1, x_2, x_3, x_4) pokládáme za posloupnost celých čísel. Dílčí posloupnosti budeme prodlužovat zleva doprava a k prodlužování budeme používat přednostně nejvyšší povolené hodnoty.

Poněvadž všechny proměnné mohou nabývat jen nezáporných hodnot, můžeme při prohledávání stromu řešení do hloubky provést návrat, jakmile součet hodnot na levé straně nerovnosti přesáhne číslo 6. Takovou posloupnost totiž nelze doplnit na přípustné řešení.

Horní odhad budeme pro každé částečné řešení počítat tak, že částečné řešení doplníme nejvyššími možnými hodnotami na délku úplného řešení a tuto posloupnost dosadíme do účelové funkce (bez ohledu na omezující podmínku). Např. posloupnost $(1, 2)$ doplníme na $(1, 2, 1, 1)$, odhad tedy je 13. Je zřejmé, že vyšší hodnoty prodlužováním posloupnosti $(1, 2)$ nelze dosáhnout, ve skutečnosti nedosáhneme ani oněch 13.

Pro stručnost uvedeme jen ty posloupnosti, ze kterých provádíme návrat:

$(2, 2)$	nepřípustné,
$(2, 1)$	nepřípustné,

(2, 0, 1, 1)	nepřípustné,
(2, 0, 1, 0)	úplně přípustné řešení s hodnotou 10,
(2, 0, 0)	odhad $9 < 10$,
(1, 2)	nepřípustné,
(1, 1, 1, 1)	nepřípustné,
(1, 1, 1, 0)	odhad $9 < 10$,
(1, 1, 0)	odhad $8 < 10$,
(1, 0)	odhad $7 < 10$,
(0)	odhad $9 < 10$.

Optimálním řešením je posloupnost (2, 0, 1, 0) s hodnotou účelové funkce 10. Všimněte si, že toto řešení jsme našli poměrně brzy. Zbytek výpočtu však byl nutný pro ověření, že neexistuje lepší řešení.

11.3 Metoda větví a mezí

Metoda větví a mezí se používá pouze k řešení optimalizačních úloh. Podobně jako u backtrackingu se zde prohledává strom řešení, ten však bývá definován jiným, složitějším způsobem.

V tomto oddíle podáme obecný výklad metody větví a mezí a uvedeme příklad použití na negrafové úloze. Příklady použití na úlohy teorie grafů jsou uvedeny v 12.4, str. 203.

11.3.1 Větvení množiny přípustných řešení. Množinu přípustných řešení úlohy U označme symbolem $Př(U)$.

Mějme optimalizační úlohu U s účelovou funkcí f . Úlohu U můžeme řešit (mimo jiné) tak, že vytvoříme úlohy U_1, \dots, U_n se stejnou účelovou funkcí (tzv. *podúlohy* úlohy U) takové, že

$$Př(U) = Př(U_1) \cup \dots \cup Př(U_n).$$

Optimální řešení úlohy U pak lze získat tak, že vybereme nejlepší ze všech optimálních řešení podúloh U_1, \dots, U_n . Samozřejmě se při tom snažíme (v tom je smysl větvení), aby podúlohy U_1, \dots, U_n byly v nějakém smyslu snazší než původní úloha U , tj. aby např. byly menší. Dodejme, že při tom je výhodné, když množiny $Př(U_1), \dots, Př(U_n)$ jsou navzájem disjunktní, ale není to nutné.

Nalezení optimálního řešení úlohy U je snadné, pokud pro každou podúlohu U_i :

1. buď najdeme optimální řešení úlohy U_i ,
2. nebo prokážeme, že úloha U_i nemá žádné přípustné řešení,
3. nebo prokážeme, že žádné přípustné řešení úlohy U_i není lepší než nějaké v té době již známé přípustné řešení úlohy U .

Často se ovšem stává, že pro některou podúlohu (a často ne jen jednu) nejsme schopni přímo zjistit ani jednu z výše uvedených tří možností. V tom případě pro každou takovou podúlohu, označme ji U_i , použijeme stejný postup jako pro

úlohu U , tj. rozvětvíme ji na podúlohy U_{i_1}, \dots, U_{i_m} a pro takto získané podúlohy se opět snažíme zjistit některou z výše uvedených možností. Některé z takto získaných podúloh může být nutno dále větvit.

11.3.2 Strom řešení. Jednotlivé podúlohy můžeme pokládat za vrcholy kořenového stromu, který nazýváme stromem řešení. Kořenem je výchozí úloha U a z každé úlohy, která byla větvena na podúlohy, vedou hrany ke všem jejím podúlohám. Během výpočtu se strom řešení mění (zvětšuje).

Pro výpočet jsou zajímavé pouze listy stromu řešení. Listy dělíme na tzv. *živé* a *mrtvé*. Za mrtvé pokládáme ty listy (podúlohy), u nichž se již podařilo prokázat některou ze tří možností uvedených v 11.3.1. Těmito podúlohami se již není třeba dále zabývat. Ostatní listy (podúlohy) jsou živé.

Živou podúlohu je třeba buď umrtvit (zjištěním některé ze tří možností z 11.3.1), anebo rozvětvit. Rozvětvením podúloha přestává být listem, ve stromě řešení se však místo ní objeví několik nových listů. Výpočet končí v okamžiku, kdy všechny listy stromu řešení jsou mrtvé.

11.3.3 Odhady a meze. Pokud pro některou podúlohu U_i zjistíme, že platí možnost 3, tj. pokud prokážeme, že žádné přípustné řešení úlohy U_i není lepší než nějaké v té době již známé řešení úlohy U , je to při řešení dobrá zpráva, protože tuto podúlohu není třeba větvit. Nejlepší dosud známé řešení úlohy U se však během výpočtu obvykle mění (zlepšuje), a tak se snadno stane, že možnost 3 se o nějaké podúloze podaří prokázat, teprve až najdeme dostatečně dobré řešení celé úlohy U .

Proto bývá výpočet uspořádán tak, že jednak evidujeme nejlepší dosud známé řešení a hodnotu jeho účelové funkce, jednak pro každou podúlohu U_i vypočteme číslo $\text{Odh}(U_i)$ takové, že žádné přípustné řešení úlohy U_i není lepší než $\text{Odh}(U_i)$.

Číslo $\text{Odh}(U_i)$ bývá v literatuře nazýváno u minimalizačních úloh *dolním odhadem* a u maximalizačních úloh *horním odhadem*. Čím je odhad blíže skutečné hodnotě optimálního řešení podúlohy, tím je tento odhad užitečnější, neboť tím snáze jeho pomocí prokážeme, že pro podúlohu platí možnost 3. Odhad je tedy tím lepší, čím je méně optimistický. Někdy lze volit mezi několika metodami výpočtu odhadu, které dávají různě kvalitní výsledky.

Způsob výpočtu odhadu samozřejmě velice závisí na řešené úloze a na způsobu vytváření podúloh. Jako odhad pro podúlohu U_i se často používá hodnota optimálního řešení tzv. relaxované podúlohy:

11.3.4 Relaxace nepříjemných omezujících podmínek. Poněvadž množina přípustných řešení úlohy je vždy popsána (zadána) pomocí omezujících podmínek, provádí se i větvení úlohy na její podúlohy pomocí dodatečných omezujících podmínek, které se k větvené úloze přidávají.

Obvyklá situace je tato: Omezující podmínky výchozí úlohy U lze rozložit na dvě skupiny, nazvěme je *příjemné* a *nepříjemné*. Vynecháme-li (takzvaně *relaxujeme*) nepříjemné omezující podmínky, dostaneme tzv. relaxovanou úlohu, která má již jenom příjemné omezující podmínky a kterou dovedeme (pokud možno rychle) řešit.

Najdeme-li optimální řešení r úlohy, která vznikla relaxací z U , jsou dvě možnosti: buď řešení r splňuje, nebo nesplňuje nepříjemné omezující podmínky. Pokud splňuje, pak máme štěstí, protože řešení r je zároveň optimálním řešením i samotné úlohy U , a úlohu U tedy není třeba větvit. Obvykle však štěstí nemáme a řešení r nepříjemné omezující podmínky nesplňuje. V takové situaci uděláme dvě věci: za prvé, hodnotu optimálního řešení relaxované úlohy můžeme použít jako odhad, a za druhé, platnost oněch nepříjemných podmínek si snažíme vynutit přidáváním podmínek příjemných, a to i za cenu toho, že se nám úloha rozpadne (rozvětví) na několik podúloh.

Podstatné je, že při větvení úlohy přidáváme vždy jen příjemné omezující podmínky. Tím je totiž zaručeno, že takto vzniklé podúlohy jsou stejného typu jako větvená úloha, a můžeme tedy opět použít onen trik s relaxací a řešením relaxované podúlohy.

Které omezující podmínky můžeme pokládat za příjemné, je dáno naší schopností rozumně rychle řešit příslušné relaxované podúlohy. Zpravidla pro danou úlohu existuje několik možností, jak zvolit prostor řešení a jak přeformulovat omezující podmínky, přitom obojí má značný vliv na pracnost řešení metodou větví a mezi. Příklady uvedeme v 12.4.

11.3.5 Volba podúlohy k větvení by mohla být založena na prohledávání stromu řešení např. do šířky nebo do hloubky. Častěji se však pro každou podúlohu vypočte hodnota, nazvěme ji prioritou, a mezi živými podúlohami vybíráme k větvení tu, která má prioritu nejlepší. Běžně se v roli priority používá (dolní nebo horní) odhad. Vybíráme pak podúlohu, která má odhad nejlepší (nejnadějnější), neboť u této podúlohy lze očekávat, že zlepšíme nejlepší dosud nalezené řešení a pomocí odhadů pak umrtvíme některé dosud živé podúlohy.

11.3.6 Shrnutí metody větví a mezi. Při výpočtu udržujeme nejlepší dosud nalezené přípustné řešení \tilde{r} dané úlohy U a jeho hodnotu účelové funkce L . Na začátku \tilde{r} není definováno a jako hodnotu L použijeme nejhorší možnou hodnotu ($+\infty$ nebo $-\infty$).

Doporučuje se však ještě před zahájením výpočtu metodou větví a mezi získat nějakým heuristickým postupem (viz 11.4) co nejlepší přípustné řešení \tilde{r} a jeho hodnotu L , neboť to může následující výpočet urychlit.

Dále při výpočtu udržujeme seznam živých podúloh.

Obvykle ještě před zařazením do seznamu pro každou podúlohu vypočteme odhad, nejčastěji řešením relaxované verze příslušné podúlohy. Pokud přitom dostaneme přípustné řešení, porovnáme je s řešením \tilde{r} a dále uchováme lepší z nich. Pokud bylo řešení \tilde{r} nahrazeno lepším, tj. pokud se zlepšila hodnota L , projdeme seznam živých podúloh a vyřadíme ty, které jsou díky nové hodnotě L umrtveny.

Pokud podúloha nemá žádné přípustné řešení nebo pokud pro ni dostaneme odhad, který je horší než L , pak tuto podúlohu okamžitě umrtvíme a do seznamu živých podúloh ji vůbec nezapisujeme.

Ze seznamu živých podúloh vybíráme zpravidla tu, která má nejlepší odhad. Tuto podúlohu ze seznamu odstraníme, rozvětvíme ji na podúlohy, pro každou z nich

ihned vypočteme odhad a případně ji zařadíme do seznamu živých podúloh, jak bylo popsáno výše.

Tento postup opakujeme, dokud je v seznamu nějaká živá podúloha. Výsledné řešení \tilde{r} pak je optimálním řešením úlohy.

11.3.7 Příklad. Postup výpočtu metodou větví a mezí předvedeme na problému batohu (viz 2.4.2, str. 44), tj. na negrafově úloze, aby vynikla obecná použitelnost metody. Další příklady viz 12.4, str. 203.

Mějme batoh o kapacitě 40 a pět předmětů s těmito vahami a cenami:

předmět i :	A	B	C	D	E
cena c_i :	50	107	31	31	69
váha v_i :	14	30	9	10	27

Jako prostor řešení vezmeme \mathbb{R}^5 , jeho prvky jsou uspořádané pětice čísel x_1, \dots, x_5 , která určují, kolikrát zabalíme do batohu jednotlivé předměty. Omezující podmínky určují, že $0 \leq x_i \leq 1$ a $\sum_{i=1}^5 v_i x_i \leq 40$. Nepříjemnou omezující podmínkou zde je, že všechna x_i musí být navíc celočíselná.

Pro účely snadného řešení relaxované úlohy si předměty seřadíme sestupně podle jejich „měrné ceny“, tj. podle podílu cena/váha. Ve výše uvedeném zadání je toto seřazení již provedeno.

Optimální řešení relaxované úlohy pak získáme velmi jednoduše tak, že do batohu budeme ukládat celé předměty v pořadí klesající měrné ceny, přičemž poslední z takto zařazených předmětů možná nebude zařazen celý. Pro naši úlohu tak dostaneme řešení o hodnotě 142,73 tvořené celým předmětem A a částí 26/30 předmětu B. Jako horní odhad použijeme celou část této hodnoty, tj. zde 142, neboť všechna přípustná řešení původní (nerelaxované) úlohy U mají hodnotu účelové funkce celočíselnou, a ta tedy nemůže přesáhnout 142.

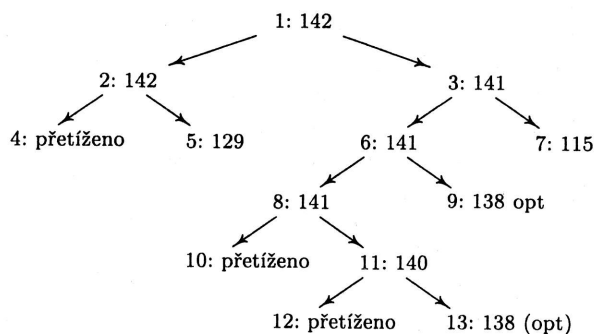
Větvení budeme provádět vždy na dvě podúlohy, a to tak, že si vybereme některý předmět, o němž v dané podúloze není ještě rozhodnuto, a v jedné z podúloh jej pevně zařadíme, zatímco ve druhé jej zakážeme. V obou podúlohách se tedy rozhoduje o menším počtu předmětů. Přidané omezující podmínky, tj. příkazy a zákazy, budeme zkráceně označovat symboly + a - tak, že např. --+. bude znamenat, že 1. a 3. předmět jsou zakázány, 2. předmět je přikázán a o ostatních předmětech není v podúloze rozhodnuto.

Průběh řešení je zaznamenán v následující tabulce. Podúlohy jsou očíslovány pořadovými čísly v pořadí, jak byly vytvořeny větvením. K větvení byla vybírána vždy podúloha s nejlepším odhadem, a to v pořadí 1, 2, 3, 6, 8, 11.

číslo podúlohy	přidaná omezení	horní odhad	komentář
1	142	větveno podle A na podúlohy 2 a 3
2	+....	142	větveno podle B na podúlohy 4 a 5
3	-....	141	větveno podle B na podúlohy 6 a 7

4	++...	-	přetíženo
5	+-....	129	umrtveno po zpracování podúlohy 9
6	-+...	141	větveno podle C na podúlohy 8 a 9
7	--...	115	umrtveno po zpracování podúlohy 9
8	---+.	141	větveno podle D na podúlohy 10 a 11
9	---..	138	(optimum) viz komentář v textu
10	----.	-	přetíženo
11	---.-	140	větveno podle E na podúlohy 12 a 13
12	----+	-	přetíženo
13	-----	138	další optimum

Strom řešení je na obrázku 11.1.



Obrázek 11.1: Strom řešení k příkladu 11.3.7. U vrcholů jsou napsána čísla podúloh a hodnoty horních odhadů.

Podúloha 9 vyžaduje podrobnější komentář: Řešením relaxované verze podúlohy 9 jsme zde (náhodou) dostali celočíselné řešení. Získali jsme tedy přípustné řešení celé úlohy, aniž by bylo nutno podúlohu větvit. V té době to bylo první přípustné řešení, které jsme našli. Použili jsme je ihned k umrtvení podúloh 5 a 7, které v té době byly ještě živé, ale měly ve srovnání s podúlohou 9 horší odhad.

Stojí za zmínku, že v této chvíli jsme již měli řešení, o kterém se později ukázalo, že je optimální, ale v té době jsme to ještě nevěděli, protože stále ještě zbývala živá podúloha 8 s odhadem, který dával naději na zlepšení. Bylo tedy nutno ve výpočtu pokračovat a teprve po rozvětvení podúloh 8 a 11 se ukázalo, že lepší řešení neexistuje.

11.3.8 Backtracking versus větve a meze. Obě metody mají některé rysy společné a u některých algoritmů je těžké říci, o kterou z obou metod jde.

Backtracking použitý k řešení optimalizační úlohy lze pokládat za speciální případ metody větví a mezí. Na posloupnost, která v backtrackingu tvoří částečné

řešení, se totiž můžeme dívat jako na podúlohu, v níž několik prvních členů posloupnosti je pomocí přidáných omezujících podmínek pevně určeno. Prodloužení posloupnosti pak znamená přidání nové omezující podmínky.

Při backtrackingu však prohledáváme strom řešení vždy do hloubky (jinak by to nebyl backtracking), zatímco v metodě větví a mezi lze další postup řešení volit svobodněji, a tedy zpravidla výhodněji (např. pomocí priorit, viz 11.3.5). V backtrackingu lze k zamezení zbytečného větvení použít i jiné triky než dolní, popř. horní odhady. A konečně, backtracking lze použít také k řešení úloh, které nemají optimalizační charakter (viz např. 13.3.1).

11.4 Heuristické algoritmy

V praxi mnohdy vystačíme s řešením, které je „dostatečně dobré“, které však získáme rychle. Algoritmy, které poskytují takováto řešení, nazýváme *heuristické*. Tyto algoritmy bývají založeny na nejrůznějších principech.

Mnohé heuristické algoritmy pracují tzv. *hladovým* způsobem: řešení vytváří (konstruuje) postupně a v každém kroku se snaží o co největší zlepšení účelové funkce, popř. o co největší úsporu. Samozřejmě se může stát, že se „chamtivost nevyplatí“ a výsledkem je řešení, které není optimální. Úlohy, v nichž hladový postup zaručeně vede k optimálnímu řešení, jsou spíše výjimečné, viz poznámka u algoritmu 4.3.7, str. 62, pro minimální kostru.

Některé heuristické algoritmy řešení mnohokrát opakují s využitím náhody a z takto získaných řešení vybírají nejlepší.

Další heuristické algoritmy využívají tzv. *lokální průzkum*: vezmou nějaké stávající řešení (zpravidla získané nějakou jinou heuristikou) a systematicky je pozměňují, tj. v jistém smyslu prozkoumávají okolí stávajícího řešení. Pokud některé z pozměněných řešení je lepší než řešení stávající, prohlásí toto lepší řešení za stávající a pokračují průzkumem okolí tohoto nového řešení. Toto se opakuje, dokud se daří řešení zlepšovat. Samozřejmě můžeme skončit i dříve, pokud najdeme řešení, jehož kvalita nám postačuje nebo pokud už nemáme čas na další zlepšování.

Metodu lokálního průzkumu lze dále zdokonalit s využitím náhody. Jednak můžeme s nějakou pravděpodobností akceptovat i zhoršení účelové funkce v naději, že takto později objevíme celkově lepší řešení, jednak můžeme celý postup mnohokrát opakovat z jiného, náhodně zvoleného výchozího řešení.

Další, složitější heuristické algoritmy jsou „ošízené“ varianty backtrackingu nebo metody větví a mezi, v nichž používáme nepřesné (ne zcela spolehlivé) odhady. Prohledáváme pak menší strom řešení a výpočet se může podstatně urychlit, ale může se stát, že díky nepřesnému odhadu přeskochíme prohledávání podstromu, který obsahuje optimální řešení.

Obecně platí, že heuristické algoritmy bývají „šity na míru“ pro určitý typ úlohy, jejich úspěšnost závisí na tom, jak se podaří využít specifických rysů daného typu úlohy i specifických rysů instancí, které chceme řešit. Při jejich návrhu hodně záleží i na zkušenosti a citu pro řešený problém.

ých členů po-
o. Prodloužení

bky (jinak by
postup řešení
t, viz 11.3.5).
íky než dolní,
ní úloh, které

F, které však
e heuristické.

řešení vytváří
pšení účelové
e „chamtivost
nichž hladový
viz poznámka

žitím náhody

nějaké stáva-
je pozměňují,
teré z pozmě-
ní za stávající
dokud se daří
řešení, jehož

. Jednak mů-
ce v naději, že
up mnohokrát

rackingu nebo
alivé) odhady.
ě urychlit, ale
stromu, který

řítý typ úlohy,
a daného typu
u hodně záleží

Kapitola 12

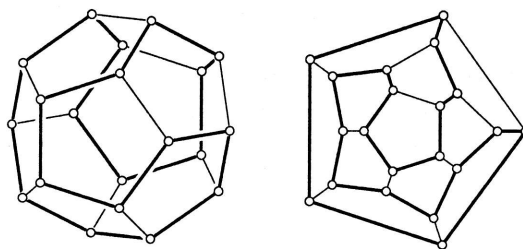
Hamiltonovské cesty a kružnice

12.1 Základní pojmy, aplikace a vzájemné převody

12.1.1 Definice. *Hamiltonovská cesta* v grafu G je taková cesta, která obsahuje všechny vrcholy grafu G .

Podobně definujeme *hamiltonovskou kružnici* a *hamiltonovský cyklus* jako kružnici nebo cyklus procházející přes všechny vrcholy grafu.

12.1.2 Hamiltonova hádanka. Výše uvedené pojmy jsou nazvány podle irského matematika W. R. Hamiltona, který v polovině 19. století zveřejnil v zeměpisném časopise hádanku, v níž šlo o nalezení uzavřené trasy vedoucí přes dvacet měst, která byla rozmístěna ve vrcholech pravidelného dvanáctistěnu, přičemž se přes žádné město nesmělo cestovat dvakrát. Příslušný graf je na obr. 12.1.



Obrázek 12.1: Dvě nakreslení grafu pravidelného dvanáctistěnu s vyznačenou hamiltonovskou kružnicí. Obrázek vpravo lze chápat jako „širokouhlý“ pohled dovnitř skrz přední stěnu dvanáctistěnu.

12.1.3 Typy hamiltonovských úloh. Úlohy, ve kterých jde o hamiltonovské pojmy, mají mnoho variant a lze je třídit z několika různých hledisek.

Především je třeba odlišit úlohy existenční a optimalizační. V existenčních úlohách jde o zjištění existence nebo dokonce o nalezení hamiltonovské cesty, kružnice nebo cyklu v daném grafu.

V optimalizačních úlohách jsou hrany grafu ohodnoceny délkami a požaduje se nalezení hamiltonovské cesty, kružnice nebo cyklu o nejmenší délce. Optimální řešení se přitom často hledá v úplném grafu, protože obecnější úlohy lze na tento speciální případ velmi snadno převést (viz 12.2.2).

Dále úlohy dělíme na orientované a neorientované podle toho, zda hledáme orientovanou nebo neorientovanou hamiltonovskou cestu, případně zda hledáme hamiltonovský cyklus nebo kružnici.

V případech hamiltonovských cest musíme navíc rozlišit tři až čtyři varianty podle toho, zda je v zadání úlohy předepsán vrchol, ve kterém musí hamiltonovská cesta začínat, a vrchol, kde musí končit.

Ve všech těchto úlohách se můžeme omezit na prosté grafy bez smyček.

12.1.4 Obtížnost hamiltonovských úloh. Všechny výše uvedené typy hamiltonovských úloh jsou pro obecné grafy NP-těžké.

Stojí za zmínku, že kdybychom tyto úlohy „maličko“ pozměnili a namísto sledu, který obsahuje přesně jedenkrát každý vrchol, požadovali sled, který obsahuje přesně jedenkrát každou hranu, dostali bychom snadno řešitelné úlohy, kterými jsme se zabývali v kapitole 10 o eulerovských tazích.

12.1.5 Problém obchodního cestujícího. Jde o nalezení nejkratší hamiltonovské kružnice v úplném neorientovaném grafu, jehož hrany jsou ohodnoceny délkami.

Název (a popularita) této úlohy je založen na následující představě: obchodní cestující má za úkol navštívit v libovolném pořadí n měst a vrátit se zpět tak, aby jeho trasa byla co nejkratší. Přitom se předpokládá, že vzdálenosti mezi všemi dvojicemi měst jsou předem známé a symetrické (v obou směrech je vzdálenost stejná).

Často se mluví také o orientované variantě problému obchodního cestujícího – hledá se nejkratší hamiltonovský cyklus v úplném orientovaném grafu.

12.1.6 Příklad: Dopravní úlohy. Problém obchodního cestujícího má četné aplikace, v nichž jde o optimalizaci pohybu dopravního prostředku, např. při rozvozu zboží, vybírání poštovních schránek apod. Stejný problém se však vyskytuje např. při vrtání velkého počtu děr do desky s plošnými spoji pomocí číslicově řízené vrtačky (minimalizace přesunů).

V některých případech jde o nalezení (neuzavřeně) hamiltonovské cesty. Např. při rozvozu pracovníků na roztroušená pracoviště můžeme požadovat, aby se dopravní prostředek nevracel prázdný, ale aby po skončení prací opět naložil v obráceném pořadí všechny pracovníky. Výchozí vrchol je přitom pevně určen, zatímco koncový obvykle nikoli.

Mnohé praktické úlohy jsou obecnější a zahrnují problém obchodního cestujícího jako částečný případ: jestliže se při rozvozu zboží nevejde všechno do jednoho auta,

je třeba vykonat několik jízd (lhostejno, zda postupně nebo několika auty najednou). Optimální rozvoz by pak měl mít tu vlastnost, že každá jízda je optimálním řešením problému obchodního cestujícího pro podmnožinu těch míst, která jsou touto jízdou zásobována.

12.1.7 Příklad: Plánování procesů. Mějme nějaké výrobní zařízení (třeba chemický reaktor), kterým máme periodicky vyrábět n produktů p_1, p_2, \dots, p_n . Má-li po výrobě produktu p_i bezprostředně následovat výroba produktu p_j , pak v závislosti na těchto dvou produktech buď je, anebo není třeba zařízení vyčistit, nastavit nebo seřadit, což stojí nějaký čas nebo peníze. Úkolem je najít cyklický plán výroby všech produktů (tj. cyklické pořadí produktů) bez ztrátových časů.

Tuto úlohu lze snadno převést na hledání hamiltonovského cyklu v grafu, jehož n vrcholů odpovídá produktům p_1, p_2, \dots, p_n a jehož hrany vyjadřují možnost změn vyráběných produktů bez časových nebo peněžních ztrát.

Jestliže v této úloze hamiltonovský cyklus neexistuje, cyklická výroba bez ztrátových časů není možná. V takovém případě se můžeme snažit alespoň o minimalizaci ztrát, což vede na orientovanou variantu problému obchodního cestujícího.

Má-li být výroba jednorázová (každý produkt se má vyrábět jen jedenkrát), pak odpadnou náklady na uvedení výrobního zařízení do výchozího stavu, což vede k hledání nejkratší orientované hamiltonovské cesty.

12.2 Vzájemné převody úloh

Situace, kdy máme řešit nějakou úlohu, ale nemáme k dispozici vhodnou metodu řešení (nebo počítačový program), je bohužel častá. Mnohdy si však lze pomoci tím, že danou úlohu převedeme (transformujeme) na úlohu jinou, kterou řešit dovedeme, a z nalezeného řešení této druhé úlohy pak odvodíme řešení úlohy původní. Dovednost převádět jednu úlohu na druhou je obecně velice užitečná. V případě hamiltonovských úloh, které mají značný počet variant, je toto „umění“ tím užitečnější.

Souhrnně lze konstatovat, že všechny existenční hamiltonovské úlohy lze vzájemně převádět (každou na každou) a totéž platí i o úlohách optimalizačních. Navíc lze každou existenční úlohu převést na analogickou úlohu optimalizační. Všechny tyto převody jsou P-redukce, viz 11.1.7, str. 188.

Uvedeme zde jen některé převody, ostatní přenecháme čtenáři jako cvičení.

12.2.1 Převod existenční úlohy na optimalizační je snadný: Daný graf (prostý a bez smyček) doplníme na úplný graf. Délky původních hran zvolíme nulové, přidané hrany budou mít délku 1. Nyní platí, že v původním grafu existuje hamiltonovská cesta (kružnice, cyklus) právě tehdy, když v odpovídajícím úplném grafu existuje hamiltonovská cesta (kružnice, cyklus) o nulové délce. Žádná cesta (kružnice, cyklus) nemůže mít délku zápornou, proto nejkratší cesta, má-li nulovou délku, je řešením původní existenční úlohy. Pokud nejkratší cesta v úplném grafu má délku kladnou, pak v původním grafu cesta s požadovanými vlastnostmi neexistuje: kdyby existovala, musela by mít nulovou délku.

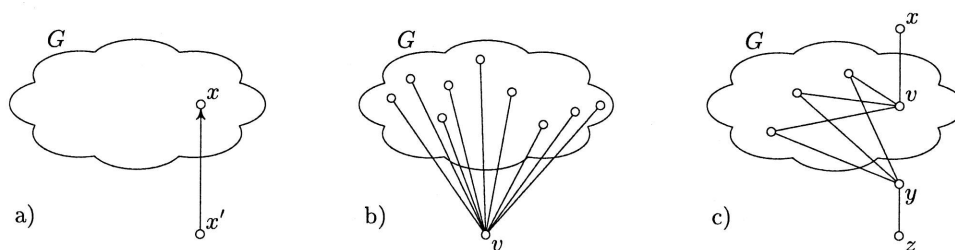
12.2.2 Převod optimalizační úlohy na úplný graf. Podobný trik se používá k převodu optimalizační úlohy v obecném grafu na analogickou optimalizační úlohu v grafu úplném. Původní graf doplníme na úplný, přičemž jako délku nově přidaných hran zvolíme nějaké dost velké kladné číslo M . Jako M lze použít např. součet absolutních hodnot délek všech hran. Potom najdeme optimální řešení v úplném grafu. Pokud toto řešení neobsahuje žádnou z přidaných hran, je toto řešení i optimálním řešením úlohy pro původní graf. Pokud naopak nějakou přidanou hranu obsahuje, pak v původním grafu přípustné řešení neexistuje.

12.2.3 Fixovaný počáteční vrchol. Hledáme-li hamiltonovskou cestu, jejíž počáteční vrchol x v grafu G je pevně dán, můžeme takovou úlohu převést na obdobnou úlohu ve větším grafu G' , kde již počáteční vrchol není předepsán. Trik spočívá v tom, že ke grafu G přidáme kopii x' vrcholu x a hranu z vrcholu x' do x (viz obr. 12.2a). Poněvadž z vrcholu x' vychází v G' jen jediná hrana, nemůže žádná hamiltonovská cesta v grafu G' začínat jinde než v tomto přidaném vrcholu a odtud musí pokračovat přidanou hranou do vrcholu x . Všechny hamiltonovské cesty v G začínající v x a všechny hamiltonovské cesty v G' si vzájemně jednoznačně odpovídají, liší se pouze v hraně (x, x') . Postupujeme tedy tak, že v G' najdeme hamiltonovskou cestu (např. nejkratší), pak vynecháním přidané hrany dostaneme (stejně dlouhou) hamiltonovskou cestu v G , která začíná tam, kde to bylo předepsáno, tj. v x .

Stejný trik lze použít, je-li předepsán koncový vrchol hamiltonovské cesty.

Pro optimalizační úlohu by bylo možno postupovat stejně (délka přidané hrany by byla nulová), existuje však lepší možnost. Popíšeme ji pro orientovanou verzi úlohy: graf ponecháme beze změny, ale všem hranám z množiny $E^+(x)$ zvětšíme jejich délku o dostatečně velké číslo M a hranám z množiny $E^-(x)$ zvětšíme jejich délku o hodnotu $2M$. Tím všechny orientované hamiltonovské cesty, které začínají ve vrcholu x , prodloužíme o hodnotu M , zatímco všechny ostatní hamiltonovské cesty prodloužíme o $3M$ nebo $2M$ podle toho, zda vrcholem x prochází nebo v něm končí. Cesty, které nezačínají v x , jsou tím natolik znevýhodněny, že nejkratší hamiltonovská cesta při změněných délkách bude začínat v x (pokud taková existuje).

12.2.4 Převod neorientované cesty na kružnici. Úlohu najít neorientovanou hamiltonovskou cestu s libovolnými krajními vrcholy v grafu G lze převést na úlohu najít hamiltonovskou kružnici ve větším grafu G' , který získáme tak, že ke grafu



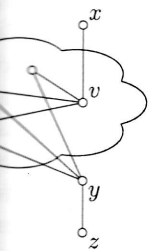
Obrázek 12.2: Ilustrace převodů hamiltonovských úloh.

nik se používá
 alizační úlohu
 vě přidaných
 např. součet
 ení v úplném
 toto řešení
 ou přidanou

estu, jejíž po-
 st na obdob-
 Trik spočívá
 do x (viz obr.
 žádná hamil-
 a odtud musí
 sty v G začí-
 ě odpovídají,
 hamiltonovskou
 jně dlouhou)
 tj. v x .
 é cesty.

řidané hrany
 ovanou verzi
 (x) zvětšíme
 ětšíme jejich
 které začínají
 hamiltonovské
 rochází nebo
 hodněny, že
 v x (pokud

orientovanou
 vést na úlohu
 že ke grafu

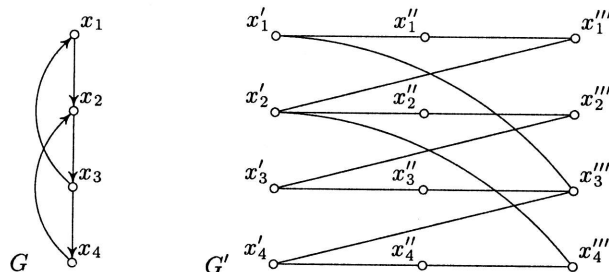


G přidáme jeden nový vrchol v a spojíme jej hranami se všemi vrcholy grafu G . V optimalizační úloze zvolíme délky nových hran nulové. V grafu G' existuje hamiltonovská kružnice právě tehdy, když v G existuje hamiltonovská cesta. Viz obr. 12.2b.

12.2.5 Převod hamiltonovské kružnice na neorientovanou cestu. Úlohu najít hamiltonovskou kružnici v grafu G lze naopak převést na hledání hamiltonovské cesty s libovolnými krajními vrcholy ve větším grafu G' , který získáme takto: Ke grafu G přidáme tři nové vrcholy x, y, z a v grafu G zvolíme libovolně vrchol v . Ke každé hraně mezi vrcholem v a jeho sousedem a v grafu G přidáme v grafu G' stejně dlouhou hranu mezi a a y . Navíc spojíme vrcholy y a z a vrcholy v a x hranami o nulové délce. V takto sestrojeném grafu G' existuje hamiltonovská cesta (z x do z) právě tehdy, když v G existuje (stejně dlouhá) hamiltonovská kružnice. Viz obr. 12.2c.

12.2.6 Převod orientované verze na neorientovanou. Jako poslední ukážeme převod orientovaných verzí hamiltonovských úloh na verze neorientované. Každému vrcholu x_i původního orientovaného grafu přiřadíme trojici vrcholů x'_i, x''_i, x'''_i . Každý vrchol x''_i spojíme neorientovanými hranami o nulové délce s vrcholy x'_i a x'''_i . Každé orientované hraně z x_i do x_j v grafu G přiřadíme v grafu G' stejně dlouhou neorientovanou hranu spojující vrcholy x'''_i a x'_j . Viz obr. 12.3.

Vtip této konstrukce je v tom, že všechny dvoučárkované vrcholy musí hamiltonovská cesta procházet stejným směrem, tj. buď všechny od jednočárkované ke tříčárkované, nebo všechny naopak. Orientované hamiltonovské cesty a cykly v původním grafu G tedy vzájemně jednoznačně odpovídají stejně dlouhým neorientovaným hamiltonovským cestám a kružnicím v grafu G' .



Obrázek 12.3: Převod orientované verze hamiltonovských úloh na neorientovanou.

12.2.7 Cvičení. Promyslete detaily převodu úlohy najít nejkratší neorientovanou cestu s fixovaným počátečním i koncovým vrcholem na úlohu bez fixovaných vrcholů. Použijte analogii k 12.2.3.

12.2.8 Cvičení. Jak velké musí být číslo M , které je použito v převodech 12.2.2 a 12.2.3?

12.2.9 Cvičení. Najděte příklad ohodnoceného grafu, v němž nejkratší neorientovaná hamiltonovská cesta není částí nejkratší hamiltonovské kružnice.

Jinak řečeno, najděte příklad, který ukazuje, že nejkratší hamiltonovskou cestu nelze hledat tak, že vezmeme řešení problému obchodního cestujícího a vynecháme z něj nejdelší hranu.

12.2.10 Správnost převodu je třeba dokázat. Předchozí cvičení ukazuje, že ne každý nadějně vypadající „převod“ jedné úlohy na druhou je korektní a skutečně použitelný. Chcete-li nějaký převod použít bez rizika chyby, měli byste o tomto převodu *dokázat* dvě věci:

- pokud původní úloha má řešení, pak pomocí převodu takové řešení najdeme,
- pokud původní úloha řešení nemá, pak to pomocí převodu spolehlivě zjistíme.

12.3 Existenční hamiltonovské úlohy

Existenční hamiltonovské úlohy obecně patří k NP-těžkým úlohám. V některých speciálních případech však přesto lze rychle prokázat, že řešení existuje nebo naopak neexistuje.

12.3.1 Cvičení. Graf, který obsahuje most (viz 4.4.1, str. 65), nemůže obsahovat hamiltonovskou kružnici. Najděte graf, který žádný most neobsahuje a přesto v něm hamiltonovská kružnice také neexistuje.

12.3.2 Věta. (Chvátal 1972.) Nechť G je prostý graf bez smyček s $n \geq 3$ vrcholy. Jestliže $d_1 \leq d_2 \leq \dots \leq d_n$ je soubor stupňů jeho vrcholů a jestliže

$$(12.1) \quad d_k \leq k \Rightarrow d_{n-k} \geq n - k \quad \text{pro } 1 \leq k < n/2,$$

pak graf G obsahuje hamiltonovskou kružnici.

Jestliže naopak posloupnost čísel $d_1 \leq d_2 \leq \dots \leq d_n$ nesplňuje podmínku 12.1, pak existuje graf o n vrcholech, který neobsahuje hamiltonovskou kružnici a pro jehož všechny vrcholy v_i platí $d(v_i) \geq d_i$.

POZNÁMKA: Prvá část věty zaručuje existenci hamiltonovské kružnice, jsou-li stupně vrcholů „dost velké“. Druhá část vlastně říká, že lepší podmínka založená pouze na stupních vrcholů již neexistuje. Dodejme, že pokud není splněna podmínka 12.1 (a takových grafů je spousta), hamiltonovská kružnice může a nemusí existovat, věta o tom nic neříká.

Důkaz této věty není jednoduchý, lze jej najít např. v knize [Nešetřil79]. Poznamenejme pouze, že důkaz je proveden sporem a nedává žádný návod k hledání hamiltonovské kružnice, a to ani v případě, když věta její existenci zaručuje.

12.3.3 Věta. (Ghouilla-Houri 1960.) Nechť G je silně souvislý prostý orientovaný graf s n vrcholy a nechť pro každý vrchol x platí $d^+(x) + d^-(x) \geq n$. Potom graf G obsahuje hamiltonovský cyklus.

POZNÁMKA: Také tato věta nedává návod, jak hamiltonovský cyklus hledat, a to ani v případě, kdy jeho existenci zaručuje.

12.3.4 Cvičení. Zjistěte, zda existuje hamiltonovská kružnice v úplném bipartitním grafu $K_{6,8}$?

12.3.5 Hledání hamiltonovských cest, cyklů a kružnic. Můžeme použít backtracking (viz 11.2, str. 189), který v tomto případě bude modifikací algoritmu 3.3.10 k prozkoumání všech cest, které začínají ve zvoleném vrcholu. Algoritmus 3.3.10 lze snadno doplnit o test, zda právě nalezená cesta je hamiltonovskou cestou, kružnicí, popř. cyklem. Takto můžeme získat dokonce úplný seznam všech hamiltonovských cest, kružnic či cyklů v daném grafu, je to však časově velice náročné i pro nepřilíš velké grafy.

Celý postup lze poněkud zrychlit tím, že se v každém kroku snažíme rozpoznat hrany, které musíme nebo naopak nesmíme přidávat ke stávající cestě, aby ji bylo možno doplnit na hamiltonovskou. Dosažitelné zrychlení však je řádově jen asi dvojnásobné. Detaily lze najít v [Chr75].

Další možností je převod na optimalizační úlohu, viz 12.2.1.

12.3.6 Heuristické postupy. Pro řešení hamiltonovských úloh byla vypracována také řada poměrně rychlých algoritmů, u nichž však není zaručeno, že řešení najdou vždy, když existuje. V jednom z nich např. prodlužujeme stávající cestu, dokud je to možné. Nelze-li již cestu prodloužit, pokoušíme se závěrečnou část cesty odpojit a připojit obráceně (viz obr. 12.4) a pokračovat v prodlužování z jiného vrcholu.



Obrázek 12.4: Přepojování cest.

Pro zvýšení naděje na úspěch se doporučuje tyto algoritmy použít opakovaně na též graf s náhodně přechíslovanými vrcholy. Podrobnosti lze najít v [Chr75] a [Kučera83].

12.4 Optimalizační hamiltonovské úlohy

Optimalizační hamiltonovské úlohy jsou přinejmenším stejně obtížné jako úlohy existenční, neboť každou existenční úlohu lze velmi jednoduše převést na řešení úlohy optimalizační (viz 12.2.1). Všechny tyto úlohy jsou pro obecné grafy NP-těžké.

Problém obchodního cestujícího lze asi nejjednodušeji řešit takzvanou metodou hrubé síly, tedy jednoduchým, ale pracným vyzkoušením všech přípustných řešení, tj. všech permutací (pořadí) vrcholů: pro každou permutaci vypoč-

teme délku odpovídající hamiltonovské kružnice a vybereme z nich nejkratší. Počet všech přípustných řešení však roste jako faktoriál počtu vrcholů, proto přidání i jen jediného vrcholu prodlouží výpočet mnohonásobně. Tento primitivní způsob řešení je prakticky použitelný jen na grafy s nejvýše asi tak tuctem vrcholů.

Pro obecnější hamiltonovské optimalizační úlohy lze podobně použít jednoduchý backtracking (např. podle 3.3.10, str. 56) k nalezení všech hamiltonovských cest, kružnic nebo cyklů a z nich pak opět vybrat nejkratší. Pokud však při tom nepoužijeme nějaké triky pro urychlení výpočtu, je to jen jiná forma metody hrubé síly. V grafech, které mají hodně hran (zejména pak v úplných grafech), to není o mnoho rychlejší než výše zmíněné vyzkoušení všech permutací.

Dále uvedeme několik algoritmů, které lze použít na úlohy o něco rozsáhlejší (několik desítek vrcholů).

12.4.1 Hledání nejkratší neorientované hamiltonovské cesty. Popíšeme algoritmus, který je jednoduchým použitím metody větví a mezí (viz 11.3, str. 191). Úlohu přeformulujeme takto:

Hledáme nejlevnější podgraf H daného grafu G takový, že:

1. podgraf H je kostrou grafu G ,
2. všechny vrcholy podgrafu H mají stupeň $d_H(x) \leq 2$.

Prvá z uvedených omezujících podmínek je příjemná ve smyslu 11.3.4, relaxováním (uvolněním) druhé (nepříjemné) podmínky totiž dostaneme známou úlohu o minimální kostře, kterou lze snadno řešit, viz 4.3, str. 61.

Pokud v nejlevnější kostře, kterou dostaneme řešením relaxované úlohy, mají všechny vrcholy stupeň nejvýše 2, pak tato kostra je nejlevnější nejen mezi kostrami, ale i mezi všemi hamiltonovskými cestami, máme tedy optimální řešení.

Pokud má některý vrchol x v kostře H stupeň $d_H(x) > 2$, pak úlohu rozvětvíme na podúlohy. Nejjednodušší verze větvení je založena na faktu, že hrany z množiny $E_H(x)$ nemohou být všechny obsaženy v téže hamiltonovské cestě. Proto pro každou hranu $e \in E_H(x)$ vytvoříme podúlohu, v níž je hrana e zakázána. Tím dostaneme $d_H(x)$ podúloh, které pak řešíme stejným postupem jako původní úlohu: hledáme nejlevnější kostru a testujeme stupně vrcholů v této kostře. Je-li třeba, podúlohu dále větvíme zakazováním dalších hran.

Dodejme, že po zákazu hrany není nutno opakovat celý výpočet nejlevnější kostry. Zákazem hrany se dosavadní minimální kostra rozpadla na dvě komponenty a tyto komponenty stačí spojit nejlevnější dosud nezakázanou hranou. Vyplývá to z věty 4.3.14, str. 64.

Právě popsáný postup řešení je předveden v příkladě 12.4.2.

Dodejme dále, že výše uvedený způsob větvení na podúlohy je sice jednoduchý, ale neefektivní, poněvadž vytvořené podúlohy nemají disjunktní množiny přípustných řešení. Snadno se pak stane, že totéž řešení se zbytečně vyskytuje v několika větvích stromu řešení, strom se příliš rozrůstá a výpočet je zbytečně pomalý. Platí to zejména v případě, kdy stupeň vrcholu v kostře je větší než tři.

Efektivnější způsob větvení využívá nejen zákazy, ale i vynucenou přítomnost hrany v kostře:

- zakážeme hranu $e_1 \in E_H(x)$,
- vynutíme hranu e_1 a zakážeme $e_2 \in E_H(x)$,
- vynutíme hrany e_1, e_2 a zakážeme všechny ostatní hrany z množiny $E_G(x)$ (pozor, nikoli pouze z $E_H(x)$).

Všimněte si, že u třetí z těchto podúloh využíváme fakt, že nejvýše dvě hrany z množiny $E_G(x)$ mohou být obsaženy v hamiltonovské cestě. Dále si všimněte, že tímto způsobem dostaneme vždy jen tři podúlohy nezávisle na stupni vrcholu x v kostře.

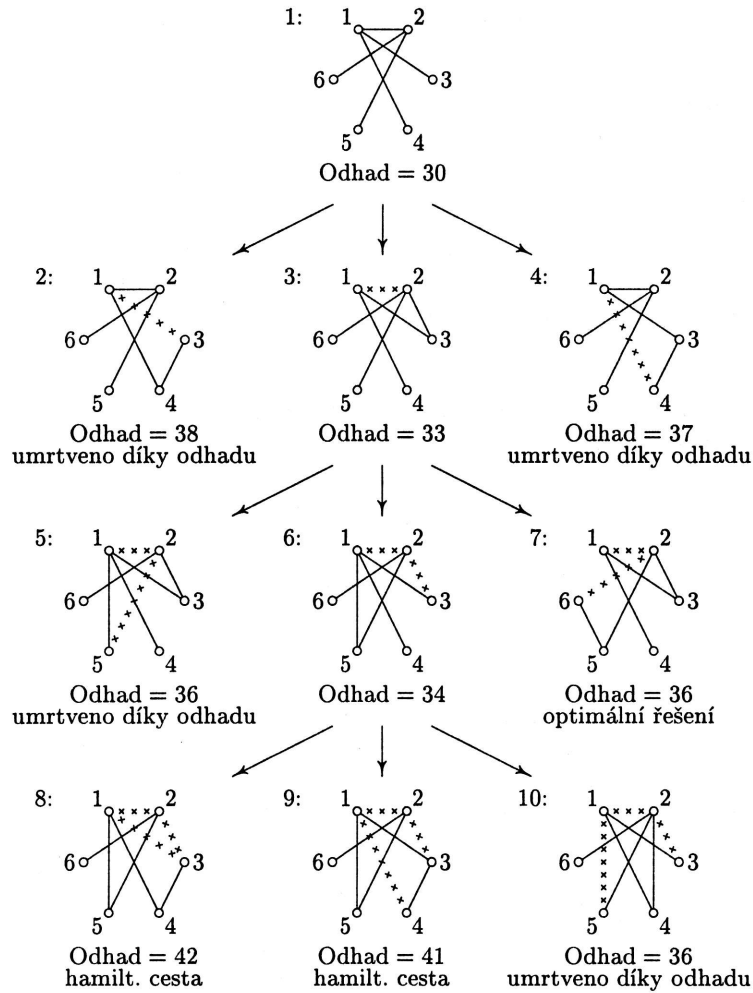
12.4.2 Příklad. Hledejme nejkratší hamiltonovskou cestu v úplném grafu, ve kterém jsou délky hran dány maticí

$$\begin{pmatrix} \infty & 7 & 2 & 3 & 11 & 17 \\ 7 & \infty & 10 & 13 & 8 & 10 \\ 2 & 10 & \infty & 10 & 19 & 15 \\ 3 & 13 & 10 & \infty & 16 & 15 \\ 11 & 8 & 19 & 16 & \infty & 13 \\ 17 & 10 & 15 & 15 & 13 & \infty \end{pmatrix}.$$

K řešení použijeme právě popsany algoritmus 12.4.1. K větvení budeme vybírat vždy živou podúlohu s nejnadhjnějším (tj. zde nejmenším) odhadem, větvení budeme dělat prvním a jednodušším z obou popsanych způsobů (tj. pouze zakazujeme jednotlivé hrany). Průběh výpočtu je patrný z následující tabulky a z obr. 12.5 na následující straně.

podúloha	zakázané hrany	odhad	komentář
1	—	30	větveno na podúlohy 2, 3, 4
2	(1, 3)	38	umrtveno díky odhadu
3	(1, 2)	33	větveno na podúlohy 5, 6, 7
4	(1, 4)	37	umrtveno díky odhadu
5	(1, 2), (2, 5)	36	umrtveno díky odhadu
6	(1, 2), (2, 3)	34	větveno na podúlohy 8, 9, 10
7	(1, 2), (2, 6)	36	optimální řešení
8	(1, 2), (2, 3), (1, 3)	42	umrtveno díky odhadu
9	(1, 2), (2, 3), (1, 4)	41	umrtveno díky odhadu
10	(1, 2), (2, 3), (1, 5)	36	umrtveno díky odhadu

12.4.3 Cvičení. Řešte znovu příklad 12.4.2, ale použijte efektivnější způsob větvení, popsany v 12.4.1. Výsledek bude týž, ale dosáhnete jej s menším stromem řešení. (Obecně by výsledkem nemusela být stejná hamiltonovská cesta, ale délka výsledné cesty samozřejmě stejná být musí.)



Obrázek 12.5: Strom řešení k příkladu 12.4.2.

12.4.4 Nejkratší hamiltonovský cyklus lze hledat rovněž metodou větvi a mezí. Úlohu přeformulujeme takto:

Hledáme nejlevnější podgraf H daného grafu G takový, že:

1. hrany podgrafu H tvoří soustavu vrcholově disjunktních cyklů, které procházejí všemi vrcholy grafu G ,
2. podgraf H je souvislý.

Prvou z podmínek opět pokládáme ve smyslu 11.3.4 za příjemnou, druhou za nepříjemnou. Relaxovanou úlohu, v níž je uvolněna (relaxována) podmínka 2, lze řešit převodem na přiřazovací úlohu 9.4, str. 171:

Každému vrcholu původního grafu je třeba přiřadit vrchol, který po něm v cyklu následuje, každý vrchol zároveň musí být přiřazen jako následník přesně jednomu vrcholu, totiž svému předchůdci. Toto přiřazení můžeme chápat jako perfektní párování v bipartitním grafu, který dostaneme tak, že každý vrchol původního grafu zdvojíme.

Jinou možností, jak hledat zmíněnou soustavu cyklů, je pomocí toků v síti. Hledáme nejlevnější cirkulaci takovou, že tok každým vrcholem má jednotkovou velikost. Tuto úlohu nejprve trikem podle 8.1.13, str. 132, převedeme na standardní úlohu hledání nejlevnější přípustné cirkulace a tu pak řešíme algoritmem 8.7.

Pokud nejkratší soustava cyklů nespĺňuje podmínku 2, tj. pokud jde o soustavu více než jednoho cyklu, je zřejmé, že hrany některého (libovolného) z těchto cyklů nemohou být všechny obsaženy v cyklu hamiltonovském. Proto můžeme úlohu rozvést na podúlohy nejjednodušeji tak, že vybereme jeden z cyklů (doporučuje se, aby měl co nejméně hran) a pro každou jeho hranu e_i vytvoříme podúlohu U_i , v níž je hrana e_i zakázána.

V přiřazovací úloze lze zákaz hrany e_i docílit zvýšením její ceny na dostatečně vysokou hodnotu (tzv. prohibivní cena). Přitom není nutno opakovat celý výpočet přiřazovací úlohy, stačí změnit cenu zakazované hrany a pokračovat od poslední fáze výpočtu. Při použití toku v síti docílíme zákazu změnou kapacity hrany e_i . Viz též poznámku 8.7.9 k algoritmu Out of Kilter.

Výše popsany jednoduchý způsob větvení na podúlohy je neefektivní, neboť podúlohy nemají disjunktní množiny přípustných řešení. Disjunktnosti lze dosáhnout vynucováním hran podobně jako v 12.4.1.

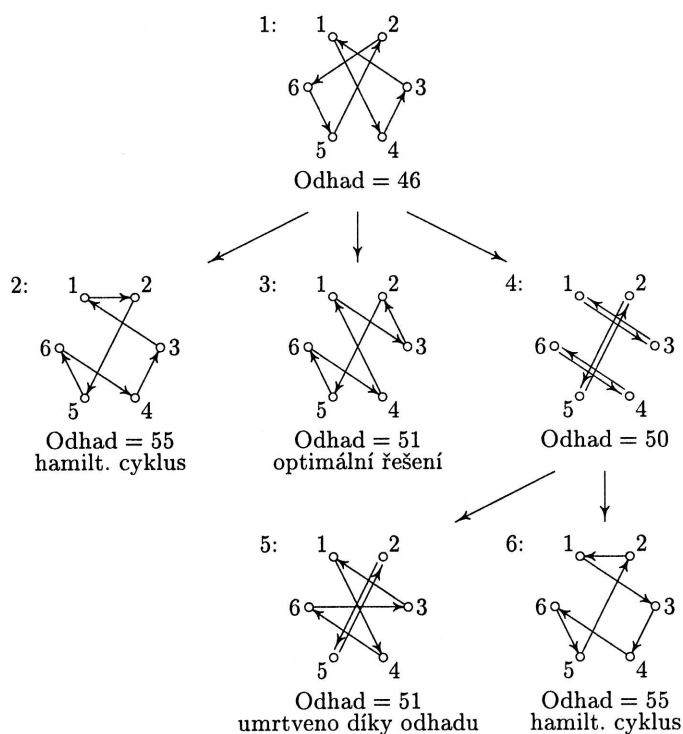
Lepší výsledky (menší strom řešení) však dává následující způsob větvení: Zvolme některý cyklus a označme M množinu všech jeho vrcholů. Ve výsledném hamiltonovském cyklu musí zcela jistě být zahrnuta alespoň jedna hrana z množiny $W^+(M)$, jinými slovy, je třeba „donutit cyklus, aby opustil množinu M “. Toho lze docílit tím, že vytvoříme podúlohu pro každý vrchol $v_i \in M$, a to tak, že zakážeme všechny hrany, které z vrcholu v_i vedou do množiny M .

Uvedeným postupem lze řešit orientovanou verzi problému obchodního cestujícího. Lze jej samozřejmě použít i pro řešení verze neorientované (kde matice délek hran je symetrická).

12.4.5 Příklad. Použijeme právě popsanou metodu k nalezení nejkratšího hamiltonovského cyklu v úplném grafu s maticí délek hran z příkladu 12.4.2.

Průběh řešení je patrný z následující tabulky a obrázku 12.6 na následující straně. K větvení byla použita poslední z uvedených metod.

podúloha	zakázané hrany	odhad	komentář
1	–	46	větveno na podúlohy 2, 3, 4
2	(1, 3), (1, 4)	55	hamilt. cyklus
3	(3, 1), (3, 4)	51	optimální řešení
4	(4, 1), (4, 3)	50	větveno na podúlohy 5, 6
5	(4, 1), (4, 3), (1, 3)	51	umrtveno díky odhadu
6	(4, 1), (4, 3), (3, 1)	55	hamilt. cyklus



Obrázek 12.6: Strom řešení k příkladu 12.4.5.

12.4.6 1-strom a metoda penalizace. Jde o docela poučný trik, který lze použít k hledání nejkratší hamiltonovské kružnice, a tedy i k řešení problému obchodního cestujícího. Mějme graf s množinou vrcholů $V(G) = \{1, \dots, n\}$.

1-strom je faktor H daného grafu G , který má stejný počet vrcholů a hran a vrchol 1 v něm má stupeň $d_H(1) = 2$. Jinak řečeno, libovolný 1-strom daného grafu o n vrcholech dostaneme tak, že vezmeme podgraf indukovaný množinou $\{2, \dots, n\}$, najdeme nějakou kostru tohoto podgrafu a přidáme k ní vrchol 1 spolu s nějakými dvěma hranami z tohoto vrcholu. Každý 1-strom tedy obsahuje přesně jednu kružnici a tato kružnice prochází vrcholem 1.

Nyní můžeme úlohu najít nejkratší hamiltonovskou kružnici přeformulovat takto: hledáme podgraf H daného grafu G takový, že

1. podgraf H je 1-stromem,
2. každý vrchol x podgrafu H má stupeň $d_H(x) = 2$.

Opět platí, že prvá z obou podmínek je příjemná, protože nejlevnější 1-strom lze snadno najít pomocí algoritmu pro nejlevnější kostru (viz 4.3): najdeme nejlevnější kostru na množině $\{2, \dots, n\}$ a přidáme dvě nejlevnější hrany z množiny $E_G(1)$.

Pokud nejlevnější 1-strom náhodou splňuje i druhou podmínku, pak tento 1-strom je zároveň hledanou nejkratší hamiltonovskou kružnicí.

Druhá z podmínek je však nepříjemná a činí úlohu obtížnou. Platnost této podmínky samozřejmě lze vynucovat metodou větví a mezi podobně jako při hledání nejkratší hamiltonovské cesty v 12.4.1. Na tuto metodu zpravidla stejně dojde, ale napřed je vhodné na podúlohu zkusit trik, kterému se říká *penalizace vrcholů*.

Každému vrcholu x grafu G přiřadíme číslo $p(x)$, tzv. *penále*. Každé hraně grafu, která spojuje vrcholy i a j , nyní změňme délku z hodnoty $a(i, j)$ na hodnotu $a(i, j) + p(i) + p(j)$ a v takto ohodnoceném grafu znovu vyhledáme nejlevnější 1-strom.

Vtip penalizace spočívá v tom, že délky všech hamiltonovských kružnic se vlivem penále zvětšily o stejnou hodnotu, totiž o dvojnásobek sumy všech penále, tj. o hodnotu $2 \sum_{i=1}^n p(i)$. Délky ostatních 1-stromů se přitom mohly změnit jinými způsoby, některé mohly stoupnout, jiné klesnout. Proto, podaří-li se vhodně zvoleným penále docílit toho, že nejlevnější 1-strom bude zároveň hamiltonovskou kružnicí, budeme mít jistotu, že tento 1-strom (tj. kružnice) je také nejkratší ze všech hamiltonovských kružnic.

Penále, popř. jeho změna, se zpravidla odvozuje ze stupňů vrcholů v 1-stromu. Má-li vrchol vysoký stupeň, dostane kladné penále, tím se prodraží hrany z jeho okolí a je naděje, že se stupeň sníží. Má-li naopak vrchol stupeň 1, dostane záporné penále, ceny hran v jeho okolí klesnou a stupeň vrcholu se může zvětšit. Nejjednodušší pravidlo radí zvětšovat dosavadní penále vrcholu v o hodnotu $k(d(v) - 2)$, kde konstanta k nemá být příliš velká, může být i rovna jedné. Různé způsoby volby penále a výpočetní zkušenosti jsou podrobně popsány v knize [Chr75].

Výpočet obvykle začíná s nulovým penále, tj. s původními délkami hran tím, že vypočteme nejlevnější 1-strom. Pokud není nalezena hamiltonovská kružnice, tj. pokud některý vrchol má jiný stupeň než 2, vypočteme změny (přírůstky) penále pro jednotlivé vrcholy, změňme délky hran a znovu hledáme nejlevnější 1-strom.

Metoda penalizace bohužel nevede vždy k cíli. Existují grafy, kde sebedůmyslnější volba penále nevynutí vznik hamiltonovské kružnice. Výše popsaný výpočet by pak nikdy neskončil. Proto je nutno neúspěšný výpočet ukončit buď když zjistíme, že se opakuje stejná situace, nebo prostě po přiměřeném počtu neúspěšných penalizací.

Ovšem i neúspěšná penalizace není zbytečná, lze ji použít v metodě větví a mezi k získání lepšího dolního odhadu. Jestliže jsme při použití penále p dostali 1-strom o ceně C , pak číslo $C - 2 \sum_{i=1}^n p(i)$ lze použít jako dolní odhad: délka žádné hamiltonovské kružnice (měřená v původních, tj. nepenalizovaných cenách) nemůže být kratší než toto číslo. V průběhu neúspěšné penalizace takovýchto odhadů získáme několik a jako odhad pro celou podúlohu použijeme nejlepší (tj. zde největší) z nich.

12.4.7 Cvičení. Metodu penalizace lze snadno upravit pro hledání nejkratší neorientované hamiltonovské cesty z daného počátečního vrcholu x do daného koncového vrcholu y . V takovém případě namísto nejlevnějšího 1-stromu hledáme minimální kostru takovou, že vrcholy x a y v ní mají stupeň 1. Takovou minimální kostru lze snadno najít buď tak, že najdeme minimální kostru na množině vrcholů $V(G) \setminus \{x, y\}$ a přidáme nejlevnější hrany z vrcholů x a y , nebo (jednodušeji) vrcholům x a y dáme velmi vysoké počáteční penále.

Jak se změní délky všech hamiltonovských cest z x do y při použití penále p ? Jakou hodnotu lze použít jako dolní odhad v metodě větví a mezí?

12.4.8 Heuristické algoritmy pro problém obchodního cestujícího. Poměrně dobré výsledky dává *metoda nejvzdálenějšího ukládání*:

Začneme s triviální kružnicí v nějakém vrcholu a do této kružnice postupně zařazujeme další vrcholy, dokud nedostaneme kružnici hamiltonovskou. K zařazení do kružnice vybíráme vrchol, který je od kružnice nejvíce vzdálen (odtud název metody). Přesněji, máme-li kružnici K , která prochází vrcholy v_1, \dots, v_k , pak vybereme vrchol v , pro který je výraz $\min\{a(v, v_i) \mid i = 1, \dots, k\}$ největší. Vrchol v pak zařadíme do kružnice mezi takové dva sousední vrcholy, aby se tím kružnice prodloužila co nejméně.

Stojí za zmínku, že o něco „hladovější“ postup, při němž se vkládá vrchol, který je ke kružnici nejbližší, dává v průměru o něco horší výsledky. Je vidět, že chamtivost se vždycky nevyplácí.

Dobré výsledky dává také lokální průzkum (viz 11.4, str. 196), v němž se stávající řešení snažíme vylepšovat prováděním „drobných změn“. Osvědčilo se kombinovat dva typy změn. Jedna spočívá v tom, že ze stávající hamiltonovské kružnice odstraníme dvě hrany a zbývající části kružnice propojíme „do kříže“ tak, aby znovu vznikla hamiltonovská kružnice. Druhý typ spočívá v odstranění vrcholu a jeho zařazení na jiné místo kružnice.

K

B

Pojm
mají
barv
jsou
V
nosti
omez

13.

13.1
vrch
dva
může
G
grafu
barev

13.1
že to

tí penále p ?

jícího. Po-

ce postupně

. K zařazení

odtud název

..., v_k , pak

větší. Vrchol

ím kružnice

vrchol, který

e chamtivost

v němž se

Osvědčilo se

miltonovské

o kříže“ tak,

nění vrcholu

Kapitola 13

Barevnost, nezávislost, kliky

Pojmy uvedené v nadpise bývají někdy označovány jako kombinatorické. Třebaže mají důležité aplikace, jsou úlohy s nimi spojené většinou NP-těžké. Algoritmy pro barvení grafu a pro hledání nejpočetnějších nezávislých množin, které zde uvedeme, jsou založeny na backtrackingu.

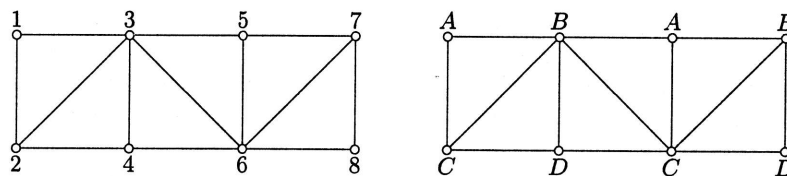
Všechny pojmy studované v této kapitole jsou nezávislé na orientaci a na násobnosti hran. Můžeme se tedy omezit na **prosté neorientované grafy**. Navíc se omezíme na grafy **bez smyček**.

13.1 Definice a základní fakta

13.1.1 Obarvení grafu G (nebo též obarvení vrcholů grafu) je ohodnocení vrcholů hodnotami z množiny B (takzvanými *barvami*), a to takové, že žádné dva sousední vrcholy nejsou ohodnoceny (obarveny) stejnou barvou. V roli barev můžeme brát libovolné prvky z libovolné množiny, často používáme např. čísla.

Graf nazýváme r -barevným, jestliže existuje jeho obarvení r barvami. *Barevnost grafu* (někdy též *chromatické číslo grafu* nebo *vrcholová barevnost*) je nejmenší počet barev, který je potřebný k obarvení grafu. Barevnost grafu G značíme $\chi(G)$.

13.1.2 Cvičení. Najděte obarvení grafu z obrázku 13.1 třemi barvami. Ověřte, že toto obarvení je až na záměnu barev jediné. Lze obarvit graf, který má smyčky?



Obrázek 13.1: Graf a jeho obarvení čtyřmi barvami A, B, C, D .

13.1.3 Nezávislá množina vrcholů v grafu G je taková množina, jejíž žádné dva vrcholy nejsou spojeny hranou. Podgraf indukovaný touto množinou je tedy diskrétní. *Maximální nezávislá množina* je nezávislá množina, která je maximální vzhledem k inkluzi, tj. nezávislá množina, ke které již nelze přidat žádný vrchol, aniž by přestala být nezávislou.

Nejpočetnější nezávislá množina je nezávislá množina, která má největší počet prvků (mezi všemi nezávislými množinami v grafu G). Každá nejpočetnější nezávislá množina je samozřejmě i maximální nezávislou množinou, ale naopak to neplatí. Jak nejpočetnějších, tak i maximálních nezávislých množin může být v témže grafu několik. Počet vrcholů v nejpočetnější nezávislé množině grafu G nazýváme *nezávislostí grafu G* . Nezávislost grafu G značíme $\alpha(G)$. V aplikacích se setkáváme také s úlohou najít nejdrazší nezávislou množinu ve smyslu součtu ohodnocení vrcholů cenami.

Graf na obr. 13.1 vlevo má tyto maximální nezávislé množiny: $\{1, 4, 5, 8\}$, $\{1, 4, 7\}$, $\{1, 6\}$, $\{2, 5, 8\}$, $\{2, 6\}$, $\{2, 7\}$, $\{3, 7\}$ a $\{3, 8\}$. Nejpočetnější nezávislá množina byla uvedena jako první, nezávislost tohoto grafu je rovna 4. Množina $\{1, 4, 5\}$ je příkladem množiny, která je nezávislá, ale není maximální nezávislá.

Všimněte si, že množina vrcholů, které mohou být obarveny stejnou barvou, je vždy nezávislá.

13.1.4 Klika v grafu G je maximální podgraf grafu G , který je úplným (neorientovaným) grafem. Počet vrcholů v největší klice nazýváme *klikovostí grafu*. Klikovost grafu budeme značit $\omega(G)$.

V grafu na obr. 13.1 je každá klika trojúhelníkem, klikovost tohoto grafu je tedy rovna 3, v grafu je celkem šest klik.

Kliky a maximální nezávislé množiny spolu velmi úzce souvisí. Abychom to ukázali, sestrojme k danému grafu G tzv. *doplňkový graf* $-G$ takto: Oba grafy budou mít stejné vrcholy. V doplňkovém grafu $-G$ budou dva vrcholy spojeny hranou právě tehdy, když nejsou spojeny hranou v původním grafu G . Je zřejmé, že doplňkový graf $-(-G)$ k doplňkovému grafu $-G$ je shodný s původním grafem G . Stejně tak je zřejmé, že kliky v grafu G odpovídají vzájemně jednoznačně maximálním nezávislým množinám v doplňkovém grafu $-G$. Platí tedy $\alpha(G) = \omega(-G)$ a $\omega(G) = \alpha(-G)$.

13.1.5 Příklad: Skladování nebezpečných látek. Mějme n nebezpečných látek. Bezpečnostní předpisy zakazují skladovat některé dvojice látek ve stejné místnosti (skříní, polici). Kolik nejméně místností (skříní, polic) je nezbytně třeba k uskladnění všech n látek?

Řešení lze nalézt pomocí barevnosti: Sestrojme graf G , jehož vrcholy odpovídají zmíněným látkám a v němž jsou dva vrcholy spojeny hranou, jestliže odpovídající látky nesmí být skladovány společně. Přípustné umístění látek do místností pak odpovídá obarvení grafu G . Roli barev zde hrají místnosti. Potřebujeme tedy nejméně $\chi(G)$ místností.

Podívejme se nyní na jinou úlohu: Máme za úkol přepravit najednou, např. v jednom autě, co největší počet vzorků těchto látek. Přepravované vzorky musí

tvořit
vzork
P
množ
nezáv
2.4.2.

p_1, p_2
souča
tek je
jedno

Se

hranc

obarv

najed

prove

vždy

proce

pináč

každ

tedy

bude

pro s

NÁVO

13.1.6

stejno

DŮKA

barva

Se

Pak p

x_i oba

vrchol

Po

exist

x tedy

s pouz

tvořit nezávislou množinu v grafu G , můžeme tedy najednou přepravit nejvýše $\alpha(G)$ vzorků.

Pokud by každý ze vzorků měl danou cenu, mohli bychom požadovat přepravu množiny vzorků, která bude mít co největší cenu. To by odpovídalo nejdražší nezávislé množině v grafu G . Srovnejte tuto poslední variantu s problémem batohu 2.4.2, str. 44.

13.1.6 Příklad: Plánování procesů. Máme za úkol uskutečnit n procesů p_1, p_2, \dots, p_n , přičemž některé dvojice procesů nesmějí (nebo nemohou) probíhat současně např. proto, že vyžadují použití stejných strojů. Kolik časových jednotek je třeba k uskutečnění všech procesů, trvá-li každý proces stejně dlouho (např. jednotku času)? Jaký největší počet procesů může probíhat najednou?

Sestrojme graf G , jehož vrcholy odpovídají procesům. Dva vrcholy spojíme hranou právě tehdy, když odpovídající procesy nesmějí probíhat současně. Každé obarvení vrcholů grafu G rozděluje procesy na skupiny, které mohou probíhat najednou (odpovídající vrcholy jsou obarveny stejnou barvou). Nejkratší možné provedení všech procesů tedy trvá $\chi(G)$. Procesy, které mohou probíhat najednou, vždy tvoří nezávislou množinu. Maximální možný počet současně probíhajících procesů tedy je roven $\alpha(G)$.

13.1.7 Cvičení. Na škole se vyučují předměty p_1, p_2, \dots, p_n ve studijních skupinách s_1, s_2, \dots, s_r . Na škole učí učitelé u_1, u_2, \dots, u_m . Předpokládejme, že pro každou studijní skupinu a předmět je předem pevně určen učitel. Učební plán je tedy dán jako seznam trojic (s_i, p_j, u_k) , které vyjadřují, že ve studijní skupině s_i bude předmět p_j učit učitel u_k . Kolik nejméně vyučovacích hodin bude nezbytných pro splnění učebního plánu?

NÁVOD: Vyučovací hodinu vyjádřenou trojicí (s_i, p_j, u_k) pokládejte za proces (viz 13.1.6). Dva procesy nemohou probíhat současně, vyžadují-li stejného učitele nebo stejnou studijní skupinu.

13.1.8 Věta. V prostém grafu G bez smyček platí:

$$\chi(G) \leq \max\{d(x) \mid x \in V(G)\} + 1.$$

DŮKAZ: Uvedeme jednoduchý algoritmus, který vrcholy grafu G obarví $h + 1$ barvami, kde $h = \max\{d(x) \mid x \in V(G)\}$. V roli barev použijeme čísla $1, 2, \dots, h + 1$.

Seřadme vrcholy do posloupnosti x_1, x_2, \dots, x_n . Vrchol x_1 obarvíme barvou 1. Pak postupně vrcholům x_2, x_3, \dots, x_n přiřadíme barvy podle tohoto pravidla: vrchol x_i obarvíme nejnižší barvou, která se nevyskytuje mezi dosud obarvenými sousedy vrcholu x_i .

Poněvadž každý vrchol x má nejvýše h sousedů, musí mezi $h + 1$ barvami vždy existovat alespoň jedna barva b , kterou není obarven žádný soused vrcholu x . Vrchol x tedy lze obarvit barvou b . Takto lze postupně obarvit všechny vrcholy grafu s použitím nejvýše $h + 1$ barev. \square

POZNÁMKA: Právě uvedený algoritmus je rychlý a je vhodný pro první odhad barevnosti. Algoritmus 13.2.1 pro obarvení grafu nejmenším počtem barev vychází z obarvení získaného touto metodou. Viz také 13.2.7.

13.1.9 Věta. Pro graf G bez smyček o n vrcholech platí:

1. $\chi(G) \leq n$.
2. $\chi(G) \geq \omega(G)$.
3. $\chi(G)\alpha(G) \geq n$.
4. $\chi(G) + \alpha(G) \leq n + 1$.

DŮKAZ: První tvrzení je zřejmé. Druhé tvrzení vyplývá z faktu, že všechny vrcholy kliky je nutno obarvit různými barvami. K důkazu třetího tvrzení si stačí uvědomit, že vrcholy obarvené stejnou barvou tvoří nezávislou množinu. Obarvíme-li tedy graf $\chi(G)$ barvami, získáme $\chi(G)$ nezávislých množin, z nichž každá obsahuje nejvýše $\alpha(G)$ vrcholů. Odtud pak plyne nerovnost $\chi(G)\alpha(G) \geq n$.

K důkazu čtvrtého tvrzení obarvíme nejpočetnější nezávislou množinu první barvou. K obarvení zbývajících $n - \alpha(G)$ vrcholů zcela jistě stačí $n - \alpha(G)$ barev. Platí tedy $n - \alpha(G) + 1 \geq \chi(G)$, což je jen jiný zápis dokazované nerovnosti. \square

13.1.10 Věta. Nechť G je prostý graf bez smyček. Označme n počet vrcholů, m počet hran a h maximální stupeň vrcholu v grafu G . Dále označme $\lfloor x \rfloor$ a $\lceil x \rceil$ dolní a horní celou část čísla x , tj. nejbližší celé číslo takové, že $\lfloor x \rfloor \leq x$ a $\lceil x \rceil \geq x$. Potom o nezávislosti $\alpha(G)$ a barevnosti $\chi(G)$ platí:

1. $\alpha(G) \geq (2n - m)/3$, přičemž rovnost platí pouze pro úplný graf K_3 .
2. $\alpha(G) \geq n^2/(2m + n)$, přičemž rovnost platí pouze tehdy, když souvislé komponenty grafu G jsou úplné grafy téže velikosti.
3. $\alpha(G) \geq \lceil n/(h + 1) \rceil$.
4. $\chi(G) + \chi(-G) \leq n + 1$.
5. $\chi(G) \geq n^2/(n^2 - 2m)$, přičemž rovnost nastává právě tehdy, je-li G úplným $\chi(G)$ -partitním grafem se stejně velkými stranami.
6. Vrcholy grafu G lze rozložit na $h + 1$ disjunktních nezávislých množin, z nichž každá má buď $\lfloor n/(h + 1) \rfloor$, nebo $\lceil n/(h + 1) \rceil$ vrcholů.

DŮKAZ lze najít v knize [Berge73]. \square

13.1.11 Věta. Pro libovolný graf G bez smyček platí:

1. $\chi(G) = 1$ právě tehdy, když graf G je diskrétní.
2. $\chi(G) = 2$ právě tehdy, když graf G neobsahuje kružnici liché délky.
3. $\chi(G) = 2$ právě tehdy, když graf G je bipartitní.

DŮKAZ: První tvrzení je zřejmé. Dokážeme druhé tvrzení. K obarvení kružnice liché délky potřebujeme nejméně 3 barvy. Je-li tedy $\chi(G) = 2$, nemůže graf G takovou kružnici obsahovat. Naopak, nechť graf G neobsahuje kružnici liché délky.

Popíšeme, jak lze graf obarvit dvěma barvami. (Je-li graf nesouvislý, obarvíme takto každou komponentu zvlášť.)

Zvolme pevně vrchol x . Vrcholy, které mají od vrcholu x lichou vzdálenost (ve smyslu počtu hran v nejkratší cestě), obarvíme barvou 1, vrcholy, které mají sudou vzdálenost (včetně vrcholu x) obarvíme barvou 2. Musíme dokázat, že takto získáme skutečně obarvení, tj. že žádné dva stejně obarvené vrcholy nejsou spojeny hranou.

Vezměme tedy dva libovolné vrcholy u, v , které mají oba lichou (oba sudou) vzdálenost od x . Mezi vrcholy u, v vede cesta sudé délky, která je tvořena částmi nejkratších cest z vrcholu u do x a z vrcholu x do v . Kdyby byly vrcholy u, v spojeny hranou, tvořila by tato hrana spolu s cestou z u do v kružnici liché délky. Sestrojili jsme tedy skutečně obarvení dvěma barvami.

Třetí tvrzení je snadné. Každé obarvení dvěma barvami určuje rozklad množiny $V(G)$ na množiny A, B takové, že $W(A) = W(B) = E(G)$, graf je tedy bipartitní. Naopak, je-li graf bipartitní, stačí obarvit každou z jeho stran jednou barvou. \square

13.1.12 Cvičení. Jakou barevnost má strom?

13.1.13 Věta. Rovinný graf bez smyček lze obarvit čtyřmi barvami.

POZNÁMKA: Tato věta je řešením kdysi slavného tzv. *problému čtyř barev*: Stačí čtyři barvy k obarvení libovolné politické mapy tak, aby žádné dva sousední státy nebyly vybarveny stejnou barvou?

O řešení tohoto problému se marně pokoušely řady matematiků i laiků přes 100 let. Od roku 1890 byl znám důkaz, že stačí pět barev. Důkaz pro čtyři barvy však byl nesmírně komplikovaný (pomocí počítače bylo nalezeno a vyšetřeno asi 1900 dílčích případů) a byl nalezen až v roce 1976.

DŮSLEDEK: Rovinný graf G bez smyček o n vrcholech má nezávislost $\alpha(G) \geq n/4$.

13.1.14 Věta. Pro každá dvě celá kladná čísla k a r existuje graf G , který má barevnost $\chi(G) = k$ a neobsahuje kružnici délky menší nebo rovné r .

POZNÁMKA: Tato věta vlastně říká, že existují grafy s vysokou barevností, které jsou „lokálně řídké“. Speciálním případem této věty je tvrzení, že grafy, které neobsahují trojúhelníky, mohou mít libovolně vysokou barevnost. Důkaz tohoto speciálního případu lze najít v knize [Nešetřil79].

13.1.15 Hranová barevnost. V některých aplikacích je vhodnější nebarvit vrcholy, ale hrany. U obarvení hran pak požadujeme, aby každé dvě hrany, které mají společný vrchol, byly obarveny různými barvami. *Hranová barevnost* (též *chromatický index*) je nejmenší počet barev, které stačí k obarvení hran. Hranovou barevnost budeme značit $\chi'(G)$.

Vyšetřování hranové barevnosti grafu G lze převést na vyšetřování (vrcholové) barevnosti v tzv. hranovém grafu: *Hranový graf* ∂G získáme takto: Za vrcholy grafu ∂G prohlásíme hrany původního grafu G . Dva vrcholy v hranovém grafu ∂G jsou spojeny hranou, jestliže jim odpovídající hrany původního grafu G měly společný vrchol.

Obarvení hran grafu G a obarvení vrcholů hranového grafu ∂G si vzájemně jednoznačně odpovídají, platí tedy $\chi'(G) = \chi(\partial G)$. Navíc však platí:

13.1.16 Věta. (Vizing 1964.) Nechť G je prostý neorientovaný graf bez smyček a označme d maximální stupeň vrcholu. Pak hranová barevnost je rovna buď d , nebo $d + 1$.

POZNÁMKA: Navzdory této větě je přesné určení hranové barevnosti (přesněji, rozhodnutí, zda hranová barevnost je rovna d) v obecném případě NP-úplnou úlohou.

13.2 Zjišťování barevnosti

Věta 13.1.11 (a její důkaz) spolu s 3.2.7, str. 52, dává dobrý návod ke snadnému ověření, zda daný graf je jedno- či dvoubarevný. Pro tři a více barev již žádný tak efektivní postup není znám. Naopak, rozhodnutí, zda graf je vrcholově r -barevný pro $r \geq 3$, je NP-úplnou úlohou.

Dále uvedený algoritmus pro barvení grafu nejmenším možným počtem barev je klasickým příkladem backtrackingu (viz 11.2, str. 189).

13.2.1 Algoritmus pro zjišťování barevnosti.

VSTUP: graf G , jehož vrcholy jsou očíslovány čísly $1, 2, \dots, n$. Pro každý vrchol x je dána množina $P(x) = V(x) \cap \{1, 2, \dots, x - 1\}$.

POMOCNÉ PROMĚNNÉ: Právě zkoumané částečné obarvení budeme uchovávat v hodnotách $B(x)$. Hodnoty $BARVA(x)$ budou obsahovat nejlepší dosud nalezené obarvení. Proměnná $OMEZ$ bude obsahovat číslo barvy, kterou již nechceme použít ve snaze získat obarvení méně než $OMEZ$ barvami. S výjimkou počáteční fáze bude proměnná $OMEZ$ rovna počtu barev v nejlepším dosud nalezeném obarvení $BARVA$.

VÝSTUP: výše popsané proměnné $BARVA$ a $OMEZ$.

METODA: Vzhledem k pevnému očíslování vrcholů můžeme každé obarvení pokládat za posloupnost barev. Tyto posloupnosti budeme systematicky prodlužovat a zkracovat v duchu backtrackingu.

Při postupu vpřed, tj. při prodlužování částečné posloupnosti barev, dáme následujícímu vrcholu x jako výchozí vždy nejnižší barvu, která (momentálně) není použita u vrcholů z množiny $P(x)$. Nižší barvu nelze použít (při stávajících barvách předchozích vrcholů), vyšší barvy budou zkoušeny v dalším průběhu backtrackingu (po provedení návratu).

Prvé úplné obarvení tedy bude stejné jako v důkaze věty 13.1.8.

Návraty v backtrackingu budeme provádět jednak při dosažení úplného obarvení, jednak v situaci, kdy některý vrchol y dostal barvu $OMEZ$ nebo vyšší. Přidělením této barvy jsme totiž dostali částečné obarvení, jehož prodlužování již nemá smysl, neboť nemůže vylepšit nejlepší dosud známé obarvení $BARVA$.

Cílem návratu je snížit barvu ve vrcholu y . Poněvadž však nižší barvy v y již jsou vyzkoušeny, je třeba změnit (a tedy zvýšit) barvu v některém předchozím vrcholu

$x < y$. Aby to však mělo vliv na vrchol y , musí být $x \in P(y)$, proto provedeme prvý návrat až do vrcholu $x = \max(P(y))$ a v tomto vrcholu se pokusíme zvýšit barvu.

Nelze-li barvu ve vrcholu x zvýšit (byla by použita barva OMEZ), provedeme další, tentokrát již prostý návrat do vrcholu $x - 1$, $x - 2$ atd.

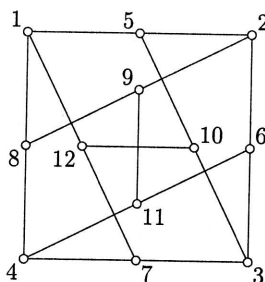
Barvu vrcholu 1 nezvyšujeme, nemá to smysl. (Proč? Zvažte záměnu barev.) Z podobného důvodu nemá smysl barvit vrchol x barvou $b > x$.

Na počátku volíme $OMEZ := n + 1$, aby hodnota OMEZ nebránila nalezení prvního obarvení.

ALGORITMUS:

1. [Inicializace.] $x := 1$; $B(x) := 1$; $OMEZ := n + 1$.
2. [Postup vpřed – obarvení dalšího vrcholu.] Položíme $x := x + 1$. Jestliže $x > n$, pokračujeme podle kroku 3. V opačném případě položíme $B(x) := \text{minimum ze všech barev nevyskytujících se v množině } P(x)$. Jestliže $B(x) \geq OMEZ$, položíme $y := x$ a pokračujeme podle kroku 4, jinak zopakujeme krok 2.
3. [Všechny vrcholy obarveny, počet barev je menší než OMEZ.] Pro všechny vrcholy x grafu G provedeme $BARVA(x) := B(x)$ a dále položíme $OMEZ := \text{maximum z hodnot } BARVA$. Označme y první vrchol, který má barvu OMEZ, a pokračujeme krokem 4.
4. [Návrat – pokus o snížení barvy ve vrcholu y .] Označme x poslední vrchol z množiny $P(y)$ a pokračujeme krokem 5.
5. [Pokračování návratu – pokus o zvýšení barvy ve vrcholu x .] Jestliže $x = 1$, výpočet končí. V opačném případě položíme $b := \text{minimum ze všech barev, které jsou větší než } B(x)$ a nevyskytují se v množině $P(x)$. Pokud $b < OMEZ$ a zároveň $b \leq x$, položíme $B(x) := b$ a pokračujeme podle kroku 2, v opačném případě položíme $x := x - 1$ a zopakujeme krok 5.

13.2.2 Příklad. Činnost algoritmu 13.2.1 předvedeme na grafu z obrázku 13.2. Postup výpočtu je patrný z následující tabulky, kde každý řádek zachycuje hodnoty B po vykonání kroku 4.

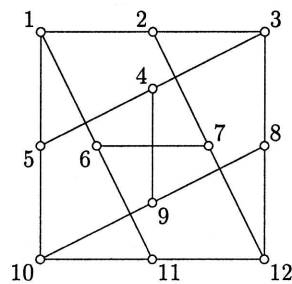


Obrázek 13.2: Graf k příkladu 13.2.2.

1	2	3	4	5	6	7	8	9	10	11	12	
1	1	1	1	2	2	2	2	3	3	4	4	(= obarvení 4 barvami)
1	1	1	1	2	2	2	3	2	3	3	4	
1	1	1	1	2	2	3	2	3	3	4		
1	1	1	1	2	2	3	3	2	3	3	2	(= obarvení 3 barvami)
1	1	1	2	2	2	3						
1	1	2	1	2	3							
1	2	1	1	3								
1												

Čtverečkem je zde vždy vyznačen vrchol y , v němž chceme snížit barvu. Návraty v backtrackingu jsou vyznačeny šipkami a první vrchol, v němž se po návratu zvýšila barva, je v kroužku.

Dodejme, že v tomto příkladě jsme zvolili očíslování vrcholů úmyslně trochu nešikovně, abychom na delším výpočtu mohli demonstrovat jeho postup. Kdybychom očíslovali vrcholy podle obrázku 13.3, proběhl by výpočet velmi rychle. Ověřte!



Obrázek 13.3: Jiné očíslování vrcholů k příkladu 13.2.2.

13.2.3 Poznámka. Rychlost algoritmu 13.2.1 velice záleží na očíslování vrcholů. Doporučuje se nejmenšími čísly očíslovat co největší kliku a vrcholy s velkým stupněm číslovat nižšími čísly. Také je vhodné, aby sousední vrcholy byly číslovány pokud možno sousedními čísly.

13.2.4 Cvičení. Vezměte kořenový strom, který má jeden list v hloubce 2 a ostatní listy v hloubce 1. Určitě víte, že barevnost každého stromu je rovna 2. Přesto tento strom obarvíte algoritmem 13.2.1. Při kterém uspořádání (očíslování) vrcholů proběhne výpočet nejrychleji a při kterém nejpomaleji?

13.2.5 Test k -barevnosti. Potřebujeme-li pouze pro dané k zjistit, zda daný graf lze obarvit k barvami, lze algoritmus 13.2.1 zrychlit: při inicializaci položíme $OMEZ := k + 1$ a v kroku 3, kde je nalezeno obarvení k barvami, můžeme výpočet ukončit. Pokud výpočet skončí v kroku 5, pak obarvení k barvami neexistuje.

13.2.6 Cvičení. Snažíme-li se jednoduchý backtracking urychlit podobně jako v algoritmu 13.2.1, je třeba pečlivě ověřit (dokázat), že urychlením nevynecháme optimální řešení.

Zjistěte, co je špatně na následujícím „vylepšení“ algoritmu 13.2.1, a najděte příklad grafu, kde takto „vylepšený“ postup selže:

Snažíme-li se snížit barvu vrcholu y a nelze-li barvu zvýšit u vrcholu $x = \max(P(y))$, pak neprovedeme další návrat do vrcholu $x - 1$, ale rovnou až do druhého nejvyššího vrcholu z množiny $P(y)$, neboť pouze vrcholy z množiny $P(y)$ ovlivňují barvu vrcholu y .

13.2.7 Heuristické postupy. Algoritmus uvedený v důkaze věty 13.1.8 lze použít pro hrubý odhad barevnosti. Jeho výhodou je fakt, že při vhodném uspořádání vrcholů máme šanci získat přímo optimální obarvení. Doporučuje se volit pořadí vrcholů tak, aby vrcholy s velkým stupněm byly použity dříve. Další možností je opakovat výpočet s náhodně změněným pořadím vrcholů. Při dostatečném počtu opakování pak poměrně rychle roste i pravděpodobnost, že se strefíme do optimálního obarvení.

13.3 Hledání maximálních nezávislých množin

Uvedeme relativně rychlý algoritmus pro vyhledání všech maximálních nezávislých množin založený na backtrackingu. Po jednoduchých úpravách jej lze použít k hledání nejpočetnější nebo nejdražší nezávislé množiny.

13.3.1 Algoritmus pro hledání maximálních nezávislých množin.

VSTUP: Graf G zadáný pomocí seznamu vrcholů a seznamů sousedů $V(x)$.

VÝSTUP: Posloupnost seznamů maximálních nezávislých množin.

POMOCNÉ PROMĚNNÉ: V proměnné R budeme udržovat posloupnost vrcholů r_1, r_2, \dots, r_k , přičemž její prvky budou tvořit k -prvkovou nezávislou množinu. Pro $i \leq k$ označme $R_i = \{r_1, r_2, \dots, r_i\}$.

Dále budeme udržovat dva systémy množin vrcholů N_i a S_i pro všechna $i = 0, 1, \dots, k$, přičemž bude platit:

1. Množiny S_i , N_i a R_i jsou po dvou disjunktní.
2. $(S_i \cup N_i) =$ množina všech vrcholů v takových, že $R_i \cup \{v\}$ je nezávislá množina, tj. vrchol v by bylo možno přidat k R_i .
3. $N_i =$ množina všech vrcholů, které dosud k množině R_i nebyly přidány, tj. žádná nezávislá množina obsahující $R_i \cup \{v\}$ dosud nebyla zkoumána.
4. $S_i =$ množina všech vrcholů v , jejichž přidání k množině R_i již bylo vyzkoušeno, tj. všechny maximální nezávislé množiny obsahující $R_i \cup \{v\}$ již byly prozkoumány.

METODA: Backtracking spočívá v systematickém prodlužování a zkracování posloupnosti R a v odpovídající údržbě množin S_i, N_i, R_i .

Máme-li posloupnost R o délce k , pak k jejímu prodloužení vybereme nějaký vrchol $x \in N_k$. Množiny N_k a S_k uschováme, budeme je potřebovat později, až se posloupnost R opět zkrátí na délku k . Nyní však posloupnost R prodloužíme o vrchol x tím, že položíme $r_{k+1} := x$, vytvoříme množiny N_{k+1} a S_{k+1} tak, aby splňovaly výše uvedené podmínky, a položíme $k := k + 1$.

Z podmínek 1–4 vyplývá, že posloupnost R obsahuje maximální nezávislou množinu právě tehdy, když $N_k = S_k = \emptyset$.

Nelze-li posloupnost R prodloužit, tj. je-li $N_k = \emptyset$, provedeme v backtrackingu návrat, tj. posloupnost zkrátíme o její poslední prvek r_k , položíme $k := k - 1$ a upravíme množiny N_k a S_k .

Návrat provedeme také tehdy, existuje-li vrchol $y \in S_k$ takový, že $V(y) \cap N_k = \emptyset$. V této situaci je totiž jakékoli další prodlužování posloupnosti R zbytečné, protože každé nezávislé množině, která by R obsahovala, by k maximálnosti scházely vrchol y .

ALGORITHMUS:

1. [Inicializace.] Položíme $k := 0$, $S_0 := \emptyset$ a $N_0 := V$. Pokračujeme krokem 2.
2. [Rozšíření nezávislé množiny.] Vybereme libovolný vrchol $x \in N_k$ a položíme $r_{k+1} := x$, $S_{k+1} := S_k \cup \{x\}$, $N_{k+1} := N_k \setminus (V(x) \cup \{x\})$ a $k := k + 1$. Pokračujeme krokem 3.
3. [Test maximálnosti množiny R_k .] Jestliže $N_k = S_k = \emptyset$, vytiskneme posloupnost $R = (r_1, r_2, \dots, r_k)$ a pokračujeme podle kroku 6, jinak pokračujeme krokem 4.
4. [Test možnosti zvětšování.] Jestliže $N_k = \emptyset$, pokračujeme podle kroku 6, jinak pokračujeme krokem 5.
5. [Test, lze-li R_k doplnit na maximální nezávislou množinu.] Jestliže existuje vrchol $y \in S_k$ takový, že $V(y) \cap N_k = \emptyset$, pokračujeme podle kroku 6, jinak pokračujeme krokem 2.
6. [Návrat, zkrácení posloupnosti R .] Jestliže $k = 0$, výpočet končí. V ostatních případech položíme $x := r_k$, $k := k - 1$, $N_k := N_{k+1} \cup \{x\}$ a $S_k := S_{k+1} \setminus \{x\}$. Pokračujeme krokem 4.

13.3.2 Poznámky. Při výpočtu na počítači lze systém množin S_i, N_i a R_i uchovávat v jediné matici o n sloupcích, kde n je počet vrcholů. Pro $i \geq 0$ mohou být v i -tém řádku zakódovány všechny tři zmíněné množiny, neboť jsou disjunktní. Posloupnost R je vhodné udržovat zvlášť. Víme-li, že nezávislost grafu $\alpha(G) < K$, pak stačí, aby matice měla pouze K řádků.

Jinou možností je reprezentovat množiny S_i, N_i jako pole bitů a využít toho při provádění množinových operací.

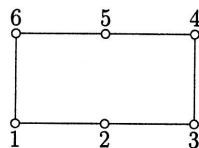
Rychlost práce tohoto algoritmu lze podstatně ovlivnit strategií pro volbu vrcholu x v kroku 2.

13.3.3
vislé m

Prů
a R_k je
do žád
označe
testem

13.3.4
závislý
„bez zá
dlouho,
dávane
i náhod

13.3.3 Příklad. Pomocí algoritmu 13.3.1 vyhledejme všechny maximální nezávislé množiny v grafu z obr. 13.4.



Obrázek 13.4: Graf k příkladu 13.3.3.

Průběh práce algoritmu je patrný z následující tabulky. Obsah množin N_k , S_k a R_k je vyznačen písmeny N, S a R. Znak - znamená, že příslušný vrchol nepatří do žádné z těchto množin. Situace, kdy R_k byla maximální nezávislá množina, jsou označeny hvězdičkou (*). Znak \surd upozorňuje, že byl proveden návrat vyvolaný testem v kroku 5.

k	R_k	N_0, S_0, R_0						N_1, S_1, R_1						N_2, S_2, R_2						N_3, S_3, R_3							
		1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6		
0	\emptyset	N	N	N	N	N	N																				
1	1							R	-	N	N	N	-														
2	13													R	-	R	-	N	-								
3	135*																			R	-	R	-	R	-	*	
2	13																			R	-	R	-	S	-		
1	1							R	-	S	N	N	-														
2	14*													R	-	-	R	-	-	*							
1	1							R	-	S	S	N	-	\surd													
0	\emptyset	S	N	N	N	N	N																				
1	2							-	R	-	N	N	N														
2	24													-	R	-	R	-	N								
3	246*																			-	R	-	R	-	R	*	
2	24																			-	R	-	R	-	S		
1	2							-	R	-	S	N	N														
2	25*													-	R	-	-	R	-	*							
1	2							-	R	-	S	S	N	\surd													
0	\emptyset	S	S	N	N	N	N																				
1	3							S	-	R	-	N	N														
2	35													S	-	R	-	R	-								
1	3							S	-	R	-	S	N														
2	36*													-	-	R	-	-	R	*							
1	3							S	-	R	-	S	S														
0	\emptyset	S	S	S	N	N	N	\surd																			

13.3.4 Heuristické postupy. Pro hledání nejpočetnějších nebo nejdražších nezávislých množin se používají také heuristické algoritmy, které bývají rychlejší, ale „bez záruky“. Nejjednodušší je zvolit některý vrchol a k němu přidávat další tak dlouho, dokud nedostaneme maximální nezávislou množinu. Pravidlo pro volbu přidávaného vrcholu má zásadní vliv na kvalitu výsledného řešení. Volba může být i náhodná, ale doporučuje se preferovat vrcholy s malým stupněm nebo s velkou ce-

nou. Výpočet je vhodné několikrát opakovat s využitím náhody a nakonec si vybrat nejlepší z dosažených řešení.

Také lze použít metodu lokálního průzkumu (viz 11.4, str. 196) k postupnému vylepšování dosaženého řešení např. tak, že jeden vrchol z nezávislé množiny odstraníme a pokoušíme se místo něj zařadit jeden nebo i několik vrcholů jiných.

Ka

R

Rovin
zejm
tříroz
při na
vyplý
někte

14.7

14.1.
 ρ je d
body
jedno
aby ž
svůj v

Je
příčer
kresle
odlišil

R
různý
kterén
planá

14.1.
nakres
(V, E ,
topolo
rovinn

nec si vybrat

t postupnému

vislé množiny

holů jiných.

Kapitola 14

Rovinné grafy

Rovinné grafy, tj. grafy, které lze nakreslit v rovině bez křížení hran, jsou důležité zejména v počítačové grafice, kde se využívá faktu, že struktura vrcholů a hran třírozměrných hranatých těles odpovídá rovinným grafům, a také v elektrotechnice při návrhu integrovaných obvodů a tištěných spojů. Další význam rovinných grafů vyplývá z jejich speciálních vlastností. Pro rovinné grafy lze například zjednodušit některé algoritmy (maximální tok v síti, prohledávání bludiště).

14.1 Nakreslení grafu a topologický rovinný graf

14.1.1 Nakreslení grafu. Formální definice říká, že *nakreslení grafu* G v rovině ρ je dvojice prostých zobrazení ϕ, ψ takových, že ϕ přiřazuje vrcholům grafu různé body roviny ρ a zobrazení ψ přiřazuje každé hraně spojující např. vrcholy $\{x, y\}$ jednoduchou křivku v rovině ρ s krajními body $\phi(x)$ a $\phi(y)$. Přitom požadujeme, aby žádná křivka, která je obrazem hrany, neobsahovala obraz žádného vrcholu jako svůj vnitřní bod.

Jednoduché křivky mohou mít rozličný tvar a mohou se vzájemně protínat, přičemž je nutno rozlišovat obrazy vrcholů od průsečíků křivek. Proto při praktickém kreslení znázorňujeme vrcholy pomocí kroužků, puntíků a podobně, abychom je odlišili od průsečíků.

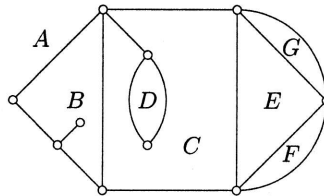
Rovinné nakreslení je takové nakreslení, v němž libovolné dvě křivky, přiřazené různým hranám, mají společně nejvýše dva – a to své krajní – body. Graf, ke kterému existuje rovinné nakreslení, nazýváme *rovinným grafem* nebo také *grafem planárním*.

14.1.2 Topologický rovinný graf je rovinný graf chápaný společně se svým nakreslením. Formálně tedy topologický rovinný graf je šestice $(V, E, \varepsilon, \rho, \phi, \psi)$, v níž (V, E, ε) je graf a (ϕ, ψ) je jeho nakreslení v rovině ρ . Rovinný graf (bez přívlastku topologický) je formálně pouze graf (tj. trojice (V, E, ε)), pro který existuje nějaké rovinné nakreslení.

Vlastnosti rovinného grafu se netýkají konkrétního nakreslení, ale pouze jeho existence nebo dokonce jen možnosti nějaké rovinné nakreslení vytvořit. Vlastnosti topologického rovinného grafu zahrnují navíc všechny vlastnosti jeho konkrétního nakreslení.

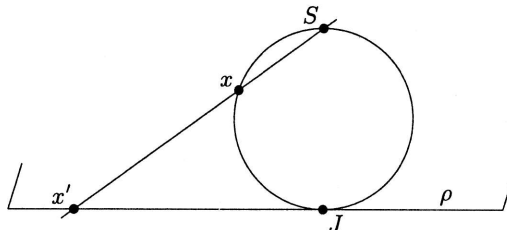
Stěna topologického rovinného grafu je část roviny ohraničená křivkami, které jsou obrazy hran. Jedna ze stěn je vždy neomezená, ostatní jsou omezené. Dvě stěny nazýváme *sousedními*, jestliže společná část jejich hranice je tvořena jednou nebo několika křivkami, které jsou obrazy hran.

Řekneme, že *stěna je incidentní s hranou*, jestliže křivka, která je obrazem hrany, tvoří část hranice (nebo celou hranici) stěny. *Stupeň stěny* definujeme jako počet hran, s nimiž je stěna incidentní, přičemž každou hranu, která je mostem, počítáme dvakrát. Například v grafu na obr. 14.1 má stěna *B* stupeň 6 a stěna *C* stupeň 8. Dvě stěny jsou sousední, jsou-li obě incidentní s touž hranou.



Obrázek 14.1: Stěny topologického rovinného grafu.

14.1.3 Kreslení grafu na kouli, stereografická projekce. Definice nakreslení grafu uvedená v 14.1.1 může být snadno zobecněna i na jiné plochy než roviny. Kreslení grafu na povrch koule má těsný vztah ke kreslení v rovině: graf lze nakreslit bez křížení hran na povrch koule právě tehdy, když je rovinný.



Obrázek 14.2: Stereografická projekce.

Nakreslení na kouli a nakreslení v rovině lze vzájemně převádět mnoha způsoby, jedním z nich je tzv. *stereografická projekce*: Na povrchu koule zvolíme dva souměrně sdružené body *S* a *J* („severní a jižní pól“) a zvolíme rovinu ρ , která se dotýká koule v bodě *J* (viz obr. 14.2). Nyní každému bodu $x \neq S$ z povrchu koule přiřadíme bod x' v rovině takto: body *S* a *x* vedeme přímkou a ta protne rovinu ρ v bodě x' . Toto zobrazení je prosté a zobrazuje celý povrch koule kromě bodu *S* na celou rovinu, existuje tedy inverzní zobrazení. Obě tato zobrazení jsou spojitá, obrazem křivky je tedy opět křivka.

Přev
pól“ *S* t

Je-li
odpovíd
stěny s
stěny o
neomez

POZNÁN
hran ne
nebo „u
„děř“ a

14.1.4

pak je t
 $E(v)$, a
Dvě rov
uspořád

Cykl
nakresle
nakresle

Pom
tačí to,

14.1.5

existuje
neomez

DŮKAZ:
tak, aby
zpět do

Důka

14.1.6

tuje siln

14.1.7

v_1, \dots, v
silně ekv
jako bod

14.1.8

silně ekv

Převádíme-li nakreslení z koule na rovinu, musíme samozřejmě zvolit „severní pól“ S tak, aby neležel v obraze žádné hrany nebo vrcholu.

Je-li graf nakreslen na povrchu koule tak, že se jeho hrany (přesněji křivky odpovídající hranám) nekříží, můžeme i zde definovat pojmy stěna, incidence stěny s hranou, stupeň stěny a sousední stěna. Na povrchu koule jsou všechny stěny omezené, stěně obsahující „severní pól“ S odpovídá v rovinném nakreslení neomezená stěna.

POZNÁMKA: Každý graf lze nakreslit na povrch nějakého tělesa tak, aby se obrazy hran nekřížily. Mnohdy k tomu však potřebujeme složité těleso s mnoha „dírami“ nebo „uchy“. Rovinné grafy jsou ty, které lze nakreslit bez křížení na tělesa bez „děr“ a bez „uch“, tj. na tělesa, která lze získat spojitou deformací z koule.

14.1.4 Silně ekvivalentní nakreslení. Je-li dáno rovinné nakreslení grafu G , pak je tím pro každý jeho vrchol v definováno *cyklické uspořádání hran* z množiny $E(v)$, a to tak, že budeme kolem vrcholu v obíhat ve směru hodinových ručiček. Dvě rovinná nakreslení nazýváme *silně ekvivalentní*, určují-li obě stejná cyklická uspořádání hran.

Cyklická uspořádání hran i silnou ekvivalenci lze definovat také pro grafy nakreslené na povrch koule. Lze dokázat, že jsou-li dána dvě silně ekvivalentní nakreslení, lze jedno z druhého získat spojitou transformací.

Pomocí cyklických uspořádání hran lze snadno a úsporně reprezentovat v počítači to, co je na rovinném nakreslení z grafového hlediska podstatné.

14.1.5 Věta. Nechť A je libovolná stěna topologického rovinného grafu G . Pak existuje silně ekvivalentní nakreslení grafu G takové, že stěna odpovídající A je neomezená.

DŮKAZ: Graf G přeneseme stereografickou projekcí na povrch koule, pootočíme jej tak, aby „severní pól“ S byl uvnitř požadované stěny a toto nakreslení promítneme zpět do roviny. \square

Důkazy tří následujících vět lze najít v knize [Plesník83].

14.1.6 Věta. Ke každému rovinnému nakreslení prostého grafu bez smyček existuje silně ekvivalentní nakreslení takové, že každá hrana je reprezentována úsečkou.

14.1.7 Věta. Buď dán prostý topologický rovinný graf bez smyček s vrcholy v_1, \dots, v_n . Dále nechť je v téže rovině dáno n různých bodů p_1, \dots, p_n . Pak existuje silně ekvivalentní rovinné nakreslení takové, v němž každý vrchol v_i je nakreslen jako bod p_i .

14.1.8 Věta. Každá dvě rovinná nakreslení vrcholově 3-souvislého grafu buď jsou silně ekvivalentní, nebo se takovými stanou zrcadlovým převrácením jednoho z nich.

14.1.9 Mnohostěny. Vypuklé mnohostěny, tj. vypuklá¹ tělesa, jejichž stěny jsou mnohoúhelníky, přirozeným způsobem určují graf: vrcholy mnohostěnu pokládáme za vrcholy grafu a hrany mnohostěnu pokládáme za hrany grafu. Ostatně, odtud pochází naše grafová terminologie.

Je zřejmé, že je-li mnohostěn vypuklý, je jeho graf rovinný.

Lze dokázat, že graf je grafem nějakého vypuklého mnohostěnu právě tehdy, když je rovinný a vrcholově 3-souvislý.

Pravidelný mnohostěn je takový mnohostěn, jehož všechny stěny jsou shodné mnohoúhelníky a všechny vrcholy mají stejný stupeň. Graf pravidelného mnohostěnu je regulární (vrcholy mají stejný stupeň) a navíc také všechny jeho stěny mají stejný stupeň. Pomocí tzv. Eulerovy formule (viz 14.3.1) lze dokázat, že existuje (až na izomorfismus) pouze pět grafů s těmito vlastnostmi, existuje tedy pouze pět pravidelných mnohostěňů. Tyto mnohostěny znali již staří Řekové, říká se jim také *Platonova tělesa*. Grafově teoretický důkaz, že je pouze pět Platonových těles, lze najít v knize [MN96], zde uvedeme pouze číselné charakteristiky Platonových těles:

počet			stupeň	
stěn	vrcholů	hran	vrcholu	stěny
4	4	6	3	3
6	8	12	3	4
8	6	12	4	3
12	20	30	3	5
20	12	30	5	3

Pravidelný dvanáctistěn je nakreslen na obr. 12.1, str. 197. Pravidelný šestistěn asi znáte pod názvem krychle.

14.2 Duální grafy

14.2.1 Dualita neorientovaných rovinných grafů. Mějme souvislý neorientovaný topologický rovinný graf G . *Duální graf* G^D sestrojíme takto: Do každé stěny s grafu G umístíme jeden vrchol s^D grafu G^D . Pro každou hranu e grafu G sestrojíme hranu e^D , která bude spojoval ty vrcholy grafu G^D , které odpovídají stěnám grafu G incidentním s hranou e . Viz příklad na obr. 14.3. Platí:

1. Duální graf je také souvislý a rovinný.
2. Duální graf k duálnímu grafu je původní graf, tj. $(G^D)^D = G$.
3. Dualita převádí vrcholy grafu G na stěny grafu G^D .
4. Smyčce v grafu G odpovídá most v grafu G^D a naopak.
5. Kružnici v grafu G odpovídá v duálním grafu řez $W_{G^D}(A)$ určený množinou vrcholů A duálního grafu, přičemž vrcholy množiny A odpovídají stěnám grafu G ležícím uvnitř kružnice.

Ověřte si to alespoň nakreslením obrázku, lépe však úvahou.

¹Těleso je vypuklé (též *konvexní*), jestliže s každými dvěma body obsahuje i celou úsečku, která tyto dva body spojuje.

14.2.
a sest

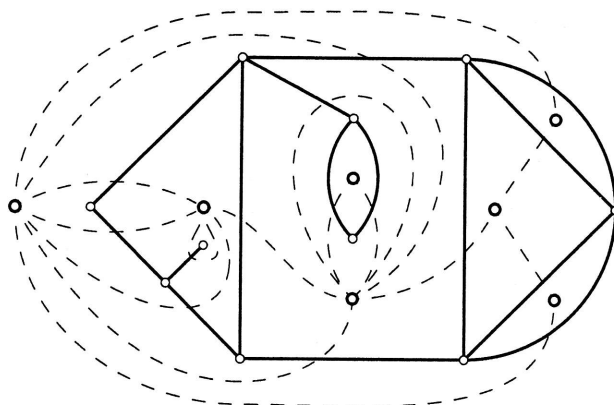
14.2.
topolo
třeba
zleva
směru

Tv
tvrzen
z něj
Da
uveder

jejichž stěny
mnohostěnu po-
rafu. Ostatně,

právě tehdy,

jsou shodné
mno-
stěny jeho stěny
at, že existuje
dy pouze pět
á se jim také
ých těles, lze
nových těles:



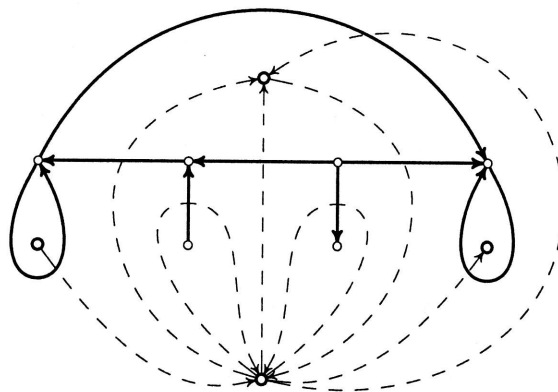
Obrázek 14.3: Duální graf ke grafu z obr. 14.1.

14.2.2 Cvičení. Nakreslete si rovinné grafy všech Platonových těles (viz 14.1.9) a sestrojte k nim grafy duální. Získáte tím opět grafy mnohostěnnů – kterých?

14.2.3 Dualita orientovaných grafů. Duální graf k orientovanému souvislému topologickému grafu je definován stejně jako pro graf neorientovaný, navíc je však třeba určit orientaci hran. Orientaci hrany e^D volíme tak, aby protínala hranu e zleva doprava. Jinými slovy, orientaci hrany e^D získáme pootočením hrany e ve směru hodinových ručiček. Viz příklad na obr. 14.4.

šestistěn

islý neorien-
o: Do každé
u e grafu G
é odpovídají
í:



Obrázek 14.4: Duální graf k orientovanému grafu.

ý množinou
stěnnám grafu

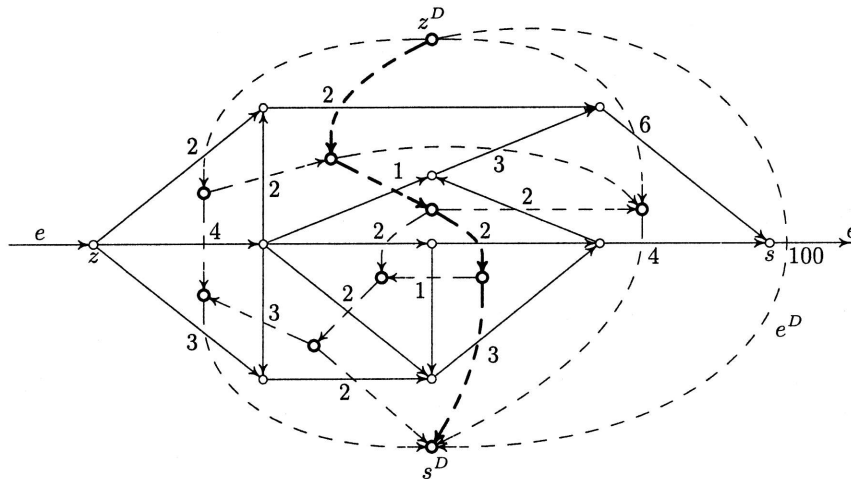
Tvrzení uvedená v 14.2.1 platí analogicky pro orientované grafy s výjimkou tvrzení 2: Duální graf k duálnímu grafu G^D není původní graf G , ale graf, který z něj dostaneme obrácením orientace všech hran.

úsečku, která

Další vztahy mezi topologickým rovinným grafem a jeho duálním grafem jsou uvedeny v 15.3.7, str. 240.

14.2.4 Minimální řez a nejkratší cesta. Uvažujme orientovaný topologický rovinný graf G , jehož hrany jsou ohodnoceny kapacitami a jehož dva vrcholy, zdroj z a spotřebič s , leží na obvodě neomezené stěny. Hledáním maximálního toku v takovéto transportní síti jsme se zabývali v 8.3.15, str. 145.

Vytvořme topologický rovinný graf G' přidáním návratové hrany e ze spotřebiče do zdroje a její kapacitu zvolme tak, aby nemohla omezit tok. Ke grafu G' vytvořme duální graf. Označme z^D počáteční a s^D koncový vrchol hrany e^D v duálním grafu G^D (viz obr. 14.5). Hrany duálního grafu ohodnotme délkami, které jsou číselně rovny kapacitám hran grafu G' .



Obrázek 14.5: Duální graf ke grafu z příkladu 8.3.16, str. 145. Nejkratší cesta odpovídá řezu s minimální kapacitou.

Každé neorientované cestě v duálním grafu z vrcholu z^D do vrcholu s^D odpovídá v grafu G řez, který odděluje zdroj z a spotřebič s . Kapacita řezu je přitom rovna délce cesty, ve které počítáme jenom hrany vpřed. Hrany, kterými cesta prochází proti jejich orientaci, totiž odpovídají hranám, které se v původním grafu díky své orientaci do kapacity řezu nepočítají.

V neorientované rovinné transportní síti je situace jednodušší. Maximální tok nasycuje vždy všechny hrany minimálního řezu, a proto je minimální kapacita řezu rovna délce nejkratší cesty v duálním grafu.

Je třeba upozornit, že v obou případech (orientovaném i neorientovaném) v grafu mohou existovat i řezy, které neodpovídají žádné cestě z vrcholu z^D do vrcholu s^D . Jsou to řezy $W(A)$ určené množinou vrcholů A takovou, že podgraf indukovaný touto množinou není souvislý. Snadno však lze nahlédnout, že žádný takový řez není minimální, neboť komponenta obsahující zdroj z určuje řez s menší kapacitou. Je-li řez určen množinou A takovou, že $z \in A$ a podgraf indukovaný množinou A je souvislý, pak hrany řezu $W(A)$ tvoří v duálním grafu cestu z vrcholu z^D do vrcholu s^D . Tato cesta vede po hranici souvislé oblasti tvořené těmi stěnami duálního grafu, které odpovídají množině A .

14.3 Vlastnosti rovinných grafů

14.3.1 Věta (Eulerova formule). Pro každý souvislý topologický rovinný graf, který má n vrcholů, m hran a s stěn, platí $n + s = m + 2$.

DŮKAZ: Graf G můžeme získat z jeho kostry postupným přidáváním hran, které v kostře neleží. Kostra topologického rovinného grafu má jedinou stěnu a počet hran je $m = n - 1$, proto pro kostru Eulerova formule platí. Přidáním hrany k topologickému rovinnému grafu rozdělíme vždy některou jeho stěnu na dvě. Levá i pravá strana Eulerovy formule se tedy přidáním hrany zvětší o jedničku. Po přidání všech hran tedy dostaneme Eulerovu formuli pro původní daný graf. \square

14.3.2 Cvičení. Dokažte zobecnění Eulerovy formule pro nesouvislé grafy: Pro topologický rovinný graf s n vrcholy, m hranami, s stěnami a k komponentami souvislosti platí $n + s = m + k + 1$.

Následující tři věty zaručují, že prosté rovinné grafy bez smyček nemají „příliš mnoho“ hran, jsou tedy řídké.

14.3.3 Věta. Pro prostý souvislý rovinný neorientovaný graf bez smyček, který má $n \geq 3$ vrcholů a m hran, platí $m \leq 3n - 6$.

DŮKAZ: Poněvadž graf je prostý a bez smyček, má každá stěna (včetně stěny neomezené) stupeň alespoň 3. Součet stupňů všech stěn je roven $2m$, neboť v něm každou hranu počítáme dvakrát. Počet stěn s tedy splňuje $s \leq 2m/3$. Dosadíme-li to do Eulerovy formule, dostaneme $n + 2m/3 \geq m + 2$, a tedy $m \leq 3n - 6$. \square

14.3.4 Věta. Pro prostý souvislý rovinný neorientovaný graf bez smyček a bez trojúhelníků, který má $n \geq 3$ vrcholů a m hran, platí $m \leq 2n - 4$.

DŮKAZ: Stupeň každé stěny grafu je větší nebo roven 4. Součet stupňů všech stěn je roven $2m$, počet stěn s tedy splňuje $s \leq 2m/4 = m/2$. Dosadíme-li to do Eulerovy formule, dostaneme $n + m/2 \geq m + 2$, a tedy $m \leq 2n - 4$. \square

14.3.5 Věta. V každém prostém rovinném neorientovaném grafu bez smyček existuje vrchol, jehož stupeň je menší nebo roven 5.

DŮKAZ: Kdyby každý vrchol měl stupeň 6 nebo více, měl by celý graf alespoň $3n$ hran (je-li n počet vrcholů), což by bylo v rozporu s větou 14.3.3. \square

14.3.6 Cvičení. Pomocí vět 14.3.3 a 14.3.4 dokažte, že grafy K_5 a $K_{3,3}$ nejsou rovinné.

14.3.7 Věta o čtyřech barvách. Rovinný graf bez smyček lze obarvit čtyřmi barvami.

POZNÁMKA: Viz komentář k větě 13.1.13, str. 215.

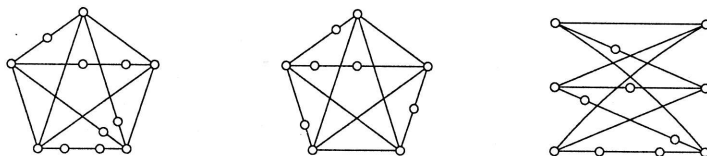
14.4 Charakterizace rovinných grafů

14.4.1 Homeomorfní grafy. Dva grafy G_1, G_2 nazýváme vzájemně *homeomorfními*, jestliže lze jeden získat z druhého těmito operacemi:

Dělení hrany. Mějme hranu e s krajními vrcholy x, y . Hranu e odstraníme a místo ní přidáme nový vrchol v a dvě nové hrany spojující nový vrchol v s vrcholy x a y .

Odstranění vrcholu stupně 2 je opačná operace: vrchol stupně 2 odstraníme spolu s oběma hranami a nahradíme hranou, která spojí bývalé sousedy odstraněného vrcholu.

Obě operace „zachovávají rovinnost“ v tom smyslu, že byl-li graf rovinný před operací, bude rovinný i po ní a naopak. Jsou-li tedy dva grafy homeomorfní, pak jsou buď oba rovinné, anebo oba nerovinné.



Obrázek 14.6: Grafy homeomorfní s grafy K_5 a $K_{3,3}$.

14.4.2 Kuratowského věta. Graf je rovinný právě tehdy, když neobsahuje podgraf homeomorfní s grafem K_5 nebo $K_{3,3}$.

DŮKAZ není jednoduchý, lze jej najít např. v [Berge58]. □

POZNÁMKA: Pomocí Kuratowského věty lze např. o daném grafu dokázat, že není rovinný. Pro praktické zjišťování, zda daný graf je rovinný, to však není vhodný způsob, protože vyhledávání podgrafů je dost obtížné. Hlavní význam Kuratowského věty spočívá v tom, že rovinné grafy charakterizuje pomocí čistě grafových pojmů. Uvědomte si, že pojem rovinného nakreslení není čistě grafovým pojmem, potřebujeme k němu pojem roviny, spojitého zobrazení, křivky apod.

14.4.3 Cvičení. Pomocí Kuratowského věty dokažte, že tzv. *Petersenův graf* nakreslený vlevo na obr. 1.2, str. 18, není rovinným grafem.

14.4.4 Algoritmy pro testování rovinnosti. Kuratowského věta nedává dobrý návod k testování rovinnosti. Existují však rychlé algoritmy, které v čase $O(n)$ nejen rozhodnou, zda graf je rovinný, ale rovinné nakreslení zakódované pomocí cyklických uspořádání hran (viz 14.1.4) dokonce sestojí. Vzhledem ke značné komplikovanosti se jimi nebudeme zabývat, jejich stručný popis lze najít např. v [Kučera83, Plesník83]. Poznamenejme pouze, že tyto algoritmy postupně přidávají kusy grafu ke vhodně zvolené kružnici a rozhodují, zda příslušná část grafu bude nakreslena uvnitř nebo vně kružnice.

Kapitola 15

Cirkulace a potenciály

Tato kapitola je věnována algebraickým vlastnostem cirkulací, potenciálových rozdílů, kružnic a řezů. Tyto vlastnosti jsou důležité zejména v elektrotechnice při analýze elektrických obvodů.

Ke studiu této kapitoly je potřebná alespoň základní znalost lineární algebry, zejména vektorových prostorů.

Poznamenejme, že následující výklad je možno dále zobecnit na vektorové prostory nad obecnými tělesy včetně těles konečných, takže pak lze mluvit např. o vektorovém prostoru kružnic v grafu apod. My se však omezíme na vektorové prostory nad tělesem reálných čísel.

V celé kapitole budeme předpokládat, že hrany grafu jsou pevně očíslovány čísly $1, 2, \dots, m$.

15.1 Vektorový prostor cirkulací

Pojem cirkulace jsme definovali v 8.1.1, str. 129, některé vlastnosti cirkulací jsou uvedeny v 8.5, str. 149. Zde nahlédneme o něco hlouběji do algebraického zákulisí toků v síti. Na rozdíl od kapitoly 8 se zde nebudeme zabývat omezeními toku (kapacitami) a nebudeme rozlišovat toky přípustné a nepřípustné.

Díky pevnému očíslování hran čísly $1, 2, \dots, m$ můžeme každou cirkulaci pokládat za m -členný aritmetický vektor.

15.1.1 Věta. Množina všech cirkulací v grafu G tvoří vektorový prostor, který je podprostorem prostoru \mathbb{R}^m , kde $m = |E(G)|$.

DŮKAZ vyplývá z faktu, že součet cirkulací i násobek cirkulace číslem je opět cirkulace (splňuje Kirchhoffův zákon). \square

15.1.2 Kružnice a cirkulace. Připomeňme, že v 8.3.3, str. 140, jsme hrany libovolné kružnice rozdělili na hrany vpřed a hrany vzad, a to podle vztahu orientace hrany ke směru průchodu kružnicí.

Každé kružnici můžeme přiřadit tzv. *elementární cirkulaci* f předpisem:

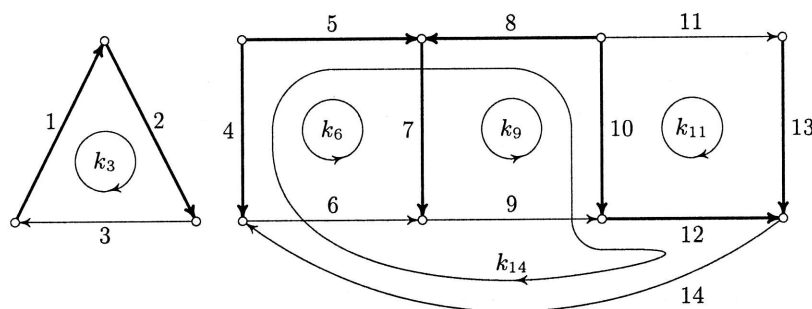
$$f(e) = \begin{cases} 0, & \text{jestliže } e \text{ neleží na kružnici,} \\ 1, & \text{je-li } e \text{ hranou vpřed,} \\ -1, & \text{je-li } e \text{ hranou vzad.} \end{cases}$$

15.1.3 Fundamentální soustava kružnic. Uvažujme graf G a nějaký jeho napnutý les K (viz 4.2.4, str. 58).

Ke každé hraně $e \in E(G) \setminus E(K)$ existuje podle věty 4.2.9 přesně jedna kružnice k_e tvořená hranou e a (některými) hranami lesa K . Tuto kružnici budeme orientovat tak, aby hrana e byla hranou vpřed. Každou takovou kružnici nazveme *fundamentální kružnicí* a množinu všech fundamentálních kružnic nazveme *fundamentální soustavou kružnic* vůči napnutému lesu K .

Každá fundamentální kružnice k_e určuje elementární cirkulaci f_e , která má v hraně e jednotkovou velikost. Takovou elementární cirkulaci nazýváme *fundamentální cirkulací*.

Fundamentální matice kružnic je matice, jejíž řádky jsou tvořeny všemi fundamentálními cirkulacemi chápanými jako řádkové vektory.



Obrázek 15.1: Fundamentální soustava kružnic.

15.1.4 Příklad. Graf na obr. 15.1 má dvě komponenty souvislosti. Napnutý les K je vyznačen silnými čarami. Tento les určuje (jednoznačně) fundamentální soustavu kružnic a fundamentální matici kružnic:

$$\begin{array}{c} e_1 \quad e_2 \quad e_3 \quad e_4 \quad e_5 \quad e_6 \quad e_7 \quad e_8 \quad e_9 \quad e_{10} \quad e_{11} \quad e_{12} \quad e_{13} \quad e_{14} \\ \begin{array}{l} k_3 \\ k_6 \\ k_9 \\ k_{11} \\ k_{14} \end{array} \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & -1 & 1 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 & -1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix} \end{array}$$

Řádky této matice tvoří bázi vektorového prostoru cirkulací, což vyplývá z následující věty.

15.1.5 Věta. V grafu G zvolme libovolný napnutý les K . Potom množina všech fundamentálních cirkulací vzhledem k lesu K tvoří bázi vektorového prostoru cirkulací.

DŮKAZ: Je třeba dokázat, že fundamentální cirkulace jsou lineárně nezávislé a že generují celý vektorový prostor cirkulací.

Nezávislost vyplývá z faktu, že každá hrana $e \in E(G) \setminus E(K)$ je obsažena pouze v jediné fundamentální kružnici, totiž v k_e , a proto pouze jediná fundamentální cirkulace, totiž f_e , je na této hraně nenulová. Má-li tedy být nulová cirkulace (tj. nulový vektor) lineární kombinací fundamentálních cirkulací, musí být všechny koeficienty této lineární kombinace rovny nule. Tím je dokázána nezávislost.

Uvažujme nyní libovolnou cirkulaci f v grafu G . Ukážeme, že f je lineární kombinací fundamentálních cirkulací. Vezměme postupně všechny hrany $e \in E(G) \setminus E(K)$ a od cirkulace f odečteme vždy $f(e)$ -násobek fundamentální cirkulace f_e . Je zřejmé, že tím vynulujeme cirkulaci na všech hranách z množiny $E(G) \setminus E(K)$. Protože však zbývající hrany neobsahují žádnou kružnici (jde o hrany lesa K), musí být výsledná cirkulace nulová i na nich. Z této úvahy plyne, že pro libovolně zvolenou cirkulaci f platí

$$f = \sum_{e \in E(G) \setminus E(K)} f(e) \cdot f_e,$$

a tedy f je lineární kombinací fundamentálních cirkulací. \square

15.1.6 Věta. V grafu o n vrcholech, m hranách a k komponentách souvislosti je dimenze vektorového prostoru cirkulací rovna $m - n + k$.

DŮKAZ: Dimenze je rovna počtu prvků báze, tedy podle předchozí věty počtu fundamentálních cirkulací, tj. počtu všech hran mimo napnutý les. Podle věty 4.2.8 má každý napnutý les $n - k$ hran. Počet hran mimo napnutý les (a tedy i dimenze vektorového prostoru cirkulací) je tedy $m - n + k$. \square

15.1.7 Cyklomatické číslo. Dimenzi vektorového prostoru cirkulací nazýváme *cyklomatickým číslem grafu*.

15.1.8 Poznámka. Vektorový prostor cirkulací má i jiné báze než fundamentální cirkulace určené nějakým napnutým lesem. V topologických rovinných grafech lze například získat bázi prostoru cirkulací ze soustavy kružnic určených omezenými stěnami. Ověřte na příkladě. Viz též následující cvičení.

15.1.9 Cvičení. Vezměte topologický rovinný graf z obrázku 1.1, str. 18 uprostřed, a zvolte jeho libovolnou orientaci. Ověřte, že čtyři elementární cirkulace určené kružnicemi kolem omezených stěn tvoří bázi prostoru cirkulací. Ověřte také, že tato báze není fundamentální bázi vzhledem k žádnému napnutému lesu (tj. kostře) daného grafu.

15.2 Potenciály a potenciálové rozdíly

15.2.1 Potenciál a potenciálový rozdíl. Buď dán orientovaný graf G . *Potenciálem* nazveme jakékoli ohodnocení vrcholů $p : V(G) \rightarrow \mathbb{R}$ grafu G reálnými čísly.

Ohodnocení hran $\pi : E(G) \rightarrow \mathbb{R}$ reálnými čísly nazýváme *potenciálovým rozdílem*, jestliže existuje potenciál p takový, že pro každou hranu e grafu platí:

$$(15.1) \quad \pi(e) = p(Kv(e)) - p(Pv(e)).$$

Všimněte si, že potenciálový rozdíl π se nezmění, zvýšíme-li potenciál p o stejnou konstantu ve všech vrcholech některé souvislé komponenty.

Zdaleka ne každé ohodnocení hran je potenciálovým rozdílem. Najděte příklad! Návod, jak ověřit, zda dané ohodnocení je potenciálovým rozdílem, podáme v následující větě 15.2.2, úspornější verzi pak ve větě 15.3.4.

Dále poznamenejme, že máme-li pevně zvoleno očíslování hran čísly $1, 2, \dots, m$, pak každý potenciálový rozdíl můžeme pokládat za m -členný aritmetický vektor.

15.2.2 Věta. Pro libovolnou kružnici k označme C_k^+ množinu hran vpřed a C_k^- množinu hran vzad. Ohodnocení hran π je potenciálovým rozdílem právě tehdy, když pro každou kružnici k v grafu platí:

$$(15.2) \quad \sum_{e \in C_k^+} \pi(e) - \sum_{e \in C_k^-} \pi(e) = 0.$$

POZNÁMKA: Podmínka (15.2) je analogií druhého Kirchhoffova zákona (součet napětí podél obvodu je roven nule). Ve větě 15.3.4 ukážeme, že podmínku (15.2) stačí ověřit pro fundamentální soustavu kružnic.

DŮKAZ: Nechť π je potenciálový rozdíl a p je příslušný potenciál. Vezměme libovolnou kružnici, tj. posloupnost vrcholů a hran

$$v_0, e_1, v_1, e_2, v_2, \dots, e_k, v_k,$$

kde $v_k = v_0$. Pro každou její hranu platí:

$$\begin{aligned} p(v_i) - p(v_{i-1}) &= \pi(e_i), \quad \text{je-li } e_i \text{ hranou vpřed, a} \\ p(v_i) - p(v_{i-1}) &= -\pi(e_i), \quad \text{je-li } e_i \text{ hranou vzad.} \end{aligned}$$

Sečtením těchto rovnic pro všechny hrany kružnice dostaneme platnost podmínky (15.2):

$$\sum_{e \in C_k^+} \pi(e) - \sum_{e \in C_k^-} \pi(e) = \sum_{i=1}^k (p(v_i) - p(v_{i-1})) = 0.$$

Nechť nyní naopak π je ohodnocení hran, které splňuje podmínku (15.2) pro každou kružnici k . Popíšeme postup, který vede k nalezení potenciálu p splňujícího podmínku (15.1).

V každé souvislé komponentě zvolíme libovolný vrchol v a ohodnotíme jej libovolným potenciálem $p(v)$. Dalším vrcholům v komponentách vypočteme jejich potenciály postupem, který připomíná značkovací proceduru: Existuje-li hrana e taková, že $Pv(e)$ je ohodnocen a $Kv(e)$ nikoli, položíme

$$p(Kv(e)) := p(Pv(e)) + \pi(e).$$

Je-li naopak hrana e taková, že $Kv(e)$ je ohodnocen a $Pv(e)$ není, položíme

$$p(Pv(e)) := p(Kv(e)) - \pi(e).$$

Platnost podmínky (15.2) pro všechny kružnice zaručuje, že každá hrana, jejíž oba vrcholy jsou již ohodnoceny, splňuje podmínku (15.1). \square

15.2.3 Věta. Množina všech potenciálových rozdílů v grafu G tvoří vektorový prostor, který je podprostorem prostoru \mathbb{R}^m , kde m je počet hran.

DŮKAZ: Vezměme dva libovolné potenciálové rozdíly π_1, π_2 . K nim existují potenciály p_1, p_2 takové, že pro každou hranu e platí:

$$\begin{aligned}\pi_1(e) &= p_1(Kv(e)) - p_1(Pv(e)), \\ \pi_2(e) &= p_2(Kv(e)) - p_2(Pv(e)).\end{aligned}$$

Součet potenciálů $p_1 + p_2$ je samozřejmě také potenciál, a poněvadž

$$\pi_1(e) + \pi_2(e) = p_1(Kv(e)) + p_2(Kv(e)) - p_1(Pv(e)) - p_2(Pv(e)),$$

je součet potenciálových rozdílů $\pi_1 + \pi_2$ také potenciálovým rozdílem. Podobně násobek potenciálu je potenciál, a tedy i násobek potenciálového rozdílu je potenciálový rozdíl. \square

15.2.4 Řezy a potenciálové rozdíly. Každému řezu $W_G(A)$ v orientovaném grafu G můžeme přiřadit potenciál p definovaný předpisem

$$p(v) = \begin{cases} 0 & \text{pro } v \in A, \\ 1 & \text{pro } v \notin A. \end{cases}$$

Odpovídající potenciálový rozdíl π nazýváme *elementárním potenciálovým rozdílem*. Platí:

$$\pi(e) = \begin{cases} 1 & \text{pro } e \in W^+(A), \\ -1 & \text{pro } e \in W^-(A), \\ 0 & \text{pro } e \notin W(A). \end{cases}$$

Připomeňme, že zvláštním případem řezu je okolí $E(v)$ vrcholu v .

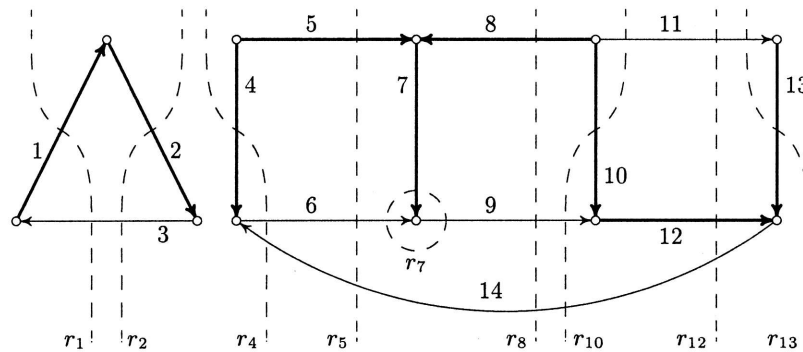
15.2.5 Fundamentální soustava řezů. Mějme graf G a jeho napnutý les K .

Odstraněním libovolné hrany $e \in E(K)$ se podle věty 4.2.9 rozpadne některá komponenta lesa K (tj. strom) na dvě komponenty. Označme A tu z nich, která obsahuje počáteční vrchol hrany e , druhou z nich označme B . *Fundamentální řez* r_e určený hranou e lesa K nyní definujeme jako řez $W_G(A)$.

Množinu všech fundamentálních řezů vůči napnutému lesu K nazýváme *fundamentální soustavou řezů*.

Fundamentální potenciálový rozdíl π_e určený hranou e napnutého lesa definujeme jako elementární potenciálový rozdíl, určený fundamentálním řezem r_e . Platí tedy $\pi_e(e) = 1$ a potenciál p_e , který tomuto potenciálovému rozdílu odpovídá, splňuje $p_e(A) = 0$ a $p_e(B) = 1$, kde A, B jsou komponenty lesa K oddělované hranou e .

Fundamentální matice řezů je matice, jejíž řádky jsou tvořeny všemi fundamentálními potenciálovými rozdíly, chápanými jako řádkové vektory.



Obrázek 15.2: Fundamentální soustava řezů.

15.2.6 Příklad. Fundamentální soustava řezů v grafu z obrázku 15.1 vzhledem k témuž napnutému lesu K je na obrázku 15.2 znázorněna čárkovaně. Fundamentální matice řezů má tvar

$$\begin{array}{c}
 r_1 \\
 r_2 \\
 r_4 \\
 r_5 \\
 r_7 \\
 r_8 \\
 r_{10} \\
 r_{12} \\
 r_{13}
 \end{array}
 \begin{pmatrix}
 e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 & e_8 & e_9 & e_{10} & e_{11} & e_{12} & e_{13} & e_{14} \\
 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\
 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 1 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & -1 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & -1 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0
 \end{pmatrix}$$

Řádky této matice tvoří bázi vektorového prostoru potenciálových rozdílů, což vyplývá z následující věty.

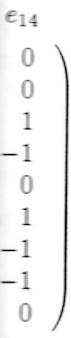
napnutý les K .
 spadne některá
 u z nich, která
 fundamentální řez

zýváme funda-

ho lesa definu-
 řezem r_e . Platí
 odpovídá, spl-
 vané hranou e .
 emi fundamen-



15.1 vzhledem
 ně. Fundamen-



ch rozdílly, což

15.2.7 Věta. V grafu G zvolme libovolný napnutý les K . Potom množina všech fundamentálních potenciálových rozdílů vzhledem k lesu K tvoří bázi vektorového prostoru potenciálových rozdílů.

DŮKAZ je podobný důkazu věty 15.1.5 pro cirkulace. Nezávislost fundamentálních potenciálových rozdílů vyplývá z faktu, že každá hrana $e \in E(K)$ je obsažena pouze v jediném fundamentálním řezu.

Vezměme nyní libovolný potenciálový rozdíl π . Ukážeme, že π je lineární kombinací fundamentálních potenciálových rozdílů. Uvažujme postupně všechny hrany $e \in E(K)$ a od potenciálového rozdílů π odečtíme vždy $\pi(e)$ -násobek fundamentálního potenciálového rozdílů π_e . Je zřejmé, že tím získáme potenciálový rozdíl

$$\pi' = \pi - \sum_{e \in E(K)} \pi(e) \cdot \pi_e,$$

který je nulový na všech hranách lesa K . Odpovídající potenciál tedy nabývá týchž hodnot vždy na celé souvislé komponentě grafu G , proto je potenciálový rozdíl π' nulový na všech hranách grafu (hrany vedou jen mezi vrcholy téže komponenty). Tím je dokázáno, že

$$\pi = \sum_{e \in E(K)} \pi(e) \cdot \pi_e$$

je lineární kombinací fundamentálních potenciálových rozdílů. □

15.2.8 Věta. V grafu o n vrcholech a k komponentách souvislosti je dimenze vektorového prostoru potenciálových rozdílů rovna $n - k$.

DŮKAZ: Dimenze je rovna počtu prvků báze, tedy podle předchozí věty počtu fundamentálních potenciálových rozdílů, tj. počtu hran napnutého lesa, který podle věty 4.2.8 má přesně $n - k$ hran. □

15.2.9 Poznámka. Vektorový prostor potenciálových rozdílů má i jiné báze než fundamentální soustavy určené nějakým napnutým lesem. Důležité jsou zejména báze tvořené z řezů tvaru $E(x)$, neboť příslušné elementární potenciálové rozdílly jsou shodné s řádky incidenční matice.

15.2.10 Věta. Vektorový prostor potenciálových rozdílů grafu G je generován množinou elementárních potenciálových rozdílů určených řezy tvaru $E(x)$, kde x je vrchol grafu G .

DŮSLEDEK: Vektorový prostor potenciálových rozdílů je generován řádky incidenční matice. Z řádků incidenční matice tedy lze vybrat bázi vektorového prostoru potenciálových rozdílů.

DŮKAZ: Dokážeme, že z elementárních potenciálových rozdílů určených okolními vrcholů, tj. řezy ve tvaru $E(x)$, lze získat jako lineární kombinaci libovolný elementární potenciálový rozdíl. Z toho pak již snadno plyne, že lze vygenerovat i jakoukoli fundamentální soustavu potenciálových rozdílů a ta již podle 15.2.7 generuje celý prostor.

Vezměme libovolný řez $W(A)$ a jím určený potenciálový rozdíl π . Označme π_x elementární potenciálový rozdíl určený okolím $E(x)$ libovolného vrcholu $x \in A$. Ukážeme, že pro všechny hrany grafu platí:

$$(15.3) \quad \pi(e) = \sum_{x \in A} \pi_x(e).$$

Pro hrany z řezu $W(A)$ tato rovnost vyplývá z faktu, že hrana e je obsažena pouze v jedné z množin $E(x)$ a příslušný sčítanec má správné znaménko. Má-li hrana e oba vrcholy v množině A , je obsažena ve dvou množinách hran $E(Pv(e))$ a $E(Kv(e))$, přičemž na hraně e mají příslušné sčítance opačná znaménka, a součet je tedy nulový. Hrany, jejichž oba vrcholy leží mimo množinu A , neleží v žádném z uvažovaných okolí, a proto je na nich součet také nulový. Platí tedy:

$$\pi = \sum_{x \in A} \pi_x.$$

Potenciálový rozdíl π tedy je lineární kombinací elementárních potenciálových rozdílů, které jsou určeny okolími $E(x)$ vrcholů $x \in A$. \square

15.2.11 Hodnost grafu. Dimenze vektorového prostoru potenciálových rozdílů je rovna hodnosti incidenční matice grafu. Tuto dimenzi nazýváme též *hodností grafu*.

15.3 Vztahy cirkulací a potenciálových rozdílů

15.3.1 Skalární součin a kolmost vektorů. Dva vektory $u, v \in \mathbb{R}^m$ jsou *kolmé*, je-li jejich skalární součin roven nule. Jsou-li P a P' dva podprostory téhož prostoru \mathbb{R}^m a je-li každý vektor $z P$ kolmý na každý vektor $z P'$, pak prostory P a P' nazýváme vzájemně kolmými. Z vlastností skalárního součinu snadno plyne toto tvrzení:

Je-li každý vektor z množiny generátorů prostoru P kolmý na každý vektor z množiny generátorů prostoru P' , pak prostory P a P' jsou vzájemně kolmé.

15.3.2 Věta. Vektorový prostor cirkulací a vektorový prostor potenciálových rozdílů jsou vzájemně kolmé.

DŮKAZ: Stačí dokázat, že libovolná fundamentální cirkulace je kolmá na libovolný fundamentální potenciálový rozdíl vůči těmto napnutému lesu. To je však snadné: Stačí si uvědomit, že libovolná fundamentální kružnice a libovolný fundamentální řez buď nemají žádnou společnou hranu, nebo mají přesně dvě společné hrany. Tyto dvě hrany buď jsou stejně orientovány vůči kružnici (obě vpřed nebo obě vzad) a rozdílně vůči řezu (jedna dovnitř druhá ven), nebo jsou rozdílně orientovány vzhledem ke kružnici a stejně vůči řezu. Ve všech těchto případech mají ve skalárním součinu činitelé odpovídající těmto hranám opačná znaménka a vzájemně se ruší. Skalární součin fundamentální cirkulace a fundamentálního potenciálového rozdílu je tedy nulový. \square

15.3.3 Lemma. Mějme kružnici k a jí odpovídající elementární cirkulaci f . Potom ohodnocení hran π splňuje podmínku (15.2) (druhý Kirchhoffův zákon) právě tehdy, když vektory π a f jsou kolmé.

DŮKAZ vyplývá bezprostředně z definice elementární cirkulace a skalárního součinu. \square

15.3.4 Věta. Buď G orientovaný graf a π ohodnocení jeho hran. Potom následující podmínky jsou ekvivalentní:

1. π je potenciálový rozdíl.
2. Podmínka (15.2) (druhý Kirchhoffův zákon) platí pro každou kružnici.
3. Podmínka (15.2) platí pro každou fundamentální kružnici.
4. Vektor π je kolmý na každou fundamentální cirkulaci vzhledem k nějakému napnutému lesu K .
5. Vektor π je kolmý na každou cirkulaci f v grafu G .

DŮKAZ: $1 \Leftrightarrow 2$ vyplývá z věty 15.2.2; $2 \Rightarrow 3$ platí triviálně; $3 \Rightarrow 4$ vyplývá z lemmatu 15.3.3; $4 \Rightarrow 5$ vyplývá z vlastností vektorových prostorů; $5 \Rightarrow 2$ vyplývá opět z lemmatu 15.3.3. \square

15.3.5 Lemma. Mějme řez $W(A)$ a jemu odpovídající elementární potenciálový rozdíl π . Potom ohodnocení hran f splňuje podmínku

$$(15.4) \quad \sum_{e \in W^+(A)} f(e) - \sum_{e \in W^-(A)} f(e) = 0$$

právě tehdy, když vektory f a π jsou kolmé.

DŮKAZ plyne bezprostředně z definice elementárního potenciálového rozdílu. \square

15.3.6 Věta. Buď G orientovaný graf a f ohodnocení jeho hran. Potom následující podmínky jsou ekvivalentní:

1. f je cirkulace.
2. Podmínka (15.4) platí pro každý řez.
3. Podmínka (15.4) platí pro každý fundamentální řez.
4. Vektor f je kolmý na každý fundamentální potenciálový rozdíl vzhledem k nějakému napnutému lesu K .
5. Vektor f je kolmý na každý potenciálový rozdíl.

DŮKAZ: $1 \Leftrightarrow 2$ je zřejmé z definice cirkulace a řezu; $2 \Rightarrow 3$ platí triviálně; $3 \Rightarrow 4$ vyplývá z lemmatu 15.3.5; $4 \Rightarrow 5$ vyplývá z vlastností vektorových prostorů; $5 \Rightarrow 2$ opět vyplývá z lemmatu 15.3.5. \square

15.3.7 Cirkulace a potenciálové rozdíly v rovinných grafech mají další zajímavé vlastnosti. Uvažujme orientovaný topologický rovinný graf G a k němu duální graf G^D (viz 14.2.3, str. 227). Každou cirkulaci v grafu G můžeme pokládat za potenciálový rozdíl v G^D , a také naopak každý potenciálový rozdíl v G můžeme pokládat za cirkulaci v G^D . (Ověřte na příkladě!) Z toho také plyne, že cykломatické číslo grafu G je rovno hodnotě grafu G^D , a naopak hodnota grafu G je rovna cykломatickému číslu grafu G^D .

15.4 Analýza elektrické sítě

V tomto oddíle naznačíme, jak lze vlastnosti vektorových prostorů cirkulací a potenciálových rozdílů použít při analýze elektrické sítě složené ze zdrojů stejnosměrného napětí a z rezistorů. Podrobnější výklad týkající se obecnějších obvodů by přesahoval rámec této knížky, zájemce odkazujeme na knihy [SwTh81, KŠCh89] a na speciální elektrotechnickou literaturu.

15.4.1 Kirchhoffovy zákony. Uvažujme elektrickou síť složenou z rezistorů a ze zdrojů napětí. Síť snadno znázorníme orientovaným grafem. Orientaci hran můžeme zvolit libovolně. Vrcholy grafu odpovídají uzlům elektrické sítě (vodičům), hrany odpovídají větvím. V každé větvi může být jeden nebo několik zdrojů napětí a jeden nebo několik rezistorů. Každou větev e ovšem můžeme zjednodušit tak, že obsahuje vždy jeden zdroj napětí ε_e (může mít i nulovou velikost) a jeden rezistor o odporu R_e : pro tento účel ve větvi sečteme napětí všech zdrojů a podobně sečteme odpory všech rezistorů a vnitřní odpory všech zdrojů.

Proud protékající větví e označíme I_e , napětí U_e na této větvi je součtem napětí zdroje ε_e a úbytku napětí na rezistoru, který podle Ohmova zákona je roven $R_e I_e$. Platí tedy $U_e = \varepsilon_e - R_e I_e$.

Proudy a napětí v síti musí splňovat 1. a 2. Kirchhoffův zákon. Jinými slovy, ohodnocení hran I_e musí být cirkulací ve smyslu definice 8.1.1, str. 129, a ohodnocení hran U_e musí pro každý obvod (tj. kružnici) splňovat podmínku (15.2), a tedy podle věty 15.2.2 musí být potenciálovým rozdílem.

Všechny tyto podmínky můžeme snadno vyjádřit pomocí soustavy lineárních rovnic, v nich jako neznámé vystupují proudy v jednotlivých hranách I_e . U podmínek z 1. Kirchhoffova zákona to je zcela přímočaré. Pro podmínky plynoucí z 2. Kirchhoffova zákona vyjádříme napětí U_e pomocí vztahu $U_e = \varepsilon_e - R_e I_e$, v němž ε_e pokládáme za konstantu a převedeme ji na pravou stranu. Typická rovnice pro některou kružnici tedy má tvar

$$R_1 I_1 + R_2 I_2 + \dots = \varepsilon_1 + \varepsilon_2 + \dots$$

Soustava rovnic pro všechny vrcholy a všechny kružnice má ovšem zbytečně mnoho rovnic a je lineárně závislá. Pro řešení soustavy je výhodné, aby rovnice byly lineárně nezávislé. Toho lze docílit takto:

Zvolíme kostru grafu (předpokládáme, že síť je souvislá) a vzhledem k této kostře nalezneme fundamentální soustavu řezů a fundamentální soustavu kružnic. Rovnice

pro proudy
pro napětí (15.3.4 a 15.3.5) a fundamentální soustavu všech kružnic

Podle věty 15.2.2 je počet m je počet m velikosti prvků Ohmova zákona

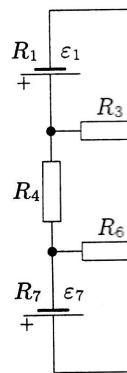
Výpočet rovnic. Rovnice totiž mají kostry jako kostru. Tato fundamentální soustavu smyčkových proudů již snadno

Lze dokázat a že její pr

(15.5)

kde k_{it} , k_{jt} jako matic transponované jsou rovny

Celý po



Obráz

pro proudy (1. Kirchhoffův zákon) sestavíme pouze pro fundamentální řezy, rovnice pro napětí (2. Kirchhoffův zákon) sestavíme pouze pro fundamentální kružnice. Z vět 15.3.4 a 15.3.6 vyplývá, že platí-li oba Kirchhoffovy zákony pro fundamentální řezy a fundamentální kružnice, platí oba v plném rozsahu, tj. pro všechny řezy i pro všechny kružnice (obvody).

Podle vět 15.1.6 a 15.2.8 má tato soustava celkem m rovnic o m neznámých, kde m je počet hran grafu. Kdybychom tuto soustavu rovnic vyřešili, získali bychom velikosti proudů v jednotlivých větvích sítě. Napětí bychom pak dopočítali podle Ohmova zákona.

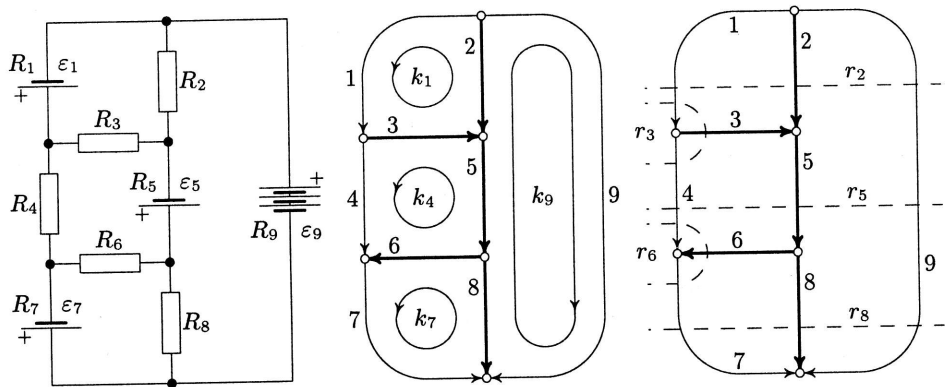
Výpočet však lze ještě dále usnadnit podstatným zmenšením velikosti soustavy rovnic. Rovnice sestavené podle 1. Kirchhoffova zákona pro fundamentální řezy totiž mají velmi jednoduchý tvar a umožňují snadno vyjádřit proudy v hranách kostry jako lineární kombinace tzv. *smyčkových proudů*, tj. proudů v hranách mimo kostru. Tato vyjádření dosadíme do rovnic získaných ze 2. Kirchhoffova zákona pro fundamentální kružnice, čímž dostaneme soustavu $m - n + 1$ rovnic pro $m - n + 1$ smyčkových proudů. Řešením této soustavy získáme smyčkové proudy a z nich pak již snadno dopočítáme proudy v ostatních hranách.

Lze dokázat, že matice S soustavy rovnic pro smyčkové proudy je symetrická a že její prvky s_{ij} lze získat i zcela mechanickým způsobem podle vzorce

$$(15.5) \quad s_{ij} = \sum_{t=1}^n R_t k_{it} k_{jt} ,$$

kde k_{it} , k_{jt} jsou prvky fundamentální matice kružnic K . Matici S lze získat také jako maticový součin $S = KRK^T$, kde K je fundamentální matice kružnic, K^T je transponovaná fundamentální matice kružnic a R je matice, jejíž prvky na diagonále jsou rovny odporům R_i a ostatní prvky jsou nulové.

Celý postup výpočtu předvedeme na příkladě.



Obrázek 15.3: Příklad elektrické sítě a fundamentální soustavy kružnic a řezů.

15.4.2 Příklad. Nalezneme proudy a napětí v jednotlivých větvích elektrické sítě z obr. 15.3 na předchozí straně. Rovnice vyjadřující 1. Kirchhoffův zákon pro fundamentální soustavu řezů mají tvar

$$(15.6) \quad \begin{aligned} r_2 : \quad I_1 + I_2 + I_9 &= 0, \\ r_3 : \quad -I_1 + I_3 + I_4 &= 0, \\ r_5 : \quad I_4 + I_5 + I_9 &= 0, \\ r_6 : \quad I_4 + I_6 - I_7 &= 0, \\ r_8 : \quad I_7 + I_8 + I_9 &= 0 \end{aligned}$$

a rovnice vyjadřující 2. Kirchhoffův zákon pro fundamentální soustavu kružnic mají tvar

$$(15.7) \quad \begin{aligned} k_1 : \quad R_1 I_1 - R_2 I_2 + R_3 I_3 &= \varepsilon_1, \\ k_4 : \quad -R_3 I_3 + R_4 I_4 - R_5 I_5 - R_6 I_6 &= -\varepsilon_5, \\ k_7 : \quad R_6 I_6 + R_7 I_7 - R_8 I_8 &= \varepsilon_7, \\ k_9 : \quad -R_2 I_2 - R_5 I_5 - R_8 I_8 + R_9 I_9 &= -\varepsilon_9 - \varepsilon_5. \end{aligned}$$

Řešením této soustavy 9 rovnic o 9 neznámých bychom mohli přímo získat proudy v jednotlivých větvích. Méně pracný však je tento postup: Proudů ve větvích, které jsou částí kostry, vyjádříme ze soustavy (15.6) jako lineární kombinace ostatních. Stejně vztahy však lze snadno vyčíst přímo z grafu na obr. 15.3 vpravo.

$$(15.8) \quad \begin{aligned} I_2 &= -I_1 - I_9, \\ I_3 &= I_1 - I_4, \\ I_5 &= -I_4 - I_9, \\ I_6 &= -I_4 + I_7, \\ I_8 &= -I_7 - I_9. \end{aligned}$$

Nyní dosadíme ze vztahů (15.8) do soustavy (15.7), čímž získáme pouze $m - n + 1 = 4$ rovnice o 4 neznámých I_1, I_4, I_7, I_9 :

$$(15.9) \quad \begin{aligned} (R_1 + R_2 + R_3)I_1 - R_3 I_4 + R_2 I_9 &= \varepsilon_1, \\ -R_3 I_1 + (R_3 + R_4 + R_5 + R_6)I_4 - R_6 I_7 + R_5 I_9 &= -\varepsilon_5, \\ -R_6 I_4 + (R_6 + R_7 + R_8)I_7 + R_8 I_9 &= \varepsilon_7, \\ R_2 I_1 + R_5 I_4 + R_8 I_7 + (R_2 + R_5 + R_8 + R_9)I_9 &= -\varepsilon_9 - \varepsilon_5. \end{aligned}$$

Vyřešením této soustavy získáme proudy I_1, I_4, I_7, I_9 a dosazením do (15.8) zbývajících proudů. Ověřte, že tytéž koeficienty soustavy rovnic (15.9) poskytuje i vzorec (15.5).

15.4.3 Cvičení. Vyřešte příklad 15.4.2 pro konkrétní hodnoty $\varepsilon_1 = \varepsilon_5 = \varepsilon_7 = 6\text{V}$, $\varepsilon_9 = 12\text{V}$, $R_1 = R_5 = R_7 = 1\ \Omega$, $R_9 = 2\ \Omega$, $R_2 = R_3 = R_4 = R_6 = R_8 = 10\ \Omega$.

Pak zvolte jinou kostru, popř. i jinou orientaci hran a řešení opakujte. Výsledek musí být samozřejmě týž.

Výsledky cvičení

V-1.3.2 (str. 16).

2. Platí $d(x) \geq |E(x)| \geq |V(x)|$.
3. Platí $m(x, x) = m^+(x, x) =$ počet smyček ve vrcholu x .
4. Nemůže. Její vrchol by musel zároveň patřit i nepatřit do množiny A .
5. Platí $|W(A)| \geq |V(A) \setminus A|$.

V-1.4.3 (str. 18). Vrcholy označte počínaje levým vrcholem ve směru hodinových ručiček d, c, e, b, f, a . Není to jediná možnost.

V-1.4.4 (str. 18). Prvé tři jsou navzájem izomorfní, čtvrtý nikoli.

V-1.5.4 (str. 20). Ze sledu, který není cestou, můžeme vynecháním částí mezi opakujícími se vrcholy získat cestu, která vede z téhož vrcholu do téhož vrcholu, a zajišťuje tedy stejnou dostupnost.

V-1.8.2 (str. 24). Dvanáct hran. Kdyby graf H měl navíc smyčku, bylo by to 13 hran.

V-1.8.3 (str. 25). Mohou. Všechny hrany obou grafů jsou smyčkami, neboť jiné hrany by se v obou maticích sousednosti projevily nestejnými hodnotami.

V-1.8.5 (str. 25). Pro orientovaný graf je

$$BB^T = \begin{cases} -m^+(i, j) & \text{pro } i \neq j, \\ d(i) & \text{pro } i = j, \end{cases}$$

zatímco pro neorientovaný graf je

$$BB^T = \begin{cases} m(i, j) & \text{pro } i \neq j, \\ d(i) & \text{pro } i = j. \end{cases}$$

V-2.2.9 (str. 41). Vzhledem k obrovskému počtu vrcholů (několik trilionů) je hledání cest v takovém grafu nereálné. V takových případech je vždy třeba využít speciálních vlastností konkrétní úlohy.

V-3.1.6 (str. 49). V kroku 2 vybíráme vždy hranu, jejíž jeden vrchol má značku a druhý nikoli.

V-3.1.7 (str. 49). V algoritmu 3.1.1, str. 47, upraveném podle cvičení 3.1.6, zařadíme do kroku 2 test, zda existuje hrana, která spojuje dva označované vrcholy a která nebyla použita v kroku 3 k označování žádného z těchto dvou vrcholů. K tomu lze použít hodnoty ODKUD.

V-3.3.8 (str. 55). Není to pravda. Například v grafu na obr. 5.2, str. 72, v době návratu z vrcholů 5 nebo 6 není označován vrchol 9.

V-4.2.10 (str. 60). Všechny vrcholy jsou (dokonce orientovaně) dostupné z kořene, proto je graf souvislý. Z vlastností vstupních stupňů plyne, že je-li n počet vrcholů, je počet hran $n - 1$. Podle věty 4.2.9 je tedy stromem.

V-4.2.12 (str. 60). Je-li souvislý a má-li alespoň tolik hran, kolik má vrcholů.

V-4.2.13 (str. 60). Dvě nebo tři.

V-4.2.14 (str. 61). K_4 má dvě neizomorfní kostry, K_5 má tři.

V-4.3.12 (str. 64). Kdyby ceny hran nebyly různé, mohla by přidáním všech hran splňujících podmínku (*) vzniknout v grafu L kružnice. Jednoduchým příkladem je graf, kde několik nejlevnějších hran má stejnou cenu a tvoří kružnici.

Úprava Borůvkova algoritmu spočívá v tom, že ceny hran učiníme různými. Lze to udělat buď zvětšením všech cen o velmi malá různá čísla, anebo lépe tak, že při porovnávání cen dvou hran v případě stejných cen rozhodneme podle pořadových čísel obou hran.

V-4.4.6 (str. 66). Cesty, které jsou vrcholově disjunktní, jsou i hranově disjunktní. Naopak to ovšem neplatí.

Rovnost vrcholového a hranového stupně souvislosti neplatí např. pro graf, který dostaneme z K_4 odstraněním jedné hrany.

V-5.2.10 (str. 77). V grafu na obr. 5.3 je 9 různých topologických uspořádání vrcholů: 123456879, 123546879, 123564879, 123568479, 124356879, 132456879, 132546879, 132564879, 132568479, 123456879.

V-5.2.11 (str. 77). Je-li v_1, v_2, \dots, v_n topologické uspořádání vrcholů, pak topologické uspořádání hran získáme tak, že nejprve zařadíme všechny hrany z množiny $E^+(v_1)$, pak všechny hrany z $E^+(v_2)$ atd.

Je-li e_1, e_2, \dots, e_m topologické uspořádání hran, můžeme topologické uspořádání vrcholů získat tak, že nejprve zařadíme vrcholy s kladným výstupním stupněm, a to v pořadí, v němž se vyskytují poprvé v posloupnosti $Pv(e_1), Pv(e_2), \dots, Pv(e_m)$ a na konec zařadíme v libovolném pořadí vrcholy s výstupním stupněm nulovým.

V-5.3

splňují

4 =

8 =

souvisl

9 =

V-5.5

Jádrem

V-6.3

a: $\frac{v_1}{v_2}$ $\frac{v_2}{v_3}$

0

b: $\frac{v_1}{v_2}$ $\frac{v_2}{v_3}$

0

V-6.5

V-6.7

V-6.7

a výkor

V-7.3

Výc

také ho

do sou

polookr

sledů z

V-5.3.5 (str. 79). Tvrzení 8 a 9 přidat lze, viz následující důkaz. Tvrzení 10 splňují i grafy, které mají více než $n - 1$ hran, proto tvrzení 10 přidat nelze.

$4 \Rightarrow 8$: z existence kořene plyne souvislost.

$8 \Rightarrow 9$: z vlastností stupňů vrcholů plyne, že graf má přesně $n - 1$ hran. Ze souvislosti a počtu hran plyne, že G je stromem, a proto neobsahuje kružnici.

$9 \Rightarrow 5$: neobsahuje-li graf kružnici, neobsahuje ani cyklus.

V-5.5.5 (str. 84). Graf hry je podobný jako na obrázku 5.7, ale bez vrcholu 0. Jádrem je množina $\{1, 5, 9, 13\}$.

V-6.3.9 (str. 98).

a: vrchol	1	2	3	4	5	6	7	8	9	10
vzdálenost	0	14	6	10	22	28	31	42	30	35
ODKUD	-	3-2	1-3	3-4	4-5	5-6	6-7	7-8	5-9	6-10
							4-7			
b: vrchol	1	2	3	4	5	6	7	8	9	
vzdálenost	0	2	2	6	1	0	9	4	6	
ODKUD	-	1-2	2-3	2-4	3-5	5-6	5-7	6-8	8-9	
						3-6				

V-6.5.3 (str. 101). Příkladem je graf z obr. 6.3, str. 97.

V-6.7.5 (str. 106). Výsledná matice vzdáleností:

$$\begin{pmatrix} 0 & 2 & 8 & 4 & 4 \\ 8 & 0 & 6 & 2 & 2 \\ 3 & -5 & 0 & -3 & -3 \\ 7 & -1 & 4 & 0 & 1 \\ 6 & -2 & 4 & 0 & 0 \end{pmatrix}.$$

V-6.7.6 (str. 106). Matici Q inicializujeme příkazem $Q(i, j) := j$ pro všechna i, j a výkonný příkaz uvnitř cyklů nahradíme příkazem:

když $M(i, j) > M(i, k) + M(k, j)$,

pak proved' $M(i, j) := M(i, k) + M(k, j)$ a navíc $Q(i, j) := Q(i, k)$.

V-7.3.5 (str. 121). Oba algoritmy začínají s rozdílně definovanými maticemi.

Výchozí matice A ve Floydově algoritmu 6.7.3 zahrnuje kromě hodnot hran také hodnoty triviálních sledů (nuly na diagonále), neboť jejich opakované zahrnutí do součtů \oplus nevadí. Kleeneho algoritmus však musí kvůli obecným uzavřeným polookruhům započítat triviální sledy jen jednou, proto jsou hodnoty triviálních sledů z výchozí matice vyloučeny a jsou započítány až na konci výpočtu.

V-7.3.7 (str. 121). Matice šířek nejširších sledů:

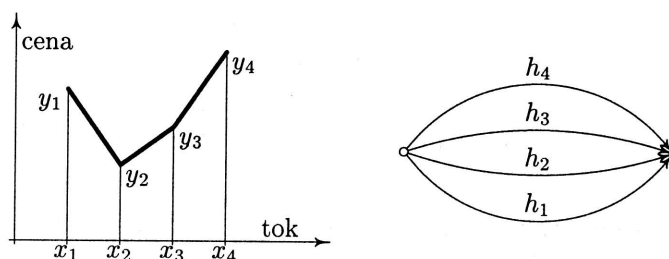
$$\begin{pmatrix} \infty & 6 & 7 & 7 & 7 & 7 \\ 4 & \infty & 5 & 4 & 4 & 4 \\ 4 & 6 & \infty & 4 & 4 & 4 \\ 4 & 6 & 8 & \infty & 4 & 4 \\ 4 & 6 & 7 & 7 & \infty & 7 \\ 4 & 6 & 8 & 8 & 4 & \infty \end{pmatrix}.$$

V-7.4.7 (str. 125). $s(v_1, v_1) = 1$, $s(v_1, v_2) = 3\frac{1}{4}$, $s(v_1, v_3) = 3\frac{1}{2}$, $s(v_1, v_4) = 1$.
 $s(v_1, v_5) = \frac{7}{8}$. $s(v_1, v_6) = \frac{1}{8}$.

V-8.1.14 (str. 133). Záporný tok v hraně incidentní s v by nemusel procházet hranou e_v a omezení toku v hraně e_v by bylo neúčinné.

V-8.1.17 (str. 133). Je-li např. tok v hraně h omezen na interval $\langle x_1, x_4 \rangle$ a je-li závislost ceny toku na jeho velikosti dána grafem na obr. 15.4, nahradíme hranu h čtveřicí rovnoběžných hran h_1, \dots, h_4 takto:

e	$l(e)$	$c(e)$	$a(e)$
h_1	x_1	x_1	y_1/x_1
h_2	0	$x_2 - x_1$	$(y_2 - y_1)/(x_2 - x_1)$
h_3	0	$x_3 - x_2$	$(y_3 - y_2)/(x_3 - x_2)$
h_4	0	$x_4 - x_3$	$(y_4 - y_3)/(x_4 - x_3)$



Obrázek 15.4: Konvexní závislost ceny toku na velikosti toku.

V-8.2.10 (str. 139). Kapacity hran budou jednotkové, dolní omezení nulové. Ke grafu přidáme návratovou hranu z cílového vrcholu c do výchozího vrcholu r a v této jediné hraně bude dolní i horní omezení toku jednotkové. Nejkratší cestu budou tvořit původní hrany s jednotkovým tokem.

Cyklus se zápornou délkou může způsobit chybný výsledek, neboť jeho hrany mohou být nasyceny tokem, a tím mohou být nepoužitelné pro tok, který má určit nejkratší cestu.

V-8.4.10 (str. 149). Pokud přípustný tok najdeme, bylo omezení stanoveno dobře. (To platí pro hledání přípustného toku. Pro hledání toku maximálního by omezení mohla být i takto příliš silná.)

Pokud přípustná cirkulace neexistuje a návratová hrana je součástí řezu se zápornou kapacitou, pak je třeba omezení v návratové hraně změnit (uvolnit). Pokud přípustná cirkulace neexistuje, ale řez se zápornou kapacitou návratovou hranu neobsahuje, pak omezení v návratové hraně bylo stanoveno dobře, ale přípustný tok v původním grafu neexistuje.

V-9.1.5 (str. 162). Doprostřed hromady hnoje umístíme fiktivní stromek.

V-9.1.8 (str. 162). Nechť je dána úloha o nejlevnějším maximálním párování. Zvolíme nějaké dosti velké kladné číslo M , např. větší než součet cen všech hran, a pak pro každou hranu e její cenu $c(e)$ nahradíme cenou $M - c(e)$. V grafu s takto změněnými cenami pak řešíme úlohu o nejdražším párování.

Označme m počet hran v maximálním párování. Výsledné párování P , které je nejdražší vůči novým cenám, musí obsahovat také m hran, neboť jinak by kterékoli maximální párování mělo lepší (totiž vyšší) cenu. Párování P je ovšem nejdražší i mezi všemi maximálními párováními a jeho cena je $mM - \sum_{e \in P} c(e)$. Párování P tedy má ze všech maximálních párování nejmenší hodnotu $\sum_{e \in P} c(e)$.

V-9.3.6 (str. 171). Kdyby bylo $|X| > |Y|$, platilo by $\sum_{x \in X} d(x) > \sum_{y \in Y} d(y)$, což není možné, neboť součty stupňů na obou stranách grafu musí být stejné. Rovnost $|X| = |Y|$ je možná pouze tak, že všechny vrcholy mají stejný stupeň.

V-9.3.7 (str. 171). Předpoklady věty 9.3.5 jsou splněny, existuje tedy párování, které nasycuje jednu z obou množin. Graf je však regulární (všechny vrcholy mají stejný stupeň), proto $|X| = |Y|$. Zmíněné párování je tedy perfektní.

V-10.1.6 (str. 179). Pro $k = 3$ a $n = 2$ jsou de Bruijnovy posloupnosti dvě: 00011101 a 00010111. (Posloupnost, kterou získáme cyklickou záměnou, pokládáme za tutéž cyklickou posloupnost.)

Pro $k = 2$ a $n = 3$ má de Bruijnova posloupnost délku 9 a těchto posloupností je 22.

V-10.2.4 (str. 180). Graf splňuje předpoklady věty 10.2.2, existuje v něm tedy uzavřený orientovaný tah, který prochází všemi hranami. Díky souvislosti tento tah prochází i všemi vrcholy. Každé dva vrcholy tedy jsou spojeny orientovaným sledem.

V-10.2.8 (str. 181). Oba vrcholy lichého stupně leží v téže komponentě souvislosti. Pro každou komponentu souvislosti tedy bude stačit jeden tah, jeden z těchto tahů bude neuzavřený.

Kdyby počet vrcholů lichého stupně byl 4, byly by nutné 3 nebo 4 tahy v závislosti na tom, zda všechny vrcholy lichého stupně leží v téže komponentě, nebo dva a dva v různých komponentách.

V-10.2.10 (str. 182). Označme $K = \sum_{v \in V} |R(v)|$. Je-li $K = 0$, pokrývá hrany grafu jeden uzavřený (a eulerovský) orientovaný tah. Je-li $K > 0$, je k pokrytí hran třeba $K/2$ neuzavřených orientovaných tahů.

DŮKAZ: Je-li $K > 0$, označme k^+ součet všech kladných a k^- součet všech záporných hodnot R . Poněvadž $\sum_{v \in V} R(v) = 0$, platí $k^+ = -k^- = K/2$. Můžeme tedy ke grafu přidat celkem k^+ hran, které vedou z vrcholů se záporným R do vrcholů s kladným R , a to tak, aby vznikl graf G' , v němž každý vrchol má stejný vstupní i výstupní stupeň. V grafu G' existuje orientovaný uzavřený eulerovský tah. Odstraněním přidaných hran se tento tah rozpadne na k^+ neuzavřených tahů, které pokrývají hrany původního grafu.

Méně než $K/2$ tahů k pokrytí hran nestačí, neboť přidáním menšího počtu hran nelze získat graf, v němž by existoval orientovaný uzavřený eulerovský tah. \square

V-10.3.4 (str. 184). Každou hranu, kterou sled prochází k -krát, nahradme k kopiemi. Kdyby pro některou hranu bylo $k > 2$, bylo by možno dvě z těchto hran odebrat. Sudost stupňů by tím zůstala zachována, ale eulerovský tah by se zkrátil.

V-10.3.5 (str. 184). Kdyby některé dvě cesty měly společnou hranu e , párování P by nebylo nejlevnější. Vynecháním hrany e z obou cest se tyto cesty rozpadnou na části, z nichž lze sestavit jiné dvě cesty o menší celkové délce. Viz obr. 15.5.



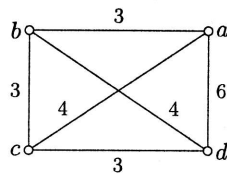
Obrázek 15.5: Náhrada dvou cest se společnou hranou e dvěma hranově disjunktivními cestami.

V-10.3.7 (str. 184). Je-li v grafu hrana se zápornou délkou, úloha ztrácí smysl. Opakovaným procházením takovou hranou sem a tam lze totiž dosáhnout libovolně malé (záporné) délky sledu. Žádný sled pak není nejkratší, vždy existuje sled ještě kratší.

V-12.2.7 (str. 201). Jsou-li fixovány vrcholy x, y , pak prodloužíme o dostatečně velkou hodnotu M všechny hrany z množiny $E(x) \cup E(y)$. Hamiltonovské cesty, které začínají ve vrcholu x a končí ve vrcholu y (nebo naopak), se tím prodlouží o $2M$, zatímco všechny ostatní hamiltonovské cesty se prodlouží o $3M$ nebo o $4M$. Pokud v daném grafu existuje vůbec nějaká hamiltonovská cesta spojující x a y , pak nejkratší hamiltonovská cesta v grafu s prodlouženými hranami bude spojuvat právě vrcholy x a y .

V-12.2.8 (str. 201). Stačí, aby M bylo větší než délka optimální cesty. Poněvadž ji předem neznáme, lze zvolit např. délku nejdelší hrany vynásobenou počtem vrcholů.

V-12.2.9 (str. 202). Viz obr. 15.6.



Obrázek 15.6: Příklad, že nejkratší hamiltonovská cesta nemusí být obsažena v nejkratší hamiltonovské kružnici.

V-12.3.1 (str. 202). Příkladem je graf $K_{2,3}$.

V-12.3.4 (str. 203). V bipartitním grafu musí hamiltonovská kružnice přecházet střídavě mezi oběma stranami grafu, obě strany by tedy musely mít stejný počet vrcholů.

V-12.4.3 (str. 205). V prvních třech podúlohách sice dostaneme stejné minimální kostry jako v 12.4.2, ale s podstatně silnějšími přidanými omezeními, což urychlí následující výpočet. Průběh výpočtu může být tento:

podúloha	zakázané hrany	vynucené hrany	odhad	komentář
1	-	-	30	větveno na 2, 3, 4
2	(1, 3)	-	38	umrtveno odhadem
3	(1, 4)	(1, 3)	37	umrtveno odhadem
4	(1, 2), (1, 5), (1, 6)	(1, 3), (1, 4)	33	větveno na 5, 6, 7
5	(1, 2), (1, 5), (1, 6), (2, 5)	(1, 3), (1, 4)	38	hamilt. cesta
6	(1, 2), (1, 5), (1, 6), (2, 3)	(1, 3), (1, 4), (2, 5)	36	umrtveno odhadem
7	(1, 2), (1, 5), (1, 6), (2, 4), (2, 6)	(1, 3), (1, 4), (2, 5), (2, 3)	36	optimální řešení

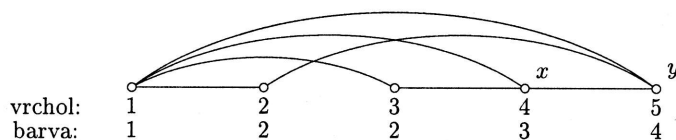
V-12.4.7 (str. 210). Délky všech hamiltonovských cest z x do y se penalizací zvýší o hodnotu $(2 \sum_{i=1}^n p(i)) - p(x) - p(y)$. Jestliže jsme při použití penále p dostali nejlevnější 1-strom o ceně C , lze jako dolní odhad použít číslo $C + p(x) + p(y) - 2 \sum_{i=1}^n p(i)$.

V-13.1.2 (str. 211). V obarvení podle obrázku 13.1 stačí barvu D nahradit barvou A . Graf, který má smyčku, nelze obarvit.

V-13.1.12 (str. 215). Strom o alespoň dvou vrcholech je vždy dvoubarevný.

V-13.2.4 (str. 218). Nejrychlejší výpočet bude při uspořádání vrcholů vzestupně podle jejich hloubky, tj. kořen jako první a vrchol v hloubce 2 jako poslední. Naopak výpočet bude nejpomalejší, když všechny listy budou na začátku a kořen na konci.

V-13.2.6 (str. 219). Vrcholy mezi nejvyšším a druhým nejvyšším v $P(y)$ mohou ovlivnit barvu vrcholu $x = \max(P(y))$ a tím i barvu vrcholu y . Viz graf na obr. 15.7, kde „vylepšený“ algoritmus přehledně možnost zvýšit barvu vrcholu 3 a v důsledku toho nenajde optimální obarvení.

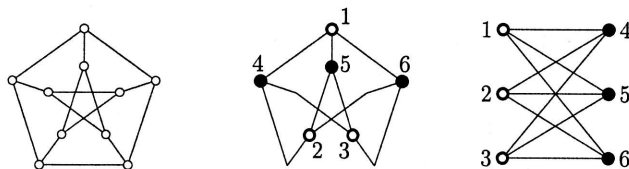


Obrázek 15.7: Protipříklad ke cvičení 13.2.6.

V-14.2.2 (str. 227). Krychle je duální k pravidelnému osmistěnu, pravidelný dvanáctistěn je duální k pravidelnému dvacetistěnu. Čtyřstěn je duální sám k sobě.

V-14.3.2 (str. 229). Důkaz je podobný důkazu věty 14.3.1, ale namísto kostry použijeme napnutý les (viz 4.2.4, str. 58, a 4.2.8, str. 59).

V-14.4.3 (str. 230). Viz obr. 15.8.



Obrázek 15.8: Petersenův graf je homeomorfní s $K_{3,3}$.

V-15.4.3 (str. 242). Rozšířená matice soustavy rovnic pro smyčkové proudy má tvar

$$\left(\begin{array}{cccc|c} 21 & -10 & 0 & 10 & 6 \\ -10 & 31 & -10 & 1 & 6 \\ 0 & -10 & 21 & 10 & 6 \\ 10 & 1 & 10 & 22 & 18 \end{array} \right).$$

Její řešení dostaneme $I_1 = 2,56468 \text{ A}$, $I_4 = 1,56498 \text{ A}$, $I_7 = 2,56468 \text{ A}$, $I_9 = -3,22084 \text{ A}$ a dále pak $I_2 = 0,65616 \text{ A}$, $I_3 = 0,99970 \text{ A}$, $I_5 = 1,65596 \text{ A}$, $I_6 = 0,99970 \text{ A}$, $I_8 = 0,65616 \text{ A}$.

Lit

[AHU74]

[Berge58]

[Berge73]

[Demel91]

[Demel88]

[FF62]

[GJ79]

[Harary6]

[Chr75]

[Chytil82]

[KŠCh89]

[Kučera8]

[Lawler76]

[MN96]

nahradit

arevný.

vzestupně
ní. Naopak
n na konci.P(y) mohou
a obr. 15.7,
v důsledkupravidelný
ám k sobě.

isto kostry

proudů má

68 A, $I_9 =$ 96 A, $I_6 =$

Literatura

- [AHU74] Aho, A. V., Hopcroft, J. E., Ullman, J. D.: *The Design and Analysis of Computer Algorithms*. Reading, Addison-Wesley, 1974.
- [Berge58] Berge, C.: *Théorie des Graphes et ses Applications*. Paris, Gautier-Villars, 1958.
- [Berge73] Berge, C.: *Graphs and Hypergraphs*. Amsterdam, North-Holland Publ. Co., 1973.
- [Demel91] Demel, J.: *Teorie grafů*. Praha, skripta ČVUT, 1982, 2. vydání 1991.
- [Demel88] Demel, J.: *Grafy*. Praha, SNTL, 1988.
- [FF62] Ford, L. R. Jr., Fulkerson, D. R.: *Flows in Networks*. Princeton, Princeton University Press, 1962.
- [GJ79] Garrey, M. R., Johnson, D. S.: *Computers and Intractability: a Guide to the Theory of NP-Completeness*. San Francisco, W. H. Freeman, 1979.
- [Harary69] Harary, F.: *Graph Theory*. Reading, Addison-Wesley, 1969.
- [Chr75] Christofides, N.: *Graph theory – An Algorithmic Approach*. New York, Academic Press, 1975.
- [Chytil82] Chytil, M.: *Automaty a gramatiky*. Praha, SNTL, 1982.
- [KŠCh89] Kolář, J., Štěpánková, O., Chytil, M.: *Logika, algebry a grafy*. Praha, SNTL, 1989.
- [Kučera83] Kučera, L.: *Kombinatorické algoritmy*. Praha, SNTL, 1983.
- [Lawler76] Lawler, E., L.: *Combinatorial Optimization – Networks and Matroids*. New York, Holt, Reinhart and Winston, Inc., 1976.
- [MN96] Matoušek, J., Nešetřil, J.: *Kapitoly z diskrétní matematiky*. Praha, Matfyzpress, 1996.

- [Nečas78] Nečas, J.: *Grafy a jejich použití*. Praha, SNTL, 1978.
- [Nešetřil79] Nešetřil, J.: *Teorie grafů*. Praha, SNTL, 1979.
- [Plesník83] Plesník, J.: *Grafové algoritmy*. Bratislava, VEDA, 1983.
- [Sedláček64] Sedláček, J.: *Kombinatorika v teorii a praxi*. Praha, NČSAV, 1964, 2. vydání, 1977.
- [Sedláček81] Sedláček, J.: *Úvod do teorie grafů*. Praha, Academia, 1981.
- [SwTh81] Swamy, M. N. S., Thulasiraman, K.: *Graphs, Networks and Algorithms*. New York, John Wiley & Sons, Inc., 1981.
- [Šišma97] Šišma, P.: *Teorie grafů 1736–1963*. Praha, Prometheus, 1997.
- [Töpfer95] Töpfer, P.: *Algoritmy a programovací techniky*. Praha, Prometheus, 1995.

Re

1-strom

algorit

Bo

Di

Fl

Fo

he

hla

Ja

Kl

ma

mo

Ou

ov

po

Ta

artikula

automa

barevno

hra

vro

Bellman

centrum

certifiká

cesta, 2

alte

har

krit

stří

zlep

cirkulac

Rejstřík

- 1-strom, 208
- algoritmus
- Borůvkův, 63
 - Dijkstrův, 101
 - Floydův, 105
 - Fordův-Fulkersonův, 141
 - heuristický, 189
 - hladový, 62, 196
 - Jarníkův-Primův, 63
 - Kleeneho, 119
 - maďarský, 171, 172
 - modifikovaný Dijkstrův, 95
 - Out of Kilter, 154
 - ověřovací, 187
 - polynomiální, 31, 187
 - Tarjanův, 69
- artikulace, 66
- automat, 120
- barevnost
- hranová, 215
 - vrcholová, 211
- Bellmanovy rovnice, 88
- centrum grafu, 46
- certifikát, 187
- cesta, 20
- alternující, 163
 - hamiltonovská, 197
 - kritická, 44
 - střídavá, 163
 - zlepšující, 139, 140
- cirkulace, 129
- elementární, 232
- cyklus, 20
- číslo cyklomatické, 233
- de Bruijn, 178
- defekt, 147, 155
- délka sledu, 20
- diagram Hasseův, 35
- dostupnost, 20, 58, 69
- dualita
- lineárního programování, 154
 - rovinných grafů, 226
- dvanáctistěn, 226
- dynamické programování, 88
- ekvivalence, 34
- nakreslení, 225
- eliminace
- smyčky, 123
 - vrcholu, 123
- Euler, 177, 229
- faktor, 18
- formule Eulerova, 229
- fronta, 50, 95
- fundamentální soustava
- kružnic, 232
 - řezů, 236
- funkce účelová, 32
- graf, 11
- acyklický, 73
 - bipartitní, 21, 52, 214

- diskrétní, 21
 doplňkový, 212
 duální, 226
 hranový, 215
 ohodnocený, 14
 orientovaný, 11
 Petersenův, 230
 planární, 12, 223
 prostý, 16
 regulární, 21
 rovinný, 12, 145, 223
 topologický, 223
 rovnosti, 172
 řídký, 95, 229
 síťový, 42
 symetrický, 14
 úplný, 21
- halda, 96
 hledání
 do hloubky, 52, 69, 76
 do šířky, 49, 144
 značkováním, 47, 141
 hodnota grafu, 238
 homeomorfní grafy, 230
 hra dvou hráčů, 83
 hrana, 11
 návratová, 129, 135, 149
 vpřed, vzad, 140, 231
 hranová barevnost, 215
 hrany rovnoběžné, 11, 12
- chromatický index, 215
- incidence, 11, 224
 inorder, 23, 56
 instance úlohy, 32, 185
 izomorfismus, 17
- jádro grafu, 83
- kapacita
 řezu, 131, 139, 142, 146, 228
 zlepšující cesty, 140
- kilter, 154
 Kirchhoff, 129, 234
- Kleene, 119, 120
 klika, 212
 kolmost vektorů, 238
 komponenta
 silné souvislosti, 67
 souvislosti, 57
 kořen grafu, 77
 kořenový strom, 21
 kostra, 58
 minimální, 61, 204
 kružnice, 20
 střídavá, 163
 zlepšující, 139
 květ, 167
- les, 58
 napnutý, 58, 232–242
 lineární programování, 154
- markovský řetězec, 116
 Masonovo pravidlo, 128
- matice
 délek hran, 86
 dostupnosti, 69, 114
 incidenční, 25, 237
 sousednosti, 24, 114
 vzdáleností, 86
- metoda
 CPM, 44
 hrubé síly, 189, 190, 203
 větví a mezí, 191
- množina, 32
 nezávislá, 212, 219
- most, 65
 mosty města Královce, 177
 multigraf, 16
- nakreslení grafu, 223
 napnutý les, 58, 232–242
 následník, 15
 násobnost hrany, 16
 nekonečno ∞ , 28, 93, 113
 nezávislá množina, 212, 219
 NP-úloha, 187
- obarvení, 211

- očíslování topologické, 74
odhad
 doby práce algoritmu, 30, 186
 v backtrackingu, 190
 v metodě větvi a mezí, 192
okolí vrcholu, 15
operace uzávěru, 112
optimalizace, 31
orientovaný graf, 11
Out of Kilter, 154
- párování, 161
penalizace, 208
Platonova tělesa, 226
podgraf, 18
podmínka omezující, 31
podúloha, 191
polookruh, 110
posloupnost de Bruijnova, 178
postorder, 23, 56
potenciál, 154, 234
preorder, 23, 56
princip optimality Bellmanův, 88
problém obchodního cestujícího, 198
proces náhodný, 116
prostor řešení, 31
proud smyčkový, 241
průzkum lokální, 196
předchůdce, 15
převody úloh, 188, 199
přípustné řešení, 31
- redukce
 polynomiální, 188
 tranzitivní, 79
rekurzivní procedura, 55
relace, 33
relaxace podmínky, 192
rozklad, 35
- řešení přípustné, 31, 187, 190, 191
řetězec markovský, 116
řez, 16, 131, 146, 235
 minimální, 131, 139, 142, 228
- síť, 14
- transportní, 130, 148
 vrstvená, 151
- sled, 19
složitost, 185
smyčka, 11
součin kartézský, 33
soused, 15
souvislost, 57
 silná, 67
spotřebič, 129
stav, 37
 absorbující, 117
 hry, 83
 markovského řetězce, 116
stěna, 224
stereografická projekce, 224
strom, 58
 kořenový, 21, 77
 napnutý, 58
 řešení, 189, 192
 střídavý, 166
 uspořádaný, 22
- stupeň
 souvislosti, 65, 134, 225, 226
 stěny, 224
 vrcholu, 16
- tah, 20
 eulerovský, 177
- tok, 129, 231
 maximální, 131
 nejlevnější, 131, 152, 207
 od zdroje ke spotřebiči, 129
 přípustný, 130
- topologické uspořádání, 74, 83, 98
topologický rovinný graf, 223
tranzitivita, 34, 79
trojúhelník, 20
typ úlohy, 32, 185
- účelová funkce, 32
úloha, 32, 185
 čínského pošťáka, 182
 dopravní, 134
 NP-těžká, 31, 185

- NP-úplná, 188
 optimalizační, 31, 187, 191
 přiřazovací, 135, 171, 206
 rozhodovací, 187
 třídy NP, 187
 třídy P, 187
 uspořádání, 35
 cyklické, 225
 topologické, 74, 83, 98
 uzávěr, 112
 tranzitivní, 79, 114
 velikost toku, 130
- věta
 Fordova-Fulkersonova, 142
 Hallova, 170
 Königova, 169
 Kuratowského, 230
 Mengerova, 66, 134
 vrchol, 11
- zákon Kirchhoffův, 129, 234
 zásobník, 53, 70–73
 zdroj, 129
 značkování vrcholů, 47, 141
 zobrazení, 36

P
 R,
 |M
 A
 n
 m
 ∞
 O(
 V(
 E(
 P
 K
 V(
 V+
 V-
 E(
 d(u
 W
 m(
 G ≅
 K_n
 K_m
 D+
 G+
 G*
 ⊕,
 ω(C
 α(C
 χ(C
 χ'(C
 -G
 G^D

Přehled značení

\mathbb{R}, \mathbb{N}	= množina reálných a přirozených čísel (včetně nuly)	1.12.1, str. 32
$ M $	= počet prvků množiny M	1.12.1, str. 33
$A \setminus B$	= rozdíl množin A a B	1.12.1, str. 33
n	= obvykle počet vrcholů grafu	1.10.6, str. 31
m	= obvykle počet hran grafu	1.10.6, str. 31
∞	= nekonečno	6.2.10, str. 93
$O(f)$	= asymptotický odhad funkce f shora	1.10.6, str. 30
$V(G)$	= množina vrcholů grafu G	1.3, str. 15
$E(G)$	= množina hran grafu G	1.3, str. 15
$Pv(e)$	= počáteční vrchol hrany e	1.1.2, str. 11
$Kv(e)$	= koncový vrchol hrany e	1.1.2, str. 11
$V(x)$	= množina sousedů vrcholu x	1.3, str. 15
$V^+(x)$	= množina následníků vrcholu x	1.3, str. 15
$V^-(x)$	= množina předchůdců vrcholu x	1.3, str. 15
$E(v), E^+(v), E^-(v)$	= okolí vrcholu	1.3, str. 15
$d(v), d^+(v), d^-(v)$	= stupeň vrcholu	1.3, str. 15
$W(A), W^+(A), W^-(A)$	= řez určený množinou A	1.3, str. 15
$m(v, w), m^+(v, w)$	= násobnost hrany	1.3, str. 15
$G \cong G'$	= izomorfismus grafů G a G'	1.4.2, str. 17
K_n	= úplný graf o n vrcholech	1.6, str. 21
$K_{m,n}$	= úplný bipartitní graf se stranami o m a n vrcholech	1.6, str. 21
$D^+(v)$	= množina vrcholů orientovaně dostupných z vrcholu v	5.1.10, str. 69
G^+	= tranzitivní uzávěr grafu G	5.4.1, str. 79
G^*	= reflexivní a tranzitivní uzávěr grafu G	5.4.1, str. 79
$\oplus, \otimes, *$	= operace v polookruhu	7.1.1, str. 110
$\omega(G)$	= klikovost grafu	13.1.4, str. 212
$\alpha(G)$	= nezávislost grafu	13.1.3, str. 212
$\chi(G)$	= vrcholová barevnost grafu	13.1.1, str. 211
$\chi'(G)$	= hranová barevnost grafu	13.1.15, str. 215
$-G$	= doplňkový graf	13.1.4, str. 212
G^D	= duální graf	14.2, str. 226