

Západočeská univerzita v Plzni

Fakulta aplikovaných věd

Katedra matematiky

Diplomová práce

**Algoritmy pro určení hranové souvislosti grafu a jejích
variant**

Bc. Tomáš Kraus

Vedoucí práce: doc. RNDr. Tomáš Kaiser, Ph.D.

Plzeň, 2016

Prohlašuji, že jsem svou diplomovou práci vypracoval samostatně pomocí uvedených zdrojů.

V Plzni dne

.....

Zde bych rád poděkoval vedoucímu diplomové práce doc. RNDr. Tomáši Kaiserovi, Ph.D. za odborné vedení, poskytnuté informace a čas, který věnoval mé diplomové práci.

Abstrakt

Tématem diplomové práce jsou algoritmy pro určení hranové souvislosti grafu. Diplomová práce obsahuje přehled vybraných algoritmů. V rámci diplomové práce byl implementován algoritmus pro určení cyklické hranové souvislosti grafu. Původními výsledky jsou algoritmus pro určení esenciální hranové souvislosti grafu a algoritmus pro určení esenciální hranové souvislosti kubického grafu.

Klíčová slova

Hranová souvislost grafu, esenciální hranová souvislost grafu, cyklická hranová souvislost grafu, algoritmus.

Abstract

The topic of thesis are algorithms for determining the edge connectivity of graphs. The thesis contains an overview of selected algorithms. In the thesis was implemented algorithm for determining cyclic edge connectivity of graphs. Original results are algorithm for determining essential edge connectivity of graphs and algorithm for determining essential edge connectivity of cubic graphs. The thesis is attached with a CD that carries program for determining cyclic edge connectivity of graphs.

Keywords

Edge connectivity of graphs, essential edge connectivity of graphs, cyclic edge connectivity of graphs, algorithm.

Obsah

1 Úvod	7
2 Základní pojmy	8
2.1 Obecné pojmy teorie grafů	8
2.2 Speciální pojmy teorie grafů k hranové souvislosti grafu	18
2.3 Speciální pojmy k implementaci algoritmů	27
3 Přehled algoritmů hranové souvislosti grafu	29
3.1 Algoritmus pro určení hranové souvislosti grafu s využitím toků	29
3.2 Algoritmus pro určení hranové souvislosti grafu pomocí platného uspořádání vrcholů	33
3.3 Algoritmus pro testování hranové 2-souvislosti grafu	35
3.4 Algoritmus pro testování hranové 3-souvislosti grafu	39
4 Algoritmus pro určení cyklické hranové souvislosti kubického grafu a jeho implementace	50
4.1 Algoritmus pro určení cyklické hranové souvislosti kubického grafu	50
4.2 Implementace algoritmu pro určení cyklické hranové souvislosti kubického grafu	58
5 Algoritmy pro určení esenciální hranové souvislosti grafu	60
5.1 Algoritmus pro určení esenciální hranové souvislosti grafu	60
5.2 Algoritmus pro určení esenciální hranové souvislosti kubického grafu	64
6 Závěr	68
Seznam použité literatury a zdrojů	69
Přílohy	70

1 Úvod

Tématem této diplomové práce jsou algoritmy pro určení hranové souvislosti grafu. Hranová souvislost je důležitým parametrem grafu a existuje několik různých algoritmů pro její určení. Dále existují různé varianty hranové souvislosti grafu a i pro jejich určení existují algoritmy. V této práci se proto věnujeme jen několika nejvýznamnějším algoritmům.

Algoritmy pro určení hranové souvislosti grafu mají uplatnění v mnoha oborech lidské činnosti, např. v informatice, biomechanice a medicíně.

V kapitole 2 připomeneme základní pojmy teorie grafů a uvedeme i první algoritmus, a to Fordův-Fulkersonův algoritmus pro nalezení maximálního toku v síti.

Přehled algoritmů pro určení hranové souvislosti grafu je podán v kapitole 3. Kapitola obsahuje popis několika vybraných algoritmů včetně matematických vět, na nichž jsou založeny. Především se jedná o algoritmus pro nalezení Gomoryho-Huova stromu, rychlý algoritmus pro určení hranové souvislosti grafu autorů Nagamochiho a Ibarakiho a algoritmus pro testování hranové 2-souvislosti a hranové 3-souvislosti grafu.

Praktickým cílem diplomové práce bylo implementovat algoritmus pro určení cyklické hranové souvislosti grafu. Popis tohoto algoritmu a výsledky implementace jsou v kapitole 4.

Původní výsledky této diplomové práce obsahuje kapitola 5, která je věnována tzv. esenciální hranové souvislosti grafu. Těžištěm kapitoly je popis dvou algoritmů pro určení esenciální souvislosti grafu. První algoritmus je určený pro všechny grafy a druhý speciálně pro kubické grafy.

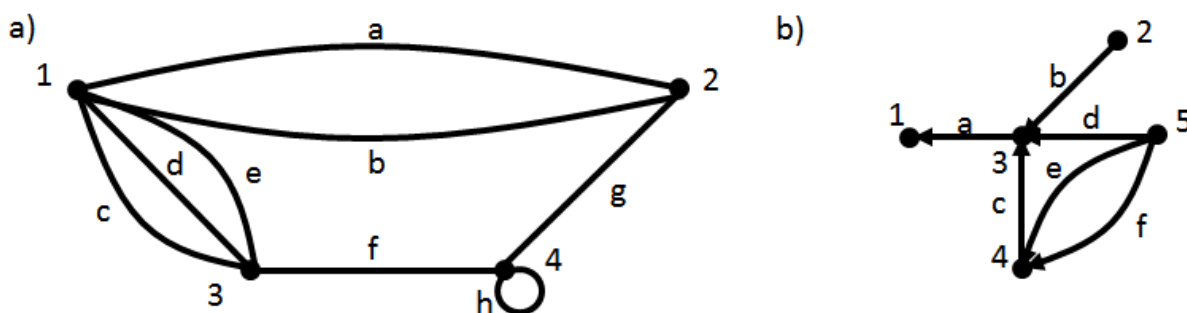
2 Základní pojmy

2.1 Obecné pojmy teorie grafů

Definice 2.1.1 [1]

Graf G (též neorientovaný graf G) je uspořádaná trojice $(V(G), E(G), \psi_G)$, kde $V(G)$ a $E(G)$ jsou navzájem disjunktní množiny a ψ_G je zobrazení, které přiřazuje každému prvku svého definičního oboru $E(G)$ jednoprvkovou nebo dvouprvkovou podmnožinu množiny $V(G)$.

Příklad grafu G je znázorněn na obr. 2.1.1a, kde $V(G) = \{1, 2, 3, 4\}$, $E(G) = \{a, b, c, d, e, f, g, h\}$, $\psi_G(a) = \{1, 2\}$, $\psi_G(b) = \{1, 2\}$, $\psi_G(c) = \{1, 3\}$, $\psi_G(d) = \{1, 3\}$, $\psi_G(e) = \{1, 3\}$, $\psi_G(f) = \{3, 4\}$, $\psi_G(g) = \{2, 4\}$ a $\psi_G(h) = \{4\}$.



Obr. 2.1.1: a) Neorientovaný graf, b) Orientovaný graf

Definice 2.1.2 [1]

Orientovaný graf D je uspořádaná trojice $(V(D), A(D), \psi_D)$, kde $V(D)$ a $A(D)$ jsou navzájem disjunktní množiny a ψ_D je zobrazení, které přiřazuje každému prvku svého definičního oboru $A(D)$ uspořádanou dvojici prvků množiny $V(D)$.

Příklad orientovaného grafu D je znázorněn na obr. 2.1.1b, kde $V(D) = \{1, 2, 3, 4, 5\}$, $A(D) = \{a, b, c, d, e, f\}$, $\psi_D(a) = (3, 1)$, $\psi_D(b) = (2, 3)$, $\psi_D(c) = (4, 3)$, $\psi_D(d) = (5, 3)$, $\psi_D(e) = (5, 4)$ a $\psi_D(f) = (5, 4)$.

Pojem graf bez přívlastku orientovaný či neorientovaný bude dále znamenat neorientovaný graf, pokud nebude výslovně uvedeno nebo z kontextu zřejmé, že se jedná o orientovaný graf. Označení výše uvedených množin a zobrazení může být u různých grafů různé, ale pokud nebude výslovně uvedeno nebo z kontextu zřejmé jiné

značení, budeme v tomto textu vždy používat značení z výše uvedených definic (tato úmluva platí i pro všechny následující definice).

Definice 2.1.3 [1]

Zobrazení ψ_G (neorientovaného nebo orientovaného) grafu G se nazývá incidenční zobrazení grafu G .

Zobrazení chápeme jako binární relaci, tj. $\psi_G = \{(a, \psi_G(a)); a \in E(G)\}$ pro neorientovaný i orientovaný graf G .

Definice 2.1.4 [1]

U neorientovaného i orientovaného grafu G se prvky množiny $V(G)$ nazývají vrcholy, velikost množiny $V(G)$ se nazývá řád grafu G a značí $v(G)$. Pokud je množina $V(G)$ prázdná, nazývá se graf G prázdný. Pokud je množina $V(G)$ jednoprvková, nazývá se graf G triviální. Pokud má množina $V(G)$ alespoň dva prvky, nazývá se graf G netriviální. Graf G obsahující vrcholy x a y se též značí $G(x, y)$.

Definice 2.1.5 [1]

U neorientovaného grafu G se prvky množiny $E(G)$ nazývají hrany, velikost množiny $E(G)$ se nazývá velikost grafu G a značí $e(G)$. Pokud je v incidenčním zobrazení grafu G obrazem hrany e jednoprvková množina, nazývá se hrana e smyčka. U orientovaného grafu D se prvky množiny $A(D)$ nazývají orientované hrany, velikost množiny $A(D)$ se nazývá velikost (orientovaného) grafu D a značí $a(D)$. Pokud je v incidenčním zobrazení grafu D obrazem hrany a dvojice stejných prvků, nazývá se hrana a smyčka (též orientovaná smyčka). Každé dvě navzájem různé hrany a, b orientovaného či neorientovaného grafu H , které mají stejný obraz v incidenčním zobrazení grafu H , se nazývají (navzájem) paralelní hrany (u orientovaného grafu H též orientované (navzájem) paralelní hrany). Graf H takový, že $E(H)$ neobsahuje smyčky ani paralelní hrany, se nazývá prostý graf.

Graf znázorněný na obr. 2.1.1a má trojici navzájem paralelních hran c, d a e , dvojici navzájem paralelních hran a a b a smyčku h . Orientovaný graf znázorněný na obr. 2.1.1b má dvojici navzájem paralelních orientovaných hran e a f a nemá žádnou smyčku.

Definice 2.1.6 [1]

Pokud jsou pro neorientovaný graf G , resp. orientovaný graf D , obě množiny $V(G)$ a $E(G)$, resp. $V(D)$ a $A(D)$, konečné, nazývá se graf G , resp. orientovaný graf D , konečný. Pokud je alespoň jedna z množin $V(G)$, $E(G)$, resp. $V(D)$, $A(D)$, nekonečná, nazývá se graf G , resp. orientovaný graf D , nekonečný.

Definice 2.1.7 [1]

Pokud pro hranu e a vrcholy u a v grafu G platí $\psi_G(e) = \{u, v\}$, říkáme, že hrana e spojuje vrcholy u a v , které pak nazýváme koncovými vrcholy hrany e . Říkáme také, že jsou vrcholy u a v incidentní s hranou e a naopak že je hrana e incidentní s vrcholem u a s vrcholem v .

Definice 2.1.8 [1]

Pokud pro hranu a a pro vrcholy u a v orientovaného grafu D platí $\psi_D(a) = (u, v)$, říkáme, že hrana a vede z vrcholu u do vrcholu v , přičemž vrchol u nazýváme počátečním vrcholem hrany a a vrchol v koncovým vrcholem hrany a . Říkáme také, že jsou vrcholy u a v incidentní s hranou a a naopak že je hrana a incidentní s vrcholem u a s vrcholem v .

Definice 2.1.9 [1]

Dva navzájem různé vrcholy u a v grafu G se nazývají (vzájemní) sousedé, pokud jsou oba incidentní se stejnou hranou e grafu G . Říkáme též, že je vrchol u sousedem vrcholu v a že je vrchol v sousedem vrcholu u nebo že jsou u a v vzájemně sousední v grafu G . Množina všech sousedů vrcholu v v grafu G se označuje $N_G(v)$.

V grafu G znázorněném na obr. 2.1.1a např. platí $N_G(2) = \{1, 4\}$.

Definice 2.1.10 [1]

Pokud vede v orientovaném grafu D hrana a z vrcholu u do vrcholu v , říkáme, že u je počáteční soused vrcholu v a v je koncový soused vrcholu u v grafu D . Množina všech počátečních sousedů vrcholu v v orientovaném grafu D se označuje $N_D^-(v)$, množina všech koncových sousedů vrcholu v v orientovaném grafu D se označuje $N_D^+(v)$.

V orientovaném grafu D znázorněném na obr. 2.1.1b např. platí $N_D^-(3) = \{2, 4, 5\}$ a $N_D^+(3) = \{1\}$.

Definice 2.1.11 [1]

Stupeň vrcholu v v neorientovaném či orientovaném grafu G , který značíme $d_G(v)$, je součet $|A| + |B|$, kde A je množina všech hran incidentních s vrcholem v v grafu G a B je množina všech smyček grafu G incidentních s vrcholem v . Vrchol stupně nula se nazývá izolovaný vrchol. Nejmenší stupeň vrcholu grafu G se značí $\delta(G)$ a největší stupeň vrcholu grafu G se značí $\Delta(G)$. Neprázdný graf G se nazývá k -regulární, jestliže je stupeň každého vrcholu grafu G roven k .

V grafu G znázorněném na obr. 2.1.1a např. platí $d_G(3) = 4$, $\delta(G) = 3$ a $\Delta(G) = 5$.

Definice 2.1.12 [1]

Vstupní stupeň vrcholu v orientovaného grafu D , který značíme $d_D^-(v)$, je počet všech hran grafu D , jejichž koncovým vrcholem je vrchol v . Vrchol s nulovým vstupním stupněm se nazývá zdroj. Výstupní stupeň vrcholu v grafu D , který značíme $d_D^+(v)$, je počet všech hran grafu D , jejichž počátečním vrcholem je vrchol v . Vrchol s nulovým výstupním stupněm se nazývá stok. Nejmenší vstupní, resp. výstupní, stupeň vrcholu grafu D se značí $\delta^-(D)$, resp. $\delta^+(D)$, a největší vstupní, resp. výstupní, stupeň vrcholu grafu D se značí $\Delta^-(D)$, resp. $\Delta^+(D)$.

V orientovaném grafu D znázorněném na obr. 2.1.1b např. platí $d_D^-(4) = 2$, $d_D^+(4) = 1$, $\delta^-(D) = 1$, $\delta^+(D) = 1$, $\Delta^-(D) = 3$ a $\Delta^+(D) = 3$.

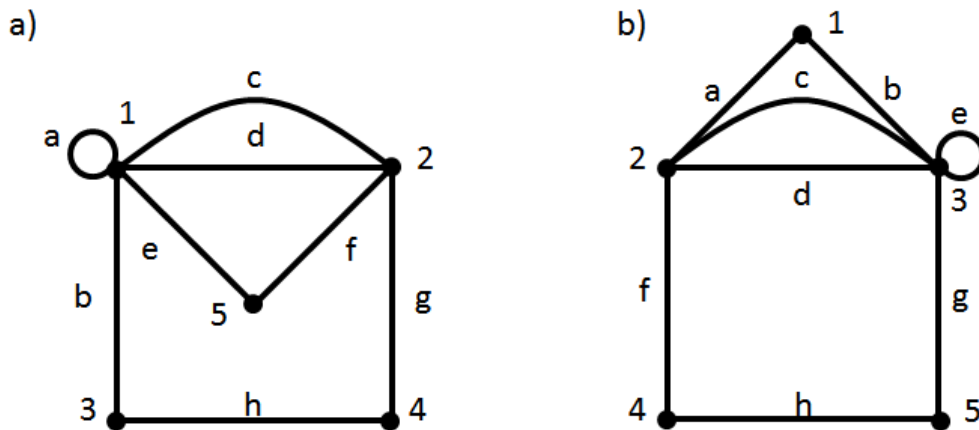
Definice 2.1.13 [1]

Dva grafy G a H , oba orientované či oba neorientované, jsou (navzájem) identické (píšeme $G = H$), jestliže $V(G) = V(H) \wedge E(G) = E(H) \wedge \psi_G = \psi_H$.

Definice 2.1.14 [1]

Dva grafy G a H , oba neorientované, resp. oba orientované, jsou (navzájem) izomorfní, píšeme $G \cong H$, jestliže existují bijekce $\theta: V(G) \rightarrow V(H)$ a $\phi: E(G) \rightarrow E(H)$ takové, že $\psi_G(e) = \{u, v\}$, resp. $\psi_G(e) = (u, v)$, právě když $\psi_H(\phi(e)) = \{\theta(u), \theta(v)\}$, resp. $\psi_H(\phi(e)) = (\theta(u), \theta(v))$. Dvojice (θ, ϕ) se nazývá izomorfismus.

Dva navzájem izomorfní grafy jsou tedy stejné až na jejich značení. Např. grafy znázorněné na obr. 2.1.2a a obr. 2.1.2b jsou navzájem izomorfní.



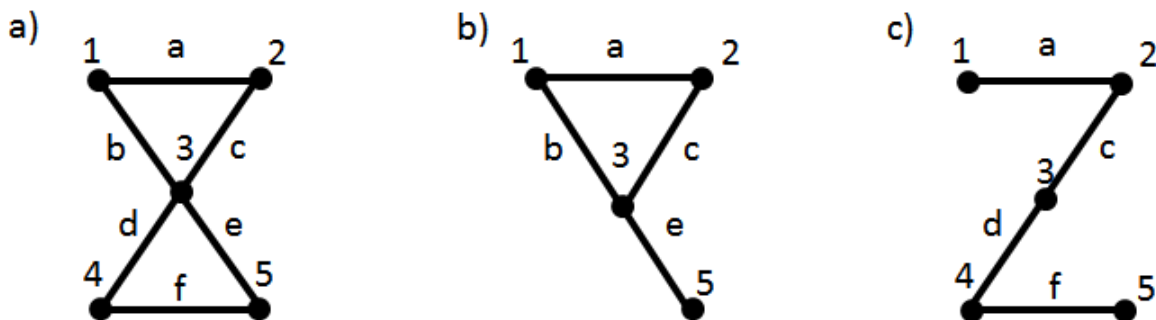
Obr. 2.1.1: a), b) Navzájem izomorfní grafy

Definice 2.1.15 [1]

Neorientovaný, resp. orientovaný, graf H je podgrafem neorientovaného, resp. orientovaného, grafu G , jestliže $V(H) \subseteq V(G) \wedge E(H) \subseteq E(G) \wedge \psi_H = \psi_G|E(H)$. Říkáme také, že je graf H obsažen v grafu G nebo že graf G obsahuje graf H , a píšeme $H \subseteq G$. Graf G se též nazývá nadgrafem grafu H .

Poznamenejme, že $\psi_G|E(H)$ je restrikce zobrazení ψ_G na množinu $E(H)$.

Na obr. 2.1.3b a 2.1.3c jsou znázorněny příklady podgrafů grafu znázorněného na obr. 2.1.3a.



Obr. 2.1.3: a) Graf G , b) Graf $G[\{1, 2, 3, 5\}]$, c) Faktor grafu G

Definice 2.1.16 [1]

Odstraněním vrcholu v v grafu G se rozumí změna grafu G v graf H , kde $V(H) = V(G) \setminus \{v\}$, $E(H) = \{e; e \in E(G) \wedge v \notin \psi_G(e)\}$ a $\psi_H = \psi_G|E(H)$. Graf H označujeme $G - v$. Pokud se v grafu G postupně odstraní všechny vrcholy z množiny S , $S \subseteq V(G)$, označuje se výsledný graf též $G - S$. Odstraněním hrany e grafu G se rozumí změna grafu G v graf K , kde $V(K) = V(G)$, $E(K) = E(G) \setminus \{e\}$ a $\psi_K = \psi_G|E(K)$. Graf K

označujeme $G \setminus e$. Pokud se v grafu G postupně odstraní všechny hrany z množiny T , $T \subseteq E(G)$, označuje se výsledný graf též $G \setminus T$.

Graf znázorněný na obr. 2.1.3b je podgraf $G - 4$ grafu G znázorněného na obr. 2.1.3a. Graf znázorněný na obr. 2.1.3c je podgraf $G \setminus \{b, e\}$ grafu G znázorněného na obr. 2.1.3a.

Definice 2.1.17 [1]

Odstraněním vrcholu v v orientovaném grafu D se rozumí změna grafu D v graf H , kde $V(H) = V(D) \setminus \{v\}$, $E(H) = \{e; e \in E(D) \wedge (\forall x \in V(D): e \neq (v, x) \wedge e \neq (x, v))\}$ a $\psi_H = \psi_D|E(H)$. Graf H označujeme $D - v$. Pokud se v grafu D postupně odstraní všechny vrcholy z množiny S , $S \subseteq V(D)$, označuje se výsledný graf též $D - S$. Odstraněním hrany a grafu D se rozumí změna grafu D v graf K , kde $V(K) = V(D)$, $E(K) = E(D) \setminus \{e\}$ a $\psi_K = \psi_D|E(K)$. Graf K označujeme $D \setminus e$. Pokud se v grafu D postupně odstraní všechny hrany z množiny T , $T \subseteq A(D)$, označuje se výsledný graf též $D \setminus T$.

Definice 2.1.18 [1]

Podgraf H neorientovaného či orientovaného grafu G , který lze vytvořit z grafu G pouze pomocí operace odstranění vrcholu, se nazývá indukovaný podgraf H grafu G na množině $V(H)$ a značí se $G[V(H)]$.

Graf znázorněný na obr. 2.1.3b je indukovaný podgraf $G[\{1, 2, 3, 5\}]$ grafu G znázorněného na obr. 2.1.3a. Indukovaným podgrafem grafu G je např. i graf G (operace odstranění vrcholu nepoužita) a prázdný graf (operace odstranění vrcholu použita $v(G)$ -krát).

Definice 2.1.19 [1]

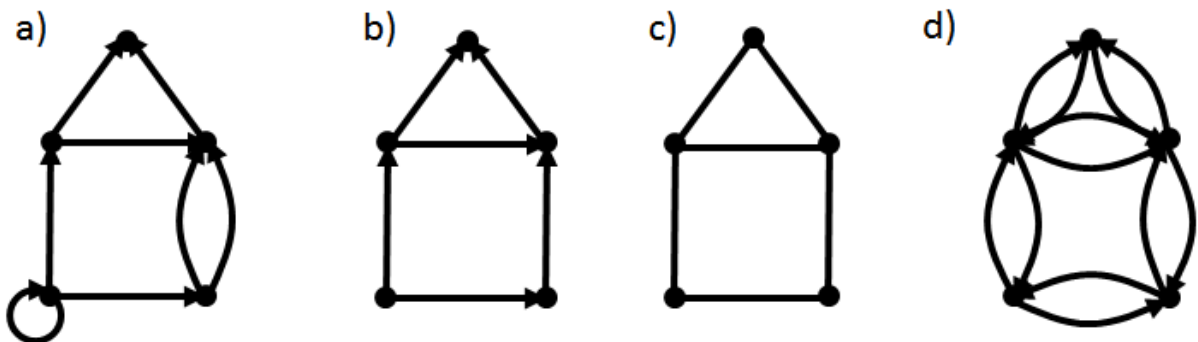
Podgraf H neorientovaného či orientovaného grafu G , který lze vytvořit z grafu G pouze pomocí operace odstranění hrany, se nazývá faktor H grafu G .

Graf znázorněný na obr. 2.1.3c je faktor grafu G znázorněného na obr. 2.1.3a. Faktorem grafu G je např. i sám graf G (operace odstranění hrany nepoužita) nebo graf G s odstraněnými všemi hranami (tzv. diskrétní graf; operace odstranění hrany použita $e(G)$ -krát).

Definice 2.1.20 [1]

Faktor H grafu G (neorientovaného nebo orientovaného), který lze vytvořit pouze odstraněním všech smyček grafu G a odstraněním $|K| - 1$ prvků každé množiny K , která obsahuje právě všechny hrany grafu G , které jsou navzájem paralelní, se nazývá základní prostý graf H grafu G .

Na obr. 2.1.4b je znázorněn základní prostý graf grafu znázorněného na obr. 2.1.4a.



Obr. 2.1.4: a), b) Orientovaný graf a jeho základní prostý graf, c), d) Graf a jeho symetrická orientace

Definice 2.1.21 [1]

Graf G , který vznikne z orientovaného grafu D nahrazením každé uspořádané dvojice (u, v) oboru hodnot incidenční funkce grafu G množinou $\{u, v\}$, se nazývá základní graf orientovaného grafu D a značí $G(D)$.

Na obr. 2.1.4c je znázorněn základní graf $G(D)$ grafu D znázorněného na obr. 2.1.4b.

Definice 2.1.22 [1]

Orientovaný graf D , který vznikne z grafu G nahrazením každého prvku $(e_i, \{u, v\})$ relace ψ_D dvěma prvky $(a_{i,1}, (u, v))$ a $(a_{i,2}, (v, u))$ a každého prvku e_i množiny $E(G)$ dvojicí $a_{i,1}$ a $a_{i,2}$, se nazývá symetrická orientace D grafu G a značí $D(G)$.

Na obr. 2.1.4d je znázorněna symetrická orientace $F(G)$ grafu G znázorněného na obr. 2.1.4c.

Definice 2.1.23 [1]

Orientovaný graf D , který vznikne z grafu G nahrazením každého prvku $(e_i, \{u, v\})$ relace ψ_D právě jedním z prvků $(a_{i,1}, (u, v))$ a $(a_{i,2}, (v, u))$ a každého prvku e_i množiny $E(G)$ příslušným prvkem z dvojice $a_{i,1}$ a $a_{i,2}$, se nazývá orientace D grafu G a značí \vec{G} .

Na obr. 2.1.4b je znázorněn příklad orientace \vec{G} grafu G znázorněného na obr. 2.1.4c.

Definice 2.1.24 [1]

Sjednocení neorientovaných, resp. orientovaných, grafů G a H je neorientovaný, resp. orientovaný, graf F takový, že $V(F) = V(G) \cup V(H) \wedge E(F) = E(G) \cup E(H)$ a ψ_F je definována takto: $\psi_F(e) = \begin{cases} \psi_G(e), & \text{pokud } e \in E(G), \\ \psi_H(e), & \text{pokud } e \in E(H). \end{cases}$ Graf F označujeme $G \cup H$.

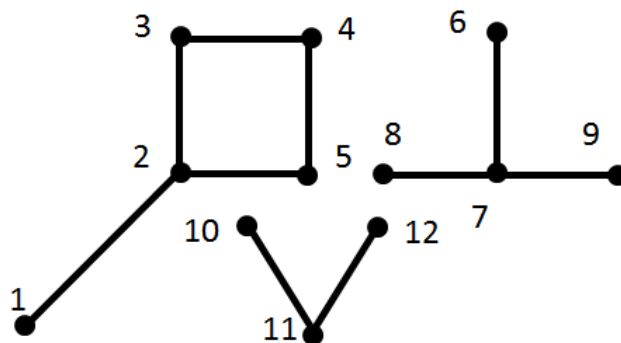
Na obr. 2.1.3a je znázorněno sjednocení grafů znázorněných na obr. 2.1.3b a 2.1.3c.

Definice 2.1.25 [1]

Neorientovaný graf G se nazývá souvislý, pokud pro každé rozdělení množiny $V(G)$ na dvě její neprázdné navzájem disjunktní podmnožiny X a Y existuje hrana grafu G , jejíž jeden vrchol je v X a druhý v Y . V opačném případě se graf G nazývá nesouvislý. Orientovaný graf D je souvislý, právě když je jeho základní graf souvislý, a je nesouvislý, právě když je jeho základní graf nesouvislý. Neprázdný souvislý podgraf neorientovaného či orientovaného grafu H , který není výsledkem odstranění jedné hrany nebo jednoho vrcholu jiného souvislého podgrafu grafu H , se nazývá komponenta grafu H .

Graf G znázorněný na obr. 2.1.5 je nesouvislý a obsahuje tři komponenty $G[\{1, 2, 3, 4, 5\}]$, $G[\{6, 7, 8, 9\}]$ a $G[\{10, 11, 12\}]$.

Dále budeme uvažovat jen grafy (neorientované nebo orientované), které jsou konečné a nejsou prázdné.

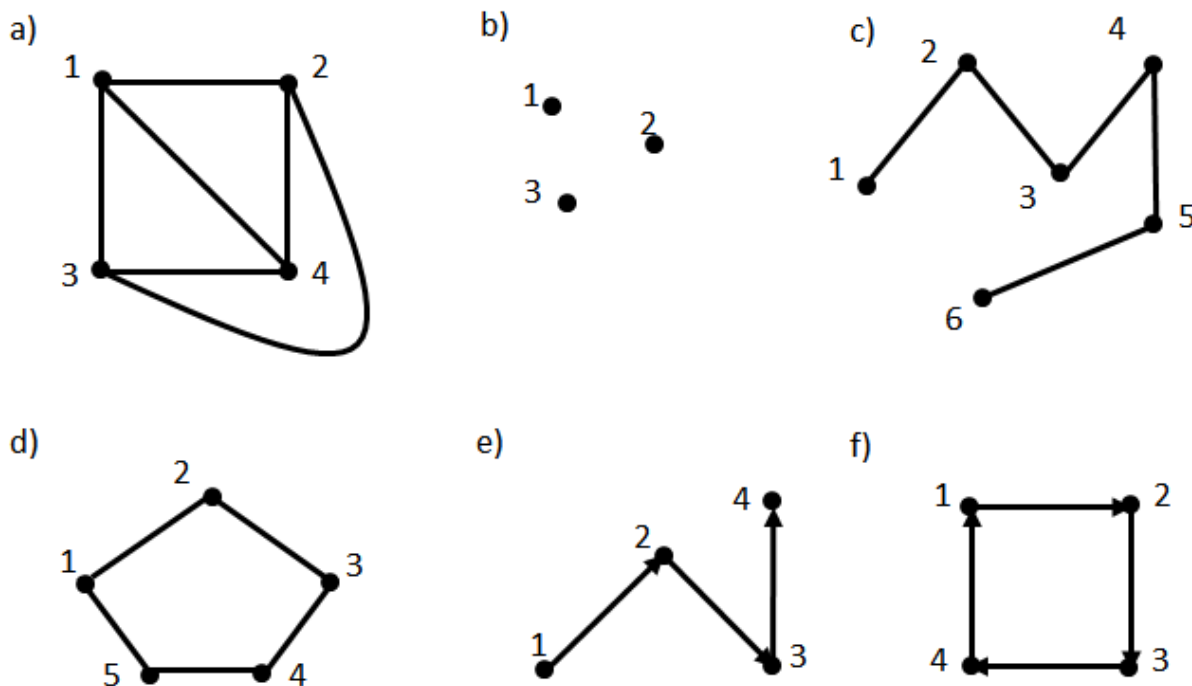


Obr. 2.1.5: Nesouvislý graf

Definice 2.1.26 [1]

Prostý graf, jehož každé dva vrcholy jsou vzájemní sousedé, se nazývá úplný graf a obvykle se značí K_n , kde n značí řád tohoto grafu. Prostý graf, jehož žádné dva vrcholy nejsou vzájemní sousedé, se nazývá diskrétní graf, a obvykle se značí D_n , kde n značí řád tohoto grafu.

Na obr. 2.1.6a je znázorněn graf K_4 a na obr. 2.1.6b graf D_3 .



Obr. 4: a) Graf K_4 , b) Graf D_3 , c) Cesta, d) Kružnice, e) Orientovaná cesta, f) Orientovaná kružnice

Definice 2.1.27 [1]

Cesta P je neorientovaný prostý graf, jehož všechny vrcholy je možné seřadit v posloupnost, pro jejíž každé dva prvky u, v platí: u bezprostředně následuje po v nebo

v následuje bezprostředně po u , právě když je u sousedem v . Pokud jsou první a poslední prvek této posloupnosti u a v , nazýváme graf P cesta z u do v nebo uv -cesta a říkáme též, že cesta P vede z vrcholu u do vrcholu v . Orientovaná cesta Q je orientace některé cesty P , přičemž orientovaná hrana vede vždy pouze z předchůdce x do jeho bezprostředního následníka y v posloupnosti vrcholů cesty P , přičemž uvažujeme jednu ze dvou možných posloupností. Pokud jsou první a poslední prvek uvažované posloupnosti u a v , nazýváme graf Q orientovaná cesta z u do v nebo orientovaná uv -cesta a říkáme též, že cesta Q vede z vrcholu u do vrcholu v . Počet hran cesty (neorientované či orientované) se nazývá délka cesty.

Na obr. 2.1.6c je znázorněna cesta délky 5 a na obr. 2.1.6e orientovaná cesta délky 3.

Definice 2.1.28 [1]

Kružnice C je buď smyčka, nebo dvě navzájem paralelní hrany, anebo neorientovaný prostý graf řádu alespoň tři, jehož všechny vrcholy je možné seřadit v posloupnost, jejíž první a poslední prvek jsou si rovny a navíc platí pro každé dva prvky u, v : u bezprostředně následuje po v nebo v následuje bezprostředně po u , právě když je u sousedem v . Cyklus neboli orientovaná kružnice D je buď orientovaná smyčka, nebo dvě paralelní orientované hrany s navzájem opačnými počátečními a koncovými vrcholy, anebo orientace některé neorientované kružnice C o alespoň třech vrcholech, přičemž orientovaná hrana vede vždy pouze z předchůdce x do jeho bezprostředního následníka y v posloupnosti kružnice C (přičemž uvažujeme jednu ze dvou možných posloupností). Délka neorientované či orientované kružnice F je počet jejích hran.

Na obr. 2.1.6d je znázorněna kružnice délky 5 a na obr. 2.1.6f cyklus délky 4.

Definice 2.1.29 [1]

Pokud neorientovaný, resp. orientovaný, graf G neobsahuje kružnici, resp. cyklus, nazývá se acyklický. Pokud neorientovaný, resp. orientovaný, graf G obsahuje kružnici, resp. cyklus, pak nejmenší délka kružnice, resp. cyklu, v grafu G se nazývá obvod grafu G . Acyklický graf G se též nazývá (neorientovaný, resp. orientovaný) les, a pokud je souvislý, nazývá se též (neorientovaný, resp. orientovaný) strom. Vrchol stromu G stupně jedna se nazývá list. Pokud je podgraf grafu G strom, nazývá se tento podgraf

také (neorientovaný, resp. orientovaný) podstrom, a pokud obsahuje všechny vrcholy grafu G , nazývá se (neorientovaná, resp. orientovaná) kostra grafu G .

Stromem je např. graf znázorněný na obr. 2.1.3c, který je zároveň cestou. Podgraf $G[\{6, 7, 8, 9\}]$ grafu G znázorněného na obr. 2.1.5 je také strom, ale není cesta. Podgraf $G[\{6, 7, 8, 9, 10, 11, 12\}]$ grafu G znázorněného na obr. 2.1.5 je les.

2.2 Speciální pojmy teorie grafů k hranové souvislosti grafu

Definice 2.2.1 [1]

Pro dvě podmnožiny X a Y množiny $V(G)$ grafu G označujeme množinu všech hran spojujících nějaký vrchol z X s nějakým vrcholem z Y $E[X, Y]$ a jejich počet $e(X, Y)$. Pokud $X = Y$, označujeme také místo $E[X, X]$ pouze $E[X]$ a místo $e(X, X)$ pouze $e(X)$. Pokud $Y = V(G) \setminus X$, nazývá se množina $E[X, Y]$ hranový řez grafu G asociovaný s X a označuje se $\partial(X)$. Hranový řez grafu G asociovaný s jednoprvkovou množinou X se nazývá triviální hranový řez.

Např. hranový řez $\partial(\{1, 3\})$ grafu znázorněného na obr. 2.1.1a je množina hran $\{a, b, f\}$.

Definice 2.2.2 [1]

Pro dvě podmnožiny X a Y množiny $V(D)$ orientovaného grafu D označujeme množinu všech hran vedoucích z nějakého vrcholu z X do nějakého vrcholu z Y $A[X, Y]$ a jejich počet $a(X, Y)$. Pokud $X = Y$, označujeme také místo $A[X, X]$ pouze $A[X]$ a místo $a(X, X)$ pouze $a(X)$. Pokud $Y = V(D) \setminus X$, nazývá se množina $A[X, Y]$ výstupní hranový řez grafu D asociovaný s X a označuje se $\partial^+(X)$ a množina $A[Y, X]$ se nazývá vstupní hranový řez grafu D asociovaný s X a označuje se $\partial^-(X)$.

Např. výstupní hranový řez $\partial^+(\{4, 5\})$ grafu D znázorněného na obr. 2.1.1b je množina orientovaných hran $\{c, d\}$. Vstupním hranovým řezem $\partial^-(\{4, 5\})$ grafu D je prázdná množina.

Definice 2.2.3 [1]

Síť S je dvojice (N, c) , kde je N orientovaný graf obsahující zdroj x a stok y a c je reálná nezáporná funkce s definičním oborem $A(N)$. Funkci c nazýváme kapacitní funkcí sítě S a hodnotu $c(a)$ pro každou hranu a nazýváme kapacita hrany a .

Sít si lze představit jako vodovodní potrubní síť, která je reprezentována grafem N , jejíž jednotlivé potrubní úseky (hrany grafu N) se křižují ve vrcholech grafu N . Funkce c pak udává propustnost každého úseku za jednotku času. Zdrojem voda do potrubní sítě přitéká a stokem odtéká.

Definice 2.2.4 [1]

Pokud je f reálná funkce na množině A a $S \subseteq A$, označujeme $\sum_{a \in S} f(a)$ jako $f(S)$. Pokud $A = A(D)$ a $X \subseteq V(D)$, kde D je orientovaný graf, označujeme $f^+(X) = f(\partial^+(X))$ a $f^-(X) = f(\partial^-(X))$. Hodnota $c(\partial^+(X))$, kde X je podmnožina množiny $V(N)$ sítě (N, c) , se obvykle značí $cap(\partial^+(X))$.

Definice 2.2.5 [1]

Tok (též (x, y) -tok) v síti (N, c) se zdrojem x a stokem y je reálná funkce f s definičním oborem $A(N)$ splňující podmínku: $\forall v \in V(N \setminus \{x, y\}): f^+(\{v\}) = f^-(\{v\})$ (tzv. konzervační podmínka). Tok f se nazývá realizovatelný, jestliže splňuje podmínku: $\forall a \in A(N): 0 \leq f(a) \leq c(a)$. Pokud $X \subseteq V(N)$, nazýváme rozdíl $f^+(X) - f^-(X)$ síťovým tokem z X a rozdíl $f^-(X) - f^+(X)$ síťovým tokem do X . Hodnota síťového toku f ze zdroje sítě (N, c) , tj. z množiny $\{x\}$, stejně jako jí rovná hodnota síťového toku f do stoku sítě (N, c) , tj. do množiny $\{y\}$, se nazývá hodnota toku f a označuje $val(f)$. Hrana a sítě (N, c) s tokem f se nazývá f -nulová, jestliže $f(a) = 0$, f -pozitivní, jestliže $f(a) > 0$, f -nenасыcená, jestliže $f(a) < c(a)$, a f -nасыcená, jestliže $f(a) = c(a)$.

Realizovatelný tok v síti tedy určuje v naší představě vodovodní potrubní sítě, jaké množství vody proteče každým potrubním úsekem (hranou) za jednotku času, a hodnota toku určuje, jaké množství vody proteče potrubní sítí celkově za jednotku času. Tok je definován tak, že co do vrcholu přiteče, to z něj též odeče (konzervační podmínka). Podmínkou pro realizovatelnost toku v síti je, aby každým úsekem protékalo nezáporné množství vody a zároveň nejvýše tolik, kolik je propustnost tohoto úseku.

Definice 2.2.6 [1]

Tok f realizovatelný v síti (N, c) se nazývá maximální, jestliže pro každý tok g realizovatelný v síti (N, c) platí: $val(g) \leq val(f)$. Řez $\partial^+(X)$ grafu N sítě (N, c) se zdrojem x a stokem y , kde $x \in X \wedge y \in V(N) \setminus X$, se nazývá minimální, jestliže pro každý řez $\partial^+(Y)$ grafu N , kde $x \in Y \wedge y \in V(N) \setminus Y$, platí $cap(\partial^+(X)) \leq cap(\partial^+(Y))$.

Věta 2.2.1 [1]

Pro každý tok f realizovatelný v síti $(N(x, y), c)$ a každou podmnožinu X množiny $V(N)$ takovou, že $x \in X \wedge y \in V(N) \setminus X$, platí $val(f) = f^+(X) - f^-(X)$.

Podle předchozí věty 2.2.1 je tedy hodnota toku v síti rovna hodnotě toku z libovolné množiny X , která obsahuje zdroj a neobsahuje stok sítě. Navíc lze z předchozí věty 2.2.1 odvodit následující větu 2.2.2.

Věta 2.2.2 [1]

Pro každý tok f realizovatelný v síti (N, c) a každý řez $\partial^+(X)$ grafu N sítě (N, c) se zdrojem x a stokem y , kde $x \in X \wedge y \in V(N) \setminus X$, platí $val(f) \leq cap(\partial^+(X))$. Rovnost nastává, právě když je každá hrana z $\partial^+(X)$ f -nasycená a každá hrana z $\partial^-(X)$ f -nulová.

Důsledkem předchozí věty 2.2.2 je následující věta 2.2.3, podle které lze zjistit, zda je daný tok v síti maximální.

Věta 2.2.3 [1]

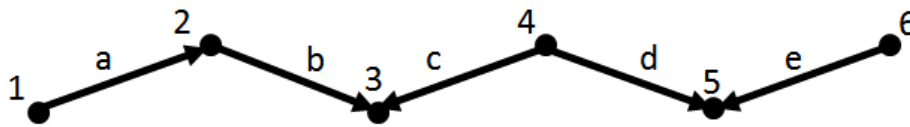
Nechť je f tok realizovatelný v síti (N, c) a $\partial^+(X)$ řez grafu N sítě (N, c) se zdrojem x a stokem y , kde $x \in X \wedge y \in V(N) \setminus X$. Jestliže $val(f) = cap(\partial^+(X))$, pak je f maximální tok a $\partial^+(X)$ minimální řez sítě (N, c) .

Definice 2.2.7 [1]

Polocesta orientovaného grafu D je podgraf P grafu D takový, že základní graf grafu P je cesta. Pokud tato cesta vede z vrcholu x do vrcholu y , nazývá se graf P polocesta z u do v nebo uv -polocesta a říkáme též, že polocesta P vede z vrcholu u do vrcholu v . Počet hran polocesty P se nazývá délka polocesty P . Souhlasná hrana xy -polocesty P grafu D je hrana polocesty P s počátečním vrcholem u a koncovým vrcholem v , pro kterou platí, že v základním grafu polocesty P má xu -cesta menší délku než xv -cesta. Nesouhlasná hrana xy -polocesty P grafu D je hrana polocesty P s počátečním vrcholem r a koncovým vrcholem s , pro kterou platí, že v základním grafu polocesty P má xr -cesta větší délku než xs -cesta.

Na obr. 2.2.1 je znázorněn příklad polocesty P . Pokud uvažujeme, že polocesta P vede z vrcholu 1 do vrcholu 6, pak hrany a , b a d jsou souhlasné a hrany c a e

nesouhlasné. Dalším příkladem polocesty je orientovaná cesta znázorněná na obr. 2.1.6e.



Obrázek 2.2.1: Polocesta

Definice 2.2.8 [1]

Nechť je $(N(x, y), c)$ síť se zdrojem x a stokem y a f je tok realizovatelný v síti $(N(x, y), c)$. Nechť $\epsilon(P) = \min\{\epsilon(a); a \in A(P)\}$, kde P je polocesta v síti $(N(x, y), c)$ vedoucí z vrcholu x , přičemž $\epsilon(a) = c(a) - f(a)$, jestliže je a souhlasná hrana polocesty P , a $\epsilon(a) = f(a)$, jestliže je a nesouhlasná hrana polocesty P . Polocesta P se nazývá f -nasyčená, jestliže $\epsilon(P) = 0$, a f -nenasyčená, jestliže $\epsilon(P) > 0$. O f -nenasyčené xy -polocestě říkáme, že je f -rezervní.

Polocesty, které jsou f -rezervní, lze použít ke zvýšení hodnoty toku v síti podle následujícího algoritmu 2.2.1.

Algoritmus 2.2.1 (Zvýšení hodnoty toku v síti) [1]

- Vstup: Tok f realizovatelný v síti $(N(x, y), c)$ se zdrojem x a stokem y , který není maximální.
- Výstup: Tok f' v síti $(N(x, y), c)$ takový, že $val(f) < val(f')$.

1) Přiřaď každé polocestě P vedoucí z vrcholu x hodnotu $\epsilon(P)$.

2) Mezi polocestami P najdi f -rezervní polocestu Q a nahraď tok f tokem f' :

$$f'(a) := \begin{cases} f(a) + \epsilon(Q), & \text{jestliže je } a \text{ souhlasná hrana polocesty } Q, \\ f(a) - \epsilon(Q), & \text{jestliže je } a \text{ nesouhlasná hrana polocesty } Q, \\ f(a) & \text{jinak.} \end{cases}$$

3) Vrať f' .

Správnost algoritmu potvrzuje následující věta 2.2.4.

Věta 2.2.4 [1]

Nechť je f tok realizovatelný v síti (N, c) . Jestliže existuje f -rezervní polocesta P v síti (N, c) vedoucí ze zdroje sítě (N, c) , pak f není maximální tok v síti (N, c) . Funkce f' , definovaná ve druhé kroku algoritmu 2.2.1, je tok v (N, c) s $val(f') = val(f) + \epsilon(P)$.

Věta 2.2.5 [1]

Nechť je f tok realizovatelný v síti $(N(x, y), c)$ se zdrojem x a stokem y a necht' neexistuje f -rezervní polocesta v síti $(N(x, y), c)$ vedoucí z x . Necht' je X množina všech vrcholů grafu N , do nichž vede z vrcholu x f -nenасыcená polocesta. Pak je f maximální tok v síti $(N(x, y), c)$ a $\partial^+(X)$ je minimální řez v síti $(N(x, y), c)$.

Důsledkem předchozích vět 2.2.4 a 2.2.5 je následující věta 2.2.6, podle které je v síti hodnota maximálního toku rovna kapacitě minimálního řezu.

Věta 2.2.6 [1]

V každé síti (N, c) platí $val(f_{max}) = cap(\partial^+(X_{min}))$, kde f_{max} je maximální tok sítě (N, c) a $\partial^+(X_{min})$ je minimální řez sítě (N, c) .

Důsledkem předchozích vět 2.2.4 a 2.2.5 je i následující věta 2.2.7, podle které můžeme algoritmus 2.2.1 rozšířit na algoritmus 2.2.2 pro nalezení maximálního toku v síti.

Věta 2.2.7 [1]

Tok f v síti (N, c) je maximální v (N, c) , právě když v (N, c) neexistuje f -rezervní polocesta vedoucí ze zdroje sítě (N, c) .

Algoritmus 2.2.2 (Fordův-Fulkersonův algoritmus) [1]

- Vstup: síť $(N(x, y), c)$ se zdrojem x a stokem y , kde je c racionální funkce, tok f realizovatelný v $(N(x, y), c)$.
 - Výstup: maximální tok f a minimální řez $\partial^+(X)$ v $(N(x, y), c)$.
- 1) Polož $X := \{x\}$, $p(v) := 0$ pro každé $v \in V(N)$.
 - 2) Pokud pro $u \in X$ a $v \in V(N) \setminus X$ existuje f -nenасыcená hrana vedoucí z u do v nebo f -pozitivní hrana vedoucí z v do u :
 - a. Polož $X := X \cup \{v\}$.
 - b. Polož $p(v) := u$.
 - c. Vrať se na 2.
 - 3) Jestliže $y \in X$, pak
 - a. Spočítej $\epsilon(P) = \min\{\epsilon(a) : a \in A(P)\}$, kde P je xy -polocesta v orientovaném stromu T daném zobrazením p , tj. T obsahuje právě ty hrany, které splnily podmínku v bezprostředně předchozím kroku 2.

- b. Pro každou souhlasnou hranu a polocesty P nahrad' $f(a)$ hodnotou $f(a) + \epsilon(P)$.
 - c. Pro každou nesouhlasnou hranu a polocesty P nahrad' $f(a)$ hodnotou $f(a) - \epsilon(P)$.
 - d. Vrat' se na 1.
- 4) Vrat' $(f, \partial^+(X))$.

Poznamenejme, že strom T v kroku 3a algoritmu 2.2.2 je jednoznačně určen zobrazením p tak, že $V(T) = \{x\} \cup \{v \in V(N); p(v) \neq 0\}$ a hrany stromu T jsou určeny takto: $\forall v \in V(T) \setminus \{x\}$: z vrcholu v vede hrana e do vrcholu $p(v)$, právě když je hrana e f -pozitivní hrana nalezená v kroku 2 vedoucí z vrcholu v do vrcholu $p(v)$, a z vrcholu $p(v)$ vede hrana e do vrcholu v , právě když je hrana e f -nasycená hrana nalezená v kroku 2 vedoucí z vrcholu $p(v)$ do vrcholu v .

V algoritmu 2.2.2 se tedy nejdříve najde f -rezervní xy -polocesta (v případě její existence), podle které se zvýší hodnota toku. Postupně se takto hledají f -rezervní xy -polocesty a zvyšuje tok do okamžiku, kdy už žádná f -rezervní xy -polocesta neexistuje. Podle věty 2.2.7 pak algoritmus 2.2.2 vrátí maximální tok (a minimální řez). Na vstupu algoritmu je vyžadována racionální funkce c , jelikož pro iracionální hodnoty by se nemuselo dojít k výsledku v konečném čase [1].

Definice 2.2.9 [1]

Říkáme, že xy -cesty P a Q obsažené v grafu G (neorientovaném nebo orientovaném) jsou hranově disjunktní, jestliže P a Q neobsahují žádnou společnou hranu.

Věta 2.2.8 [1]

Nechť je (N, c) síť, ve které platí: $\forall a \in A(N): c(a) = 1$. Graf N obsahuje k hranově disjunktních xy -cest, právě když existuje tok f realizovatelný v síti (N, c) s $val(f) = k$.

Na základě předchozí věty 2.2.8 lze dokázat věty 2.2.9 a 2.2.14 (hranové verze Mengerovy věty).

Pokud odstraníme jednu hranu z libovolného úplného grafu G řádu alespoň 3, získáme jeho souvislý podgraf. Pokud však odstraníme hranu z libovolného netriviálního stromu T , získáme nesouvislý podgraf. Oba grafy G a T jsou souvislé, ale graf G je

z určitého pohledu více souvislý než graf T . Proto se zavádějí parametry souvislosti grafů, např. hranová souvislost grafu.

Definice 2.2.10 [1]

Lokální hranová souvislost mezi navzájem různými vrcholy x a y v grafu G je největší počet po dvou hranově disjunktních xy -cest v G a označuje se $p'(x, y)$.

Definice 2.2.11 [1]

Netriviální graf G je hranově k -souvislý, jestliže $p'(u, v) \geq k$ pro každé dva navzájem různé vrcholy u, v v G . Triviální graf je hranově 0-souvislý a hranově 1-souvislý.

Definice 2.2.12 [1]

Hranová souvislost $\kappa'(G)$ grafu G je největší hodnota k , pro kterou je G hranově k -souvislý.

Hranová souvislost je tedy jedním z parametrů, podle kterých můžeme grafy porovnávat z hlediska jejich souvislosti. Např. pro každý netriviální úplný graf K_n platí $\kappa'(G) = n - 1$ a pro každý strom T platí $\kappa'(T) = 1$.

Definice 2.2.13[1]

Říkáme, že hranový řez $\partial(X)$ grafu G odděluje dva vrcholy x a y , jestliže $x \in X \wedge y \in V(G) \setminus X$. Nejmenší velikost hranového řezu oddělujícího vrcholy x a y v grafu G se označuje $c'(x, y)$.

Věta 2.2.9 (Mengerova věta – hranová verze pro neorientované grafy) [1]

Pro každý graf $G(x, y)$, kde $x \neq y$, platí $p'(x, y) = c'(x, y)$.

Předchozí věta 2.2.9 podává návod pro určení hranové souvislosti grafu tím, že převádí problém hledání hranově disjunktních cest na problém hledání hranového řezu. Toho využijeme např. v algoritmu 3.1.1.

Existují i jiné parametry souvislosti grafu než hranová souvislost. Představíme zatím ještě alespoň tzv. vrcholovou souvislost grafu, označovanou často jen jako souvislost grafu.

Definice 2.2.14 [1]

Říkáme, že xy -cesty P a Q obsažené v grafu G (neorientovaném nebo orientovaném) jsou vnitřně disjunktní, jestliže neobsahují žádný společný vrchol kromě x a y .

Definice 2.2.15 [1]

Lokální (vrcholová) souvislost mezi navzájem různými vrcholy x a y v grafu G je největší počet po dvou vnitřně disjunktních xy -cest v G a označuje se $p(x, y)$.

Definice 2.2.16 [1]

Netriviální graf G je k -souvislý, jestliže $p(u, v) \geq k$ pro každé dva navzájem různé vrcholy u, v v G . Triviální graf je 0-souvislý a 1-souvislý.

Definice 2.2.17 [1]

Souvislost $\kappa(G)$ grafu G je největší hodnota k , pro kterou je G k -souvislý.

Definice 2.2.18 [1]

Pro dva navzájem různé nesousední vrcholy x a y grafu G je vrcholový xy -řez podmnožina S množiny $V(G) \setminus \{x, y\}$ taková, že x náleží jiné komponentě grafu $G - S$ než y . Říkáme, že množina S odděluje vrcholy x a y . Nejmenší hodnota vrcholového xy -řezu grafu G se značí $c(x, y)$. Každý vrcholový xy -řez grafu G se nazývá též jen vrcholový řez grafu G , a pokud obsahuje právě k prvků, nazývá se též vrcholový k -řez grafu G .

Např. v grafu G znázorněném na obr. 2.1.1a je vrcholovým řezem oddělujícím vrcholy 1 a 4 množina $\{2, 3\}$, a jelikož žádný jiný (tudíž ani menší) vrcholový řez oddělující 1 a 4 v G neexistuje, platí $c(2, 4) = 2$.

Věta 2.2.10 (Mengerova věta – vrcholová verze pro neorientované grafy) [1]

V každém grafu $G(x, y)$, kde x a y jsou navzájem různé nesousední vrcholy, platí $p(x, y) = c(x, y)$.

Věta 2.2.11 [1]

Pokud obsahuje graf G alespoň dva navzájem různé nesousední vrcholy, pak $\kappa(G) = \min\{p(u, v) : u \in V(G) \wedge v \in V(G) \wedge u \neq v \wedge uv \notin E(G)\}$.

Podle věty 2.2.17 tedy při zjišťování souvislosti grafu (obsahujícího alespoň dva navzájem nesousední vrcholy) stačí hledat navzájem disjunktní cesty jen mezi navzájem nesousedními vrcholy. Důsledkem vět 2.2.10 a 2.2.11 je následující věta 2.2.12.

Věta 2.2.12 [1]

Pokud obsahuje graf G alespoň dva navzájem různé nesousední vrcholy, pak $\kappa(G) = \min\{c(u, v) : u \in V(G) \wedge v \in V(G) \wedge u \neq v \wedge uv \notin E(G)\}$.

Věta 2.2.12 obdobně jako věta 2.2.9 převádí problém hledání disjunktních cest na problém hledání minimálního řezu.

Pro graf G , jehož každé dva navzájem různé vrcholy jsou sousední, není vrcholový řez definován, a proto nelze pro nalezení jeho vrcholové souvislosti použít větu 2.2.12. Jeho vrcholovou souvislost lze však zjistit přímo. Pokud je takový graf G prostý, pak se jedná o úplný graf, jehož vrcholová souvislost je $v(G) - 1$ v případě netriviálního grafu G a 1 v případě triviálního grafu G . Pokud G není prostý, tak je jeho vrcholová souvislost $v(G) - 2 + k$, kde k označuje nejmenší počet hran spojujících libovolné dva navzájem různé vrcholy grafu G .

Vztah mezi vrcholovou a hranovou souvislostí grafu podává následující věta 2.2.13, která hodnoty těchto souvislostí navíc omezuje shora minimálním stupněm vrcholu v grafu G .

Věta 2.2.13 [1]

Pro každý graf G platí: $\kappa(G) \leq \kappa'(G) \leq \delta(G)$.

Pro úplnost ještě uvedeme verze Mengerovy věty pro orientované grafy.

Definice 2.2.19 [1]

Říkáme, že xy -cesty P a Q obsažené v orientovaném grafu D jsou hranově disjunktní, jestliže P a Q neobsahují žádnou společnou hranu.

Věta 2.2.14 (Mengerova věta – hranová verze pro orientované grafy) [1]

V každém orientovaném grafu $D(x, y)$, kde $x \neq y$, je maximální počet po dvou hranově disjunktních xy -cest roven velikosti minimálního hranového řezu $\partial^+(X)$, kde $x \in X$ a $y \notin X$.

Minimálním řezem $\partial^+(X)$ v předchozí větě 2.2.14 se rozumí takový řez $\partial^+(X)$, že pro každý hranový řez $\partial^+(Y)$ v grafu D , kde $x \in Y$ a $y \notin Y$, platí $|\partial^+(X)| \leq |\partial^+(Y)|$.

Definice 2.2.20 [1]

Vrcholový xy -řez orientovaného grafu D je taková podmnožina S množiny $V(D) \setminus \{x, y\}$, že graf $D - S$ neobsahuje žádnou xy -cestu.

Např. v grafu znázorněném na obr. 2.1.1b je vrcholovým 5-1-řezem množina $\{3\}$.

Věta 2.2.15 (Mengerova věta – vrcholová verze pro orientované grafy) [1]

V každém orientovaném grafu $D(x, y)$, ve kterém nevede hrana z vrcholu x do vrcholu y a $x \neq y$, je největší počet po dvou hranově disjunktních xy -cest roven nejmenší velikosti vrcholového xy -řezu.

2.3 Speciální pojmy k implementaci algoritmů

Nyní uvedeme, jak se algoritmy rozlišují podle jejich časové složitosti.

Každý algoritmus řeší nějaký problém, např. určení hranové souvislosti grafu. Instancí problému se rozumí konkrétní vstup algoritmu, např. graf G . Velikost instance problému je jeho délka v kódování použitém při dané implementaci algoritmu. V grafových algoritmech považujeme za velikost instance problému (grafu) často počet vrcholů vstupního grafu. Necht' je P deterministický algoritmus (tj. pro konkrétní vstup algoritmu proběhne vždy stejný výpočet algoritmu) a necht' p je zobrazení z \mathbb{N} do \mathbb{N} takové, že pro každé $n \in \mathbb{N}$ je $p(n)$ maximální počet elementárních kroků algoritmu P , které se vykonají při vstupním grafu G řádu n (tj. při instanci velikosti n). Algoritmus P patří do třídy časové složitosti $O(f)$, kde f je zobrazení z \mathbb{N} do \mathbb{N} , jestliže existuje jeho implementace taková, že platí: $\exists n_0 \in \mathbb{N} \forall n \in \mathbb{N}: (n > n_0) \Rightarrow |p(n)| < c|f(n)|$, kde c je reálná konstanta [4]. Říkáme, že algoritmus P pracuje v čase $O(f)$ nebo že má časovou složitost $O(f)$. Pokud např. patří grafový algoritmus do $O(n^2)$ (tj. má časovou složitost $O(n^2)$), znamená to, že existuje jeho implementace, která vyřeší problém, který algoritmus řeší, pro graf řádu n po nejvýše cn^2 elementárních krocích, např. po $100n^2$ krocích. Pokud algoritmus patří do $O(f)$, kde je f polynom, resp. lineární funkce, jedná se o polynomiální, resp. lineární, algoritmus a říkáme také, že pracuje v polynomiálním, resp. lineárním čase. Algoritmy pracující v polynomiálním čase se považují za efektivní.

Kromě časové složitosti algoritmu je také důležité, zda algoritmus vrací pro daný vstup skutečně správný výsledek. Jednou možností, jak to ověřit, je dokázat správnost algoritmu, což však může být v řadě případů obtížný problém. Např. v algoritmu pro testování rovinnosti grafů J. Hopcrofta a R. Tarjana došlo k objevení chyby až po dvou letech [8]. Druhou možností je v průběhu algoritmu vytvářet tzv. certifikát, podle kterého bude na konci algoritmu možné rychle určit, zda je výsledek algoritmu správný. V ideálním případě by mělo trvat určení správnosti výsledku algoritmu pro vstup G pomocí certifikátu nejvýše stejně dlouho jako samotný výpočet pro vstup G . Algoritmy, ve kterých se vytváří certifikát, se nazývají certifikační a jejich příkladem bude algoritmus 3.4.1 pro testování hranové 3-souvislosti grafu.

Algoritmy se implementují v tzv. programovacích jazycích. Mezi nejpoužívanější programovací jazyky patří např. jazyk C (který má několik variant, např. C++), ve kterém implementujeme i algoritmus v kapitole 4.

3 Přehled algoritmů hranové souvislosti grafu

3.1 Algoritmus pro určení hranové souvislosti grafu s využitím toků

Nejdříve uvedeme algoritmus pro určení lokální hranové souvislosti grafu.

Algoritmus 3.1.1 (Lokální hranová souvislost grafu) [3]

- Vstup: graf G bez smyček, dva navzájem různé vrcholy x a y z $V(G)$.
 - Výstup: $p'(x, y)$.
- 1) Polož $G := D(G)$.
 - 2) Vytvoř síť $(G(x, y), c)$:
 - a. Označ x jako zdroj sítě $(G(x, y), c)$, y jako stok sítě $(G(x, y), c)$.
 - b. Polož $\forall a \in A(G): c(a) := 1$.
 - 3) Najdi maximální tok f v síti $(G(x, y), c)$.
 - 4) Polož $p'(x, y) := val(f)$ a vrať $p'(x, y)$.

V algoritmu 3.1.1 se tedy ze vstupního grafu vytvoří síť a v ní se najde maximální tok. Ve třetím kroku algoritmu 3.1.1 je možné použít k nalezení maximálního toku algoritmus 2.2.2. Algoritmus je založen na platnosti vět 2.2.8 a 2.2.9. Nejlepší varianty tohoto algoritmu (podle toho, jak se implementuje třetí krok) patří do třídy časové složitosti $O(mn)$, kde m je počet hran a n počet vrcholů vstupního grafu [3].

Podle algoritmu 3.1.1 dokážeme tedy spočítat lokální hranovou souvislost grafu. S jeho pomocí lze spočítat i hranovou souvislost grafu podle následujícího algoritmu 3.1.2. Pro určení hranové souvislosti grafu stačí spočítat všechny jeho lokální hranové souvislosti a vybrat z nich minimum. Pokud si však vybereme vrchol x vstupního grafu G a budeme uvažovat minimální hranový řez grafu G , pak bude existovat v G vrchol y různý od x takový, že minimální hranový řez oddělující x a y bude zároveň minimálním hranovým řezem celého grafu (minimální hranový řez grafu je totiž zároveň minimálním řezem oddělujícím nějaké dva vrcholy v grafu). Nemusí se tedy počítat lokální hranová souvislost mezi všemi vrcholy grafu, ale jen pouze mezi vybraným vrcholem x a ostatními vrcholy v grafu tak, jak se to dělá v následujícím algoritmu 3.1.2.

Algoritmus 3.1.2 (Hranová souvislost grafu) [3]

- Vstup: souvislý netriviální graf G bez smyček.
 - Výstup: $\kappa'(G)$.
- 1) Vyber vrchol x z $V(G)$ a polož $X := V(G) \setminus \{x\}$.
 - 2) Najdi $p'(x, y)$ pro každé y z X .
 - 3) Polož $\kappa'(G) := \min\{p'(x, y); y \in X\}$ a vrať $\kappa'(G)$.

Ve druhém kroku algoritmu 3.1.2 je možné použít k nalezení $p'(x, y)$ algoritmus 3.1.1, a jelikož krok 2 probíhá v algoritmu $(n - 1)$ -krát a algoritmus 3.1.1 patří do $O(mn)$, patří algoritmus 3.1.2 do třídy časové složitosti $O(mn^2)$, kde m je počet hran a n je počet vrcholů vstupního grafu. Dalšími složitějšími modifikacemi algoritmu můžeme jeho časovou složitost ještě snížit až na $O(mn)$ [3]. Pokud nám stačí složitost $O(mn^2)$, můžeme algoritmus 3.1.2 modifikovat tak, aby vracel více než jen hranovou souvislost grafu. To dokáže následující algoritmus 3.1.3.

Definice 3.1.1 [1]

Nechť je dán graf G (neorientovaný či orientovaný) a funkce $w: E(G) \rightarrow \mathbb{R}$. Pro každou hranu e grafu G se hodnota $w(e)$ nazývá též váha hrany e v grafu G . Uspořádaná dvojice (G, w) se nazývá ohodnocený (též vážený) graf.

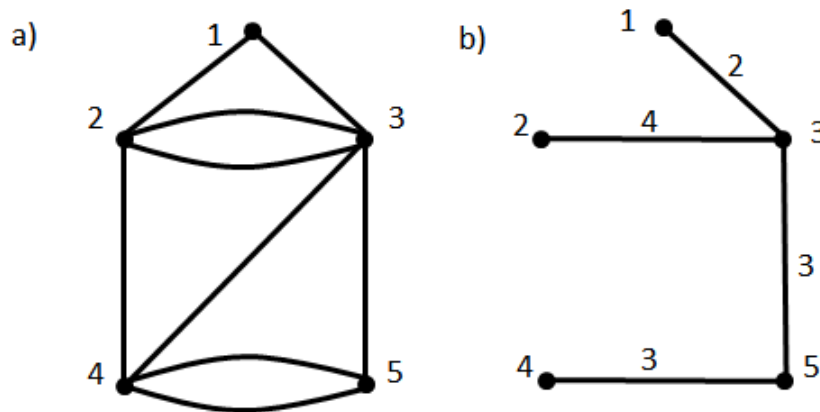
Příkladem ohodnoceného váženého grafu je např. síť podle definice 2.2.3.

Definice 3.1.2 [1]

Gomoryho-Huův strom grafu G je ohodnocený strom (T, w) s $V(T) = V(G)$ takový, že pro každou hranu e stromu T s koncovými vrcholy x a y platí $w(e) = c'(x, y)$ a hrana e představuje minimální hranový řez oddělující x a y .

Gomoryho-Huův strom grafu G tedy vzhledem k tvrzení věty 2.2.9 poskytuje informaci o všech lokálních hranových souvislostech vstupního grafu a bude výstupem algoritmu 3.1.3. Zdůrazněme, že hrany Gomoryho-Huova stromu grafu G reprezentují hranové řezy grafu G a nikoliv hrany grafu G , a tedy Gomoryho-Huův strom grafu G není obecně jeho kostrou.

Na obr. 3.1.1b je znázorněn Gomoryho-Huův strom grafu znázorněného na obr. 3.1.1a. Čísla u jednotlivých hran jsou ohodnocení těchto hran.



Obr. 3.1.1: a) Graf G , b) Gomoryho-Huův strom grafu G

Definice 3.1.3 [1]

Stažení dvou navzájem různých vrcholů x a y grafu G je odstranění všech hran spojujících x a y v G a ztotožnění vrcholů x a y . Stažení podmnožiny A množiny $V(G)$ grafu G je odstranění všech hran vedoucích mezi vrcholy z A a ztotožnění všech vrcholů z A .

Věta 3.1.1 [1]

Nechť je $\partial(X)$ minimální hranový řez v grafu G oddělující vrcholy x a y , kde $x \in X$, a necht' $\partial(Y)$ je minimální hranový řez v grafu G oddělující vrcholy u a v , kde $u \in X \wedge v \in X \wedge y \notin Y$. Pak $\partial(X \cap Y)$ je také minimální hranový řez v grafu G oddělující vrcholy u a v .

Věta 3.1.1 je použita v algoritmu 3.1.3 pro hledání Gomoryho-Huova stromu.

Věta 3.1.2 [1]

Nechť (T, w) je Gomoryho-Huův strom grafu G . Pro každé dva navzájem různé vrcholy x a y grafu G je hodnota $c'(x, y)$ rovna nejmenší váze hrany na xy -cestě v grafu T .

Na základě předchozí věty 3.1.2 a věty 2.2.9 tedy víme, že Gomoryho-Huův strom grafu G poskytuje informaci nejen o lokálních souvislostech grafu, ale i o hranové souvislosti grafu $\kappa'(G)$.

V následujícím algoritmu 3.1.3 se postupně rozrůstá orientovaný strom (T, w) , jehož vrcholy reprezentují navzájem disjunktní podmnožiny množiny $V(G)$ vstupního

grafu G , až do okamžiku, kdy bude každý vrchol stromu (T, w) reprezentovat jednoprvkovou množinu, tj. jeden vrchol grafu G .

Algoritmus 3.1.3 (Gomoryho-Huův strom) [1]

Vstup: souvislý graf G bez smyček řádu alespoň 2.

Výstup: Gomoryho-Huův strom (T, w) grafu G .

- 1) Urči minimální hranový řez S_1 oddělující dva navzájem různé vrcholy grafu G a polož $i := 1$.
- 2) Vytvoř hranu e_1 ohodnoceného stromu (T, w) , jejíž každý koncový vrchol bude představovat množinu vrcholů jedné komponenty grafu $G \setminus S_1$ (každý jinou), a polož $w(e_1) := |S_1|$.
- 3) Pokud existuje vrchol u stromu (T, w) , který představuje množinu U , jež obsahuje dva navzájem různé vrcholy x a y :
 - a. Polož $i := i + 1$.
 - b. Vytvoř z grafu G graf H stažením každé komponenty B_j (tzn. množiny všech jejích vrcholů) grafu $G - U$ do jednoho vrcholu b_j .
 - c. V grafu H urči minimální hranový řez S_i oddělující vrcholy x a y .
 - d. Vytvoř hranu e_i stromu (T, w) , jejíž jeden koncový vrchol bude představovat množinu $U \cap V(X_1)$ a druhý množinu $U \cap V(X_2)$, kde X_1 je komponenta řezu S_i obsahující x a X_2 je komponenta řezu S_i obsahující y . Každou hranu spojující nějaký vrchol v s vrcholem u ve stromu (T, w) ved' z vrcholu v (místo do vrcholu u) do vrcholu odpovídajícímu $U \cap V(X_1)$, pokud $b_j \in X_1$, nebo do vrcholu odpovídajícímu $U \cap V(X_2)$, pokud $b_j \in X_2$, kde je B_j komponenta obsahující všechny vrcholy odpovídající vrcholu v . Odstraň vrchol u a polož $w(e_i) := |S_i|$.
 - e. Jdi na 3.
- 4) Vrať (T, w) .

Vrcholy postupně se rozrůstajícího stromu v algoritmu 3.1.3 reprezentují navzájem disjunktní množiny vrcholů vstupního grafu G (sjednocením těchto množin je $V(G)$). V prvním kroku algoritmu se najde hranový řez oddělující dva libovolné vrcholy vstupního grafu a vytvoří se první hrana Gomoryho-Huova stromu. Až bude každý vrchol reprezentovat jednoprvkovou množinu vrcholů grafu G , tj. jeden vrchol grafu G ,

algoritmus skončí a výsledný strom bude Gomoryho-Huův strom grafu G (aby platilo $V(T) = V(G)$), je třeba ještě drobná změna: ve výsledném stromu zaměnit každou množinu $\{v\}$ za vrchol v). Dokud však bude ve stromu vrchol u reprezentující alespoň dvouprvkovou množinu U , najde se minimální hranový řez (tj. s minimální velikostí) oddělující dva libovolně vybrané vrcholy x a y množiny U , přičemž se využije věty 3.1.1 pro stažení komponent grafu $G - U$, a vrchol u se nahradí dvěma novými vrcholy (přestavujícími dvě podmnožiny množiny U , na které množinu U rozdělí nalezený hranový řez oddělující x a y). Mezi nově vzniklými vrcholy stromu povede hrana reprezentující nalezený hranový řez mezi x a y a ohodnocená jeho velikostí. Hrany incidentní s vrcholem u budou nyní vést mezi svým původním uzlem různým od u a jedním z nových uzlů podle toho, k jaké ze dvou částí grafu, na které graf G rozdělí hranový řez mezi x a y , budou patřit.

Při hledání minimálních hranových řezů v algoritmu 3.1.3 lze použít algoritmus 3.1.1. Algoritmus 3.1.3 má stejnou časovou složitost jako algoritmus 3.1.2, jelikož se v něm spočítá $n - 1$ lokálních hranových souvislostí (n je řád vstupního grafu). Jeho výhoda spočívá ve vytvoření Gomoryho-Huova stromu, z kterého lze vyčíst v lineárním čase jak hranovou souvislost vstupního grafu, tak libovolnou lokální hranovou souvislost vstupního grafu včetně příslušných hranových řezů.

3.2 Algoritmus pro určení hranové souvislosti grafu pomocí platného uspořádání vrcholů

Pro určení hranové souvislosti grafu lze použít algoritmus využívající platné uspořádání vrcholů autorů Nagamochiho a Ibarakiho [5].

Definice 3.2.1 [5]

Platné uspořádání vrcholů grafu G s $v(G) \geq 2$ je uspořádání jeho vrcholů v posloupnost (v_1, \dots, v_n) , pro kterou platí $(\forall i \in \mathbb{N})(\forall j \in \mathbb{N}) 2 \leq i \leq j \leq n \Rightarrow e[\{v_i\}, \{v_1, \dots, v_{i-1}\}] \geq e[\{v_j\}, \{v_1, \dots, v_{i-1}\}]$.

Věta 3.2.1 [5]

Pro každý graf G řádu alespoň dva s platným uspořádáním (v_1, \dots, v_n) platí $c'(v_n, v_{n-1}) = e[\{v_n\}, \{v_1, \dots, v_{n-1}\}]$.

Poznamenejme, že v předchozí větě 3.2.1 platí $e[\{v_n\}, \{v_1, \dots, v_{n-1}\}] = d_G(v_n)$.

Původní algoritmus má na vstupu graf bez smyček s alespoň třemi vrcholy. Odstraněním konečného počtu smyček grafu vznikne graf, který má stejnou hranovou souvislost jako graf původní, a pro graf řádu menšího než tři lze zjistit jeho hranovou souvislost triviálně. Můžeme tedy použít algoritmus s drobnou úpravou, který má na vstupu libovolný graf. Obdobným způsobem lze upravovat i jiné grafové algoritmy.

Algoritmus 3.2.1 (Hranová souvislost grafu; Nagamochi a Ibaraki) [5]

- Vstup: Graf G .
 - Výstup: $\kappa'(G)$.
- 1) Vytvoř z grafu G odstraněním všech smyček grafu G graf H . Polož $G := H$.
 - 2) Pokud $v(G) = 1$, polož $\kappa'(G) := 1$.
 - 3) Pokud $v(G) \geq 2$, polož $n := v(G)$, $G_1 := G$ a pak:
 - a. Udělej pro $i = 1, \dots, n - 1$:
 - i. Vytvoř platné uspořádání vrcholů grafu G_i , zapiš do seznamu jeho poslední vrchol x_i a $d_G(x_i)$.
 - ii. Pokud $i \leq n - 2$, vytvoř graf G_{i+1} stažením posledních dvou vrcholů platného uspořádání vrcholů grafu G_i .
 - b. Polož $\kappa'(G) := \min\{d_G(x_i)\}$.
 - 4) Vrať $\kappa'(G)$.

Pokud pro dva navzájem různé vrcholy x a y vstupního grafu G algoritmu 3.2.1 platí $d_G(x) = c'(x, y)$, pak minimální hranový řez R grafu G (tj. hranový řez oddělující dva vrcholy grafu G velikosti $\kappa'(G)$) buď odděluje vrcholy x a y , a tedy $\kappa'(G) = d_G(x)$, nebo stažení vrcholů x a y nezmění R . V i -té iteraci třetího kroku algoritmu podmínka $d_G(x_i) = c'(x_i, y_i)$ pro poslední dva vrcholy y_i a x_i platného uspořádání vrcholů grafu G_i platí podle věty 3.2.1, a stačí tedy uložit do seznamu $d_G(x_i)$ pro možnost platnosti prvního případu a stáhnout vrcholy x_i a y_i pro možnost platnosti druhého případu. Po $n - 1$ iteracích třetího kroku algoritmu získáme velikost minimálního hranového řezu R , a tedy $\kappa'(G)$. Algoritmus 3.2.1 patří do $O(mn)$, kde n je řád a m velikost vstupního grafu, jelikož platné uspořádání vrcholů grafu velikosti m lze nalézt v čase $O(m)$ a v algoritmu se platné uspořádání hledá $(n - 1)$ -krát [5].

Oproti algoritmu 3.1.3 má algoritmus 3.2.1 tedy lepší časovou složitost, ale jeho výstupem je pouze hranová souvislost vstupního grafu G a nikoliv Gomoryho-Huův strom poskytující více informací o grafu G .

Existuje i algoritmus pro určení hranové souvislosti grafu G v čase $O\left(m + \kappa'(G)^2 n \log \frac{n}{\kappa'(G)}\right)$ H. Gabowa [3]. Tento algoritmus je založen na využití matroidů a byl publikován v 90. letech 20. století [3]. Algoritmus 3.2.1 mu však konkuruje větší jednoduchostí a přitom také ještě velmi dobrou časovou složitostí.

3.3 Algoritmus pro testování hranové 2-souvislosti grafu

V praxi je třeba často zjistit, zda je graf G hranově 2-souvislý, aniž by bylo nutné zjišťovat jeho hranovou souvislost $\kappa'(G)$. V algoritmu 3.3.3 se zjistí, zda je vstupní graf 2-souvislý, přičemž časová složitost algoritmu 3.3.3 je menší než časová složitost algoritmu 3.2.1 pro určení hranové souvislosti grafu.

V algoritmu 3.3.3 se používá prohledávání grafu. Prohledávání grafu se provádí podle následujícího algoritmu 3.3.1.

Algoritmus 3.3.1 (Prohledávání grafu) [6]

- Vstup: graf G .
 - Výstup: prohledaný graf G .
- 1) Polož $N := V(G)$, $M := E(G)$ a $D := \emptyset$.
 - 2) Pokud $N \neq \emptyset$, pak:
 - a. Vyber vrchol v z N a polož $N := N \setminus \{v\}$, $D := \{v\}$.
 - b. Vyber vrchol w z D .
 - c. Pokud existuje hrana v M incidentní s vrcholem w , pak:
 - i. Vyber hranu e z M incidentní s vrcholem w a polož $M := M \setminus \{e\}$.
 - ii. Pokud pro koncový vrchol z hrany e různý od w platí $z \in N$, polož $N := N \setminus \{z\}$ a $D := D \cup \{z\}$.
 - iii. Jdi na 2c.
 - d. Polož $D := D \setminus \{w\}$, a platí-li $D \neq \emptyset$, jdi na 2b.
 - e. Jdi na 2.
 - 3) Vrať prohledaný graf G .

V algoritmu 3.3.1 se pouze projdou (nalézají) všechny vrcholy a hrany vstupního grafu. Množina N obsahuje dosud nenalezené vrcholy, množina M obsahuje dosud nenalezené hrany a množina D obsahuje nalezené vrcholy, z nichž ještě může vést nenalezená hrana. Do D se postupně přidávají nově nalezené vrcholy. Z množiny D se vždy vybere jeden vrchol w (pokud není prázdná), prohledají se všechny hrany z M incidentní s w a následně se vrchol w z D odstraní. Existují dvě významné varianty algoritmu 3.3.1 podle toho, jakým způsobem se vybírají prvky v kroku 2b. Pokud se vždy vybere prvek, který je v D nejdelší dobu (tj. D je fronta), jedná se o prohledávání do šířky (anglicky Breadth-First Search, BFS). Pokud se vždy vybere prvek, který je v D nejkratší dobu (tj. D je zásobník), jedná se o prohledávání do hloubky (anglicky Depth-First Search, DFS). Při prohledávání vytvářejí nalezené hrany incidentní s dosud nenalezeným vrcholem (tj. hrany splňující podmínku kroku 2cii) postupně tzv. strom prohledávání (resp. les prohledávání v případě nesouvislého vstupního grafu). Nalezené hrany nepatřící do stromu (resp. lesa) prohledávání se nazývají chordy. Algoritmus lze tedy upravit např. tak, že se v kroku 2cii bude ukládat strom (resp. les) prohledávání, který pak může být na výstupu algoritmu. Tento strom (les) je zároveň kostrou vstupního grafu, a získá se tak tedy jednoduchou úpravou algoritmu 3.3.1 algoritmus pro nalezení kostry grafu. Prohledávání grafu je v praxi často používané a podobných příkladů jeho uplatnění v jiných algoritmech je mnoho. Lze ho tedy považovat za jakýsi základ mnoha algoritmů, např. i algoritmu 3.3.3. Strom (resp. les) prohledávání do šířky se nazývá též BFS-strom (BFS-les) a strom (resp. les) prohledávání do hloubky se nazývá též DFS-strom (DFS-les). Pokud očíslováme vrcholy vstupního grafu podle pořadí, v jakém byly v algoritmu nalezeny, nazývají se tato čísla BFS-indexy (BFI) vrcholů v případě BFS a DFS-indexy (DFI) vrcholů v případě DFS. Vrchol BFS-stromu (lesa), resp. DFS-stromu (lesa), s nejnižším BFI, resp. DFI, se nazývá kořen BFS-stromu (BFS-lesa), resp. kořen DFS-stromu (DFS-lesa). Algoritmus 3.3.1 patří do $O(m + n)$, kde n je řád a m velikost vstupního grafu, avšak jeho různé modifikace mohou mít samozřejmě vyšší časovou složitost.

V algoritmu 3.3.3 se také používá tzv. řetězcový rozklad grafu. Řetězcový rozklad souvislého grafu G bez smyček je množina C jeho podgrafů, tzv. řetězců, která je výsledkem následujícího algoritmu 3.3.2.

Algoritmus 3.3.2 (Řetězcový rozklad grafu) [7]

- Vstup: souvislý graf G bez smyček.
- Výstup: řetězcový rozklad C grafu G .

1) Vytvoř DFS-strom T grafu G .

2) Vytvoř orientaci \vec{G} grafu G tak, že každá hrana grafu G obsažená ve stromu T bude nahrazena orientovanou hranou vedoucí z vrcholu s vyšším DFI do vrcholu s nižším DFI a každá jiná hrana (chorda) bude nahrazena orientovanou hranou vedoucí z vrcholu s nižším DFI do vrcholu s vyšším DFI. Polož $G := \vec{G}$.

3) Každý vrchol grafu G označ jako nenavštívený, polož $C := \emptyset$ a $i := 0$.

4) Pro každý vrchol v grafu G v pořadí daném DFI, od nejnižšího DFI, udělej:

a. Pokud existuje chorda vstupního grafu, jejíž orientace v grafu G vede z vrcholu v , nepatřící do žádného grafu v C , polož $i := i + 1$ a pak udělej:

i. Vyber hranu grafu G , jež je orientací některé chordy h vstupního grafu nepatřící do žádného grafu v C , vedoucí z vrcholu v , jejíž koncový vrchol má nejnižší DFI. Inicializuj prázdný graf C_i , přidej do něj hranu h (tj. polož $C_i := C_i \cup h$, kde h chápeme jako podgraf vstupního grafu obsahující pouze hranu h) a označ vrchol v jako navštívený.

ii. Pokud je koncový vrchol w naposledy přidané hrany do C_i označený jako nenavštívený, označ w jako navštívený, přidej do C_i základní graf hrany grafu G (chápané zde opět jako podgraf grafu G) s počátečním vrcholem w a koncovým vrcholem s nižším DFI než w a jdi na 4aii.

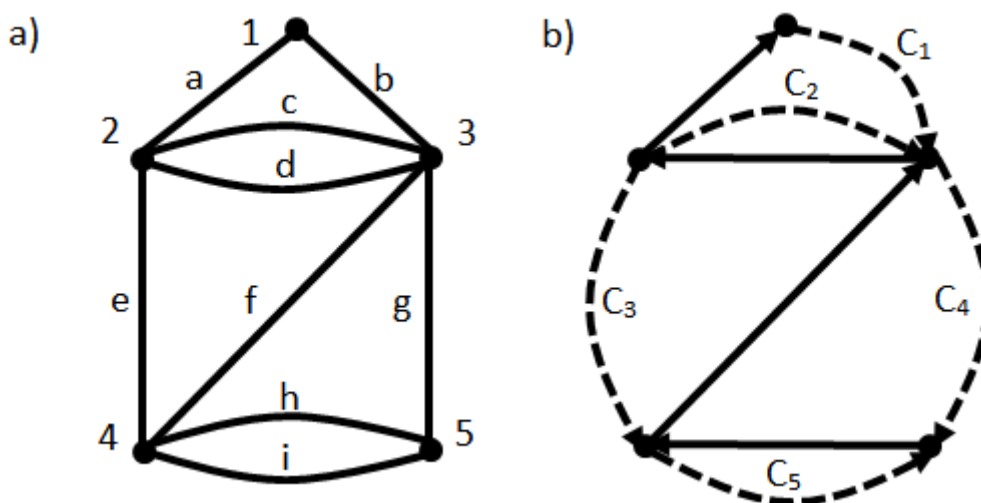
iii. Polož $C := C \cup \{C_i\}$ a jdi na 4a.

5) Vrať řetězcový rozklad C .

V prvním kroku předchozího algoritmu 3.3.2 se využije prohledávání do hloubky podle algoritmu 3.3.1, jehož výstupem bude DFS-strom a DFI každého vrcholu vstupního grafu. Algoritmus 3.3.2 patří do $O(m)$ [7].

Na obr. 3.3.1b je znázorněn řetězcový rozklad grafu G znázorněného na obr. 3.3.1a, přičemž jsou vrcholy grafu G očíslovány podle svého DFI a jsou vyznačeny

orientované hrany podle algoritmu 3.3.2. Platí $C = \cup_{j=1}^5 \{C_j\}$, kde $E(C_1) = \{b, d, a\}$, $E(C_2) = \{c\}$, $E(C_3) = \{e, f\}$, $E(C_4) = \{g, h\}$ a $E(C_5) = \{i\}$.



Obr. 3.3.1: a) Graf G , b) Řetězcový rozklad grafu G

Věta 3.3.1 [7]

Všechny prvky řetězcového rozkladu C grafu G jsou cesty nebo kružnice. Pokud je C neprázdný, pak obsahuje $e(G) - v(G) + 1$ řetězců a jeho první prvek C_1 je kružnice.

Ve shodě s předchozí větou 3.3.1 obsahuje řetězcový rozklad C grafu znázorněného na obr. 3.3.1a pět řetězců ($e(G) = 9$, $v(G) = 5$ a $9 - 5 + 1 = 5$) a řetězec C_1 je kružnice.

Následující věty 3.3.2 a 3.3.3 se použijí v algoritmu 3.3.3.

Věta 3.3.2 [7]

Nechť je C řetězcový rozklad prostého souvislého grafu G s $v(G) \geq 3$. G je hranově 2-souvislý, právě když C obsahuje všechny hrany grafu G .

Věta 3.3.3 [7]

Nechť je C řetězcový rozklad prostého hranově 2-souvislého grafu G s $v(G) \geq 3$. Graf G je 2-souvislý, právě když C obsahuje právě jeden cyklus C_1 .

Následující algoritmus 3.3.3 testuje jak hranovou 2-souvislost, tak i vrcholovou 2-souvislost vstupního grafu.

Algoritmus 3.3.3 (2-souvislost a hranová 2-souvislost; Schmidt) [7]

- Vstup: prostý souvislý graf G řádu alespoň tři.
 - Výstup: Odpověď na otázku „Je G 2-souvislý?“ a na otázku „Je G hranově 2-souvislý?“.
- 1) Vytvoř DFS-strom T grafu G .
 - 2) Vytvoř na základě stromu T řetězcový rozklad C grafu G a označ všechny hrany obsažené v C .
 - 3) Pokud G obsahuje neoznačenou hranu, vrať „ G není 2-souvislý ani hranově 2-souvislý“, jinak:
 - a. Pokud C obsahuje kromě cyklu C_1 ještě jiný cyklus, pak vrať „ G je hranově 2-souvislý, ale není vrcholově 2-souvislý“, jinak vrať „ G je 2-souvislý i hranově 2-souvislý“.

V kroku 1 algoritmu 3.3.3 se použije prohledávání do hloubky obdobně jako v kroku 1 algoritmu 3.3.2 (místo prvních dvou kroků lze použít algoritmus 3.3.2, ve kterém už je obsaženo vytvoření DFS-stromu). Po vytvoření DFS-stromu grafu G a řetězcového rozkladu grafu G se na základě věty 3.3.2 zjistí, zda je graf hranově 2-souvislý. Pokud graf G není hranově 2-souvislý, není podle věty 2.2.13 ani vrcholově 2-souvislý. Pokud graf G je hranově 2-souvislý, zjistí se na základě věty 3.3.3, zda je i vrcholově 2-souvislý. Algoritmus 3.3.3 patří do $O(m)$, a tedy je časově méně náročný než algoritmus 3.2.2. Pokud tedy nepotřebujeme určit hranovou souvislost grafu, ale stačí nám zjistit, zda je graf hranově 2-souvislý, je vhodnější použít algoritmus 3.3.3 než algoritmus 3.2.2. Algoritmus 3.3.3 lze rozšířit pro libovolný prostý graf tak, že se na začátku algoritmu triviálně vyšetří případ nesouvislého grafu a grafu řádu menšího než tři.

3.4 Algoritmus pro testování hranové 3-souvislosti grafu

Kromě hranové 2-souvislosti je v praxi často ještě třeba testovat hranovou 3-souvislost grafu. V algoritmu 3.4.1 se právě testuje hranová 3-souvislost grafu a opět má tento algoritmus lepší časovou složitost než algoritmus 3.2.1 pro určení hranové souvislosti grafu. Algoritmus 3.4.1 je certifikační, tzn. poskytuje certifikát, podle kterého lze rychle zjistit správnost výsledku algoritmu.

Definice 3.4.1 [8]

Hrana s koncovými vrcholy u a v se značí uv . Orientovaná hrana s počátečním vrcholem u a koncovým vrcholem v se také značí uv . Neorientovaná nebo orientovaná v_1v_n -cesta obsahující právě navzájem různé hrany $v_1v_2, v_2v_3, \dots, v_{n-1}v_n$, kde n označuje délku cesty a $n \geq 1$, se značí $v_1v_2 \dots v_n$. Neorientovaná, resp. orientovaná, vw -cesta P obsažená v neorientovaném, resp. orientovaném, grafu G se značí $v \rightarrow_G w$, počáteční vrchol v cesty P se značí $s(P)$ a koncový vrchol w cesty P se značí $t(P)$. Neorientovaná, resp. orientovaná, vw -cesta P obsažená v neorientované, resp. orientované, kružnici K délky alespoň 2 neorientovaného, resp. orientovaného, grafu G a obsahující všechny její vrcholy rozšířená o hranu wv , tj. graf $Q = P \cup wv$, kde hranu wv chápeme jako podgraf grafu G , se značí $v \rightarrow_G v$ a vrchol v se též značí $s(Q)$ (v tomto případě ho považujeme za počáteční vrchol grafu Q) nebo $t(Q)$ (v tomto případě ho považujeme za koncový vrchol grafu Q). Každý prvek z $V(x \rightarrow_G y) \setminus \{x, y\}$ se nazývá vnitřní vrchol grafu $x \rightarrow_G y$ a o každém prvku z $V(x \rightarrow_G y)$ se říká, že leží na $x \rightarrow_G y$.

Graf Q v předchozí definici 3.4.1 je tedy kružnice se zvoleným počátečním (a zároveň koncovým) vrcholem v a jedná se o rozšíření pojmu cesty P , které využijeme v algoritmu 3.4.1.

Definice 3.4.2 [8]

Nechť T je strom a r pevně zvolený vrchol stromu, který nazýváme kořen r . Pro každé dva vrcholy x a y z $V(T)$ říkáme, že x je předkem y a y je potomkem x , jestliže $x \in V(r \rightarrow_T y)$. Pokud navíc $x \neq y$, nazývá se x vlastním předkem y a y vlastním potomkem x . Pokud je x předkem y , resp. vlastním předkem y , píšeme $x < y$, resp. $x \leq y$. Pokud je x vlastním předkem y a neexistuje žádný vrchol, který by byl vlastním předkem y a zároveň vlastním potomkem x , nazývá se y dítětem x a x se nazývá rodičem y a značí $p(y)$.

Definice 3.4.3 [8]

Graf G bez smyček, pro který platí $v(G) = 2 \wedge e(G) = m$, se značí K_2^m .

Definice 3.4.4 [8]

Přidání hrany uv do grafu G , kde $\{u, v\} \subseteq V(G)$, znamená přidání prvku uv , kde $uv \notin E(G)$, do $E(G)$ a položení $\psi_G(uv) = \{u, v\}$. Přidání vrcholu z , kde $z \notin V(G)$, do grafu G znamená přidání vrcholu z do $V(G)$.

Definice 3.4.5 [8]

Rozdělení hrany uv v grafu G znamená odstranění hrany uv z grafu G a přidání vrcholu z a hran uz a zv do grafu G .

Věta 3.4.1 [8]

Právě každý hranově 3-souvislý graf G , uvažujeme-li pouze grafy bez smyček, je možné zkonstruovat z grafu K_2^3 pomocí pouze následujících operací:

- a) Přidání hrany do grafu.
- b) Rozdělení některé hrany xy grafu a zároveň přidání hrany zw do grafu, kde w je vrchol obsažený v grafu před rozdělením hrany xy a z je vrchol přidáný při rozdělení hrany xy .
- c) Rozdělení dvou navzájem různých hran wx a yz grafu a přidání hrany kl do grafu, kde k a l jsou navzájem různé vrcholy přidáné do grafu při rozdělení hran wx a yz .

Definice 3.4.6 [8]

Rozdělení G' grafu G je graf, který lze zkonstruovat z grafu G pouze pomocí konečného počtu operací rozdělení hrany. Větvicí vrcholy rozdělení G' grafu G jsou vrcholy v s $d_{G'}(v) \geq 3$ a nevětvicí vrcholy rozdělení G' jsou vrcholy v s $d_{G'}(v) \leq 2$. Linka rozdělení G' je každý graf $v \rightarrow_{G'} w$, pro jehož každý vrchol x platí: x není počátečním ani koncovým vrcholem grafu $v \rightarrow_{G'} w$, právě když $d_{G'}(x) = 2$.

Věta 3.4.2 [8]

Právě každé rozdělení G' hranově 3-souvislého grafu G , pokud uvažujeme grafy bez smyček, je možné zkonstruovat z grafu K_2^3 pouze pomocí následujících operací (graf G_i je zde vždy graf před i -tou provedenou operací, graf G_{i+1} je zde vždy graf po provedení i -té operace):

- a) Přidání všech vnitřních vrcholů a hran grafu $v \rightarrow_{G_{i+1}} w$ do G_i , kde v a w jsou větvicí vrcholy grafu G_i a ostatní vrcholy grafu $v \rightarrow_{G_{i+1}} w$ nejsou v $V(G_i)$.
- b) Přidání všech vnitřních vrcholů a hran grafu $v \rightarrow_{G_{i+1}} w$ do G_i , kde v je větvicí a w je nevětvicí vrchol grafu G_i a ostatní vrcholy grafu $v \rightarrow_{G_{i+1}} w$ nejsou v $V(G_i)$.
- c) Přidání všech vnitřních vrcholů a hran grafu $v \rightarrow_{G_{i+1}} w$, kde v a w jsou nevětvicí vrcholy grafu G_i ležící na navzájem různých linkách grafu G_i a ostatní vrcholy grafu $v \rightarrow_{G_{i+1}} w$ nejsou v $V(G_i)$.

Definice 3.4.7 [8]

Každý graf $v \rightarrow_{G_{i+1}} w$ v předchozí větě se nazývá Maderova cesta (vzhledem ke grafu G_i).

Věta 3.4.2 je důsledkem věty 3.4.1. Maderovy cesty se nazývají podle autora věty 3.4.1 W. Madera. Maderovy cesty budou tvořit certifikát algoritmu 3.4.1.

Definice 3.4.8 [8]

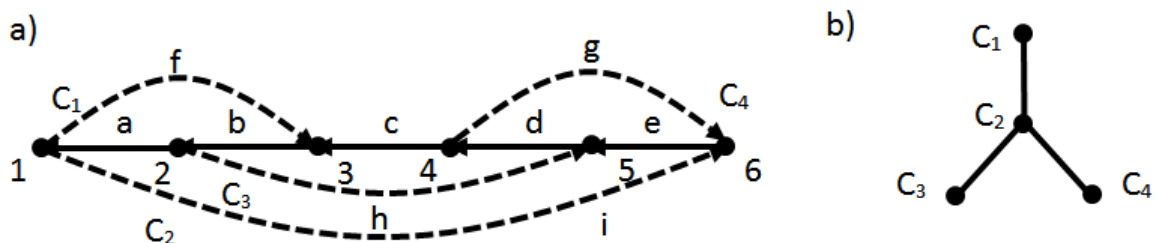
Říkáme, že v řetězcovém rozkladu C hranově 2-souvislého grafu G první vrchol řetězce C_1 (tj. kořen DFS-stromu, na jehož základě byl vytvořen rozklad C) s-náleží (silně náleží) k C_1 a že každý jiný vrchol v grafu G s-náleží (silně náleží) k řetězci C_i , pokud $vp(v) \in E(C_i)$ (kde je zobrazení p dáno DFS-stromem, podle kterého byl vytvořen řetězcový rozklad C).

Definice 3.4.9 [8]

Rodičovský strom T řetězců řetězcového rozkladu C hranově 2-souvislého grafu G je strom, jehož vrcholy jsou právě všechny řetězce z C a pro každé dva vrcholy C_i a C_j stromu T platí: $C_i = p(C_j)$, právě když $t(C_j)$ s-náleží k C_i .

Na obr. 3.4.1b je znázorněn rodičovský strom řetězcového rozkladu grafu znázorněného na obr. 3.4.1a, kde $E(C_1) = \{f, b, a\}$, $E(C_2) = \{i, e, d, c\}$, $E(C_3) = \{h\}$ a $E(C_4) = \{g\}$. Na obr. 3.4.1a jsou vrcholy očíslovány svým DFI a hrany jsou nahrazeny orientovanými hranami podle algoritmu 3.3.2.

Dále budeme pojmy předek, potomek, rodič a dítě řetězců vztahovat k jejich rodičovskému stromu.



Obr. 3.4.1: a) Řetězcový rozklad, b) Rodičovský strom řetězcového rozkladu

Věta 3.4.3 [8]

Nechť je $\cup_{i=1}^{e(G)-v(G)+1} \{C_i\}$ řetězcový rozklad hranově 2-souvislého grafu G a r označuje první vrchol grafu C_1 (tj. kořen DFS-stromu, na jehož základě byl řetězcový rozklad vytvořen). Pak:

- 1) Pro každý řetězec C_i platí $s(C_i) \leq t(C_i)$.
- 2) Pro každé $i \geq 2$ existuje rodič $p(C_i)$ řetězce C_i , přičemž $s(p(C_i)) \leq s(C_i)$ a existuje j menší než i takové, že $p(C_i) = C_j$.
- 3) Pro každé $i \geq 2$: $t(C_i) \neq r \Rightarrow t(p(C_i)) < t(C_i)$ a $t(C_i) = r \Rightarrow t(p(C_i)) = t(C_i)$.
- 4) Jestliže $u \leq v$, u s-náleží k C_i a v s-náleží k C_j , pak $C_i \leq C_j$.
- 5) Jestliže $u \leq t(C_j)$ a u s-náleží k C_i , pak $C_i \leq C_j$.
- 6) Pro každé $i \geq 2$ existuje j menší než i takové, že $s(C_i)$ s-náleží k C_j .

Věta 3.4.4 [8]

Nechť je T DFS-strom grafu G s kořenem r . Jestliže je G hranově 3-souvislý, pak pro každý podstrom S grafu T , jehož množinu vrcholů tvoří právě všichni potomci dítěte s kořene r , existují alespoň dvě hrany z $E(G) \setminus E(T)$ takové, že každá z nich spojuje kořen r s některým vrcholem stromu S .

Na základě věty 3.4.4 probíhá první krok algoritmu 3.4.1. Pokud se zjistí, že existuje pouze jen jedna hrana z $E(G) \setminus E(T)$ spojující kořen r s některým vrcholem stromu S , pak algoritmus vrátí hranový řez tvořený touto hranou a hranou spojující kořen r stromu T s jeho dítětem s příslušným ke stromu S . Pokud se zjistí, že existují alespoň dvě hrany z $E(G) \setminus E(T)$ takové, že každá z nich spojuje kořen r s některým vrcholem stromu S , pak první řetězec řetězcového rozkladu vstupního grafu bude obsahovat jednu z těchto hran a druhý druhou. Sjednocením prvních dvou řetězců řetězcového rozkladu vstupního grafu bude tedy graf $K_2^{3'}$, k němuž se v průběhu algoritmu budou přidávat Maderovy cesty podle věty 3.4.2.

Definice 3.4.10 [8]

Nechť je T rodičovský strom řetězců řetězcového rozkladu C grafu G . Podgraf grafu G , obsahující pouze řetězce z C tak, že obsahuje s každým prvkem C_i , který není kořenem T , i prvek $p(C_i)$, se nazývá rodičovsky uzavřený a značí G_c .

Definice 3.4.11 [1]

Nechť F je podgraf grafu G . Ucho grafu G vzhledem k F je netriviální cesta obsažená v grafu G , jejíž počáteční a koncový vrchol jsou v $V(F)$, ale vnitřní vrcholy nikoliv.

Věta 3.4.5 [8]

Nechť je G_c rodičovsky uzavřený podgraf grafu G obsahující řetězce C_1 a C_2 . Pak platí:

- 1) Pro každý vrchol v z G_c , který není prvním vrcholem v C_1 (tj. kořenem DFS-stromu, na jehož základě byl vytvořen řetězcový rozklad grafu G), je i vrchol $p(v)$ obsažen v G_c .
- 2) Pro každý řetězec obsažený v G_c jsou $s(C)$ a $t(C)$ větvicí vrcholy grafu G_c .
- 3) Nechť je C řetězec, který není obsažený v G_c , ale je dítětem nějakého řetězce v G_c . Pak je C ucho vzhledem k G_c a G_c obsahuje graf $t(C) \rightarrow_T s(C)$. C je Maderova cesta vzhledem k G_c , právě když existuje větvicí vrchol v grafu $t(C) \rightarrow_T s(C)$.

Na základě vět 3.4.4 a 3.4.5 lze dokázat následující větu 3.4.6.

Věta 3.4.6 [8]

Nechť G je graf a G_c je rodičovsky uzavřený podgraf řetězců grafu G takový, že žádné dítě X žádného řetězce obsaženého v G_c není Maderovou cestou vzhledem k G_c a existuje alespoň jeden takový řetězec X . Pak je množina krajních hran každé linky (tj. hran incidentních v lince s vrcholy řádu většího než dva) délky alespoň dva v G_c hranovým řezem velikosti 2 grafu G .

Důsledkem věty 3.4.6 je následující věta 3.4.7.

Věta 3.4.7 [8]

Jestliže je graf G hranově 3-souvislý, lze ho sestavit postupným přidáváním řetězců jeho řetězcového rozkladu tak, že po každém přidání řetězce je vzniklý graf rodičovsky uzavřený.

Správné pořadí přidávaných řetězců podle věty 3.4.7 lze najít v čase $O((n + m) \log(n + m))$ [8].

Definice 3.4.12 [8]

Nechť C je řetězcový rozklad hranově 2-souvislého grafu G . Všechny řetězce C_i z C s $i \geq 3$ rozdělujeme na prokládané a vnořené:

- a) Proložené jsou ty, jejichž počáteční vrchol je předkem koncového vrcholu jejich rodiče.
- b) Vnořené jsou ostatní, tj. jejichž počáteční vrchol je vlastním potomkem koncového vrcholu jejich rodiče.

Věta 3.4.8 [8]

Nechť G je graf a G_c je rodičovsky uzavřený podgraf grafu G obsahující C_1 a C_2 . Nechť C je řetězec obsažený v G_c a D je proložené dítě řetězce C , které není obsaženo v G_c . Pak je D Maderovou cestou vzhledem k G_c .

Uvedme nyní algoritmus 3.4.2. Níže v textu bude ještě vysvětleno, jak se řetězce v algoritmu přiřazují do jednotlivých segmentů.

Algoritmus 3.4.1 (Hranová 3-souvislost; Mehlhorn, Neumann a Schmidt) [8]

- Vstup: Graf G neobsahující smyčky, pro nějž platí $\kappa'(G) \geq 2 \wedge \delta(G) \geq 3$.
 - Výstup: Pokud platí $\kappa'(G) = 2$, pak hranový $\partial(X)$ velikosti 2 grafu G . Pokud platí $\kappa'(G) \geq 3$, pak posloupnost Maderových cest grafu G .
- 1) Vytvoř řetězcový rozklad C grafu G , kde $C = \bigcup_{i=1}^{e(G)-v(G)+1} \{C_i\}$, a polož $G_c := C_1 \cup C_2$.
 - 2) Pro $i = 1 \dots e(G) - v(G) + 1$ udělej:
 - a. Každý řetězec A z C takový, že $s(A)$ s-náleží do C_i , přiřaď do příslušného segmentu.
 - b. Do G_c přidej každý řetězec každého segmentu, jehož minimální řetězec (tj. řetězec segmentu, z něhož vede do C_1 v rodičovském stromu cesta nejmenší délky) je proložený.
 - c. Hledej vhodné uspořádání všech segmentů S_1, \dots, S_k , z nichž každý má minimální řetězec (tj. řetězec segmentu, z něhož vede do C_1 v rodičovském stromu cesta nejmenší délky) vnořený. Pokud ho nenajdeš, najdi a vrať hranový $\partial(X)$ velikosti 2 grafu G a skonči. Jinak pro j od 1 do k přidej do G_c každý řetězec z S_j při zachování rodičovské uzavřenosti grafu G_c .

3) Vrať G_c i s pořadím přidaných řetězců (Maderových cest) do G_c .

Vstupem algoritmu je hranově 2-souvislý graf G , pro který platí $\delta(G) \geq 3$. Na začátku algoritmu se vytvoří řetězcový rozklad grafu G podle algoritmu 3.3.2 a inicializuje se rodičovsky uzavřený graf G_c jako $C_1 \cup C_2$. Následuje druhý krok algoritmu, kde se pro každé přirozené i od 1 do $e(G) - v(G) + 1$ buď přidá do grafu G_c každý řetězec A takový, že $s(A)$ s-náleží do C_i , nebo algoritmus vrátí hranový řez velikosti 2 grafu G a skončí. Podle vět 3.4.7 a 3.4.6 lze vstupní graf sestavit z řetězců jeho řetězcového rozkladu jako z Maderových cest, pokud je vstupní graf hranově 3-souvislý. Dále naznačíme, jak se toto přidávání řetězců provádí pomocí rozdělení řetězců do segmentů. Nejdříve je třeba pro lepší pochopení průběhu algoritmu uvést věty 3.4.9 a 3.4.10.

Věta 3.4.9 [8]

V i -té iteraci kroku 2 algoritmu 3.4.1 je řetězec C_i součástí grafu G_c na začátku i -té iterace kroku 2, nebo algoritmus vrátí hranový řez velikosti 2 před i -tou iterací kroku 2.

Věta 3.4.10 [8]

Nechť je algoritmus 3.4.1 v i -té iteraci kroku 2 a dosud nevrátil hranový řez velikosti 2. Nechť je A , kde $A \neq C_2$, řetězec, který s-náleží do C_i . A není částí grafu G_c na začátku i -té iterace kroku 2. Nechť B je předek řetězce A takový, že B není obsažen v G_c . Pak platí:

- 1) Vrchol $s(B)$ s-náleží k C_i .
- 2) Jestliže je B vnořený řetězec, pak je B dítětem řetězce C_i .
- 3) Jestliže není $p(B)$ částí grafu G_c , pak je B proložený řetězec.

Definice 3.4.13 [8]

Nechť je S_i množina všech řetězců A vstupního grafu G takových, že $s(A)$ s-náleží do C_i . Nechť pro řetězec A z S_i označuje A^* minimálního předka řetězce A , který není v G_c na začátku i -té iterace kroku 2 algoritmu (tj. předka této vlastnosti, z kterého vede cesta do C_1 v rodičovském stromu nejmenší délky). Rozdělení do segmentů v i -té iteraci kroku 2 algoritmu se provádí tak, že dva prvky D, E z S_i patří do stejného segmentu S , právě když $D^* = E^*$.

Nyní tedy na základě definice 3.4.13 víme, jak se v kroku 2a algoritmu 3.4.1 rozdělují řetězce do segmentů.

Věta 3.4.11 [8]

Každý segment v algoritmu 3.4.1 odpovídá nějakému podstromu rodičovského stromu řetězců z C .

Věta 3.4.12 [8]

V i -té iteraci kroku 2 algoritmu 3.4.1 mohou být přidány všechny řetězce ze segmentu S do G_c při uchování rodičovské uzavřenosti grafu G_c , jestliže je možné přidat minimální řetězec segmentu (tj. řetězec segmentu, z kterého vede do C_1 v rodičovském stromu cesta nejmenší délky) do G_c .

Podle vět 3.4.8 a 3.4.12 je možné v i -té iteraci kroku 2 algoritmu 3.4.1 přidat do G_c všechny řetězce ze segmentu, jehož minimální řetězec je proložený. Dále je ještě třeba vysvětlit, jak se přidávají řetězce ze segmentu, jehož minimální řetězec je vnořený.

Definice 3.4.14 [8]

Nechť S je segment v i -té iteraci kroku 2 algoritmu 3.4.1, jehož minimální řetězec (tj. řetězec segmentu, z kterého vede do C_1 v rodičovském stromu cesta nejmenší délky) je vnořený. Přichycený bod segmentu S je takový vrchol segmentu, který je obsažen v G_c v i -té iteraci kroku 2) algoritmu.

Věta 3.4.13 [8]

Nechť je C vnořené dítě řetězce C_i a necht' je S segment i -té iterace kroku 2 algoritmu 3.4.1, jehož minimální řetězec (tj. řetězec segmentu, z kterého vede do C_1 v rodičovském stromu cesta nejmenší délky) je vnořený, obsahující C . Pak všechny přichycené body segmentu S leží v grafu $t(C) \rightarrow_T s(C)$, a tedy i v C_i .

Na základě věty 3.4.13 se definuje tzv. překrývající graf v následující definici 3.4.15.

Definice 3.4.15 [8]

Překrývající graf H segmentů s vnořeným minimálním řetězcem (tj. řetězcem segmentu, z kterého vede do C_1 v rodičovském stromu cesta nejmenší délky) v i -té

iteraci kroku 2 algoritmu 3.4.1 je graf, jehož vrcholy jsou právě všechny tyto segmenty a navíc speciální vrchol R a pro jehož hrany platí:

- a) Existuje hrana spojující vrchol R a segment S , právě když existují přichycené body a_1, a_2 takové, že $a_1 \leq a_2$ a existuje větvicí vrchol grafu G_c na cestě $a_2 \rightarrow_T a_1$.
- b) Existuje hrana spojující dva navzájem různé segmenty S_k a S_l , právě když existují přichycené body b_1, b_2, c_1, c_2 takové, že $b_1 \in S_k, b_2 \in S_k, c_1 \in S_l, c_2 \in S_l$ a $b_1 \leq c_1 \leq b_2 \leq c_2 \vee c_1 \leq b_1 \leq c_2 \leq b_2$. (Říkáme, že se segmenty S_k a S_l překrývají.)

Věta 3.4.14 [8]

Nechť je Q souvislá komponenta překrývajícího grafu H a S je segment vzhledem k C_i (v i -té iteraci kroku 2 algoritmu 3.4.1) takový, že jeho minimální řetězec (tj. řetězec segmentu, z kterého vede do C_1 v rodičovském stromu cesta nejmenší délky) je vnořený. Pak platí $S \in V(Q)$, právě když nastává alespoň jedna z těchto možností:

- 1) Vrchol R je prvkem $V(Q)$ a existuje větvicí vrchol v grafu $t(C) \rightarrow_T s(C)$.
- 2) Existují přichycené body a_1 a a_2 v segmentu S a přichycené body b_1 a b_2 v segmentech z Q s vlastností $a_1 \leq b_1 \leq a_2 \leq b_2 \vee b_1 \leq a_1 \leq b_2 \leq a_2$.

S pomocí věty 3.4.14 lze dokázat následující větu 3.4.15.

Věta 3.4.15 [8]

Nechť je algoritmus 3.4.1 v i -té iteraci kroku 2. Jestliže je překrývající graf H segmentů vzhledem k C_i souvislý, lze přidat všechny segmenty z C_i do G_c . Pokud H není souvislý, lze vrátit hranový řez velikosti 2 pro každou komponentu grafu H , která neobsahuje vrchol R .

Nyní tedy ještě chybí vytvořit uspořádání segmentů, které mají minimální řetězec vnořený, podle něhož se přidávají řetězce do G_c v kroku 2c algoritmu 3.4.1. Popis konstrukce tohoto uspořádání je v [8]. Pro nás je důležité, že hledání tohoto uspořádání nezvýší časovou složitost algoritmu 3.4.1, která je $O(m + n)$ [8].

Shrňme tedy základní fakta o certifikačním algoritmu 3.4.1. Algoritmus má na vstupu hranově 2-souvislý graf. Pokud vstupní graf není hranově 3-souvislý, vrátí algoritmus jako certifikát hranový řez velikosti 2 vstupního grafu v lineárním čase.

Pokud je vstupní graf hranově 3-souvislý, vrátí algoritmus jako certifikát posloupnost Maderových cest, z nichž lze vytvořit vstupní graf, v lineárním čase.

4 Algoritmus pro určení cyklické hranové souvislosti kubického grafu a jeho implementace

4.1 Algoritmus pro určení cyklické hranové souvislosti kubického grafu

Cyklická hranová souvislost je jednou z variant hranové souvislosti grafu. O cyklické hranové souvislosti se zmiňuje již roku 1880 P. G. Tait [2] a my nyní popíšeme dosud nejlepší známý algoritmus pro její určení.

Definice 4.1.1 [2]

Kubický graf G je graf, pro jehož každý vrchol v platí $d_G(v) = 3$, tj. kubický graf G je 3-regulární graf.

Definice 4.1.2 [2]

Cyklický hranový řez R grafu G je hranový řez grafu G takový, že alespoň dvě komponenty grafu $G \setminus R$ obsahují kružnici. Minimální cyklický hranový řez M grafu G je cyklický hranový řez M grafu G , který má nejmenší velikost.

Všimněme si, že každý minimální cyklický hranový řez M souvislého grafu G rozděluje graf G na právě dvě komponenty grafu $G \setminus M$.

Definice 4.1.3 [2]

Cyklická hranová souvislost grafu G je velikost minimálního cyklického hranového řezu grafu G .

Cyklická hranová souvislost grafu je dalším z parametrů souvislosti grafu. Zvláště u kubických grafů, kde je podle věty 2.2.13 hranová souvislost omezena hodnotou 3, se cyklická hranová souvislost používá pro ještě lepší odlišení grafů podle míry jejich souvislosti. Cyklická hranová souvislost kubického grafu se bude počítat v algoritmu 4.1.1.

Definice 4.1.4 [2]

Nechť je T strom s kořenem r . Nechť d je pro každý vrchol v stromu T délka cesty $v \rightarrow_T r$. Říkáme, že vrcholy, které mají stejné d , tvoří úroveň d (d -tou úroveň) stromu T .

Podle definice 4.1.4 lze tedy všechny vrcholy každého stromu rozdělit do jednotlivých úrovní. Kořen stromu tvoří nultou úroveň, jeho sousedé první úroveň atd.

Definice 4.1.5 [2]

Nechť je v grafu G provedeno BFS. BFS-graf hloubky d grafu G je graf složený ze všech hran uv grafu G takových, že vrchol u patří v BFS-stromu grafu G do úrovně nejvýše $d - 1$ a vrchol v patří v BFS-stromu grafu G do úrovně nejvýše d .

BFS-graf hloubky d je tedy podle definice 4.1.5 podgraf grafu G , který na rozdíl od indukovaného grafu na všech vrcholech BFS-stromu grafu G úrovně nejvýše d neobsahuje hrany spojující navzájem vrcholy úrovně d .

Věta 4.1.1 [2]

Nechť je G kubický graf bez smyček řádu alespoň 8 a nechť je g obvod grafu G . Pak platí $g \leq 2 \left\lceil \log_2 \left(\frac{n}{3} + 1 \right) \right\rceil$ a G má cyklický hranový řez velikosti g .

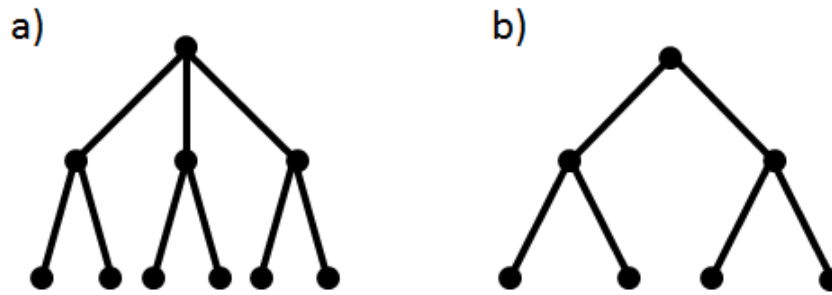
Poznamenejme, že $\lceil x \rceil$ značí horní celou část čísla x , tj. např. $\lceil 2,5 \rceil = 3$ a $\lceil 4 \rceil = 4$.

Podle věty 4.1.1 tedy můžeme u skoro všech kubických grafů omezit shora jejich cyklickou hranovou souvislost jejich obvodem.

Definice 4.1.6 [2]

Plný strom T hloubky d je strom T , jenž obsahuje právě $d + 1$ úrovní (tj. nultou až d -tou úroveň) a jehož kořen má právě tři děti a každý jiný vrchol (kromě vrcholů úrovně d) má právě dvě děti.

Příkladem plného stromu hloubky 2 je graf znázorněný na obr. 4.1.1a. Počet vrcholů k -úrovně netriviálního plného stromu T hloubky d , kde $k > 0$, je $3 * 2^{k-1}$, a tedy strom T obsahuje celkem $3 * 2^d - 2$ vrcholů.



Obr. 4.1.1: a) Plný strom, b) Kompletní binární strom

Definice 4.1.7 [2]

Nechť je F souvislý podgraf grafu G . Aktivita grafu G je součet $\sum_{v \in V(F)} (d_G(v) - d_F(v))$.

Následující věty 4.1.2 a 4.1.3 budou použity v algoritmu 4.1.1.

Věta 4.1.2 [2]

Nechť jsou G_1 a G_2 souvislé podgrafy grafu G (neobsahujícího smyčky), které neobsahují společný vrchol a oba mají aktivitu alespoň $k + 1$. Každý hranový řez R grafu G oddělující grafy G_1 a G_2 (tj. G_1 a G_2 jsou v navzájem různých komponentách grafu $G \setminus R$), který má velikost nejvýše k , je cyklický hranový řez grafu G .

Věta 4.1.3 [2]

Nechť je G kubický graf bez smyček a κ je jeho cyklická hranová souvislost, kde $\kappa \geq 1$. Pak nastává alespoň jedna z následujících možností:

- 1) Graf G obsahuje kružnici délky κ .
- 2) Graf G obsahuje cyklický hranový řez $\partial(X)$ velikosti κ takový, že obě části grafu G , které navzájem odděluje hranový řez $\partial(X)$, tj. $G[X]$ a $G[V(G) \setminus X]$, obsahují plný strom hloubky $\lceil \log_2 \frac{\kappa+1}{3} \rceil$.

Algoritmus 4.1.1 (Cyklická hranová souvislost kubického grafu; Dvořák, Kára, Král' a Pangrác) [2]

- Vstup: kubický graf G bez smyček řádu alespoň 8.
 - Výstup: cyklická hranová souvislost κ a minimální cyklický hranový řez M grafu G .
- 1) Pokud je G nesouvislý, polož $\kappa := 0$, $M := \emptyset$ a jdi na 4.

- 2) Urči obvod g grafu G a polož $\kappa := g$. Řez M inicializuj jako množinu všech hran incidentních s kružnicí délky g v G , ale nepatřících do této kružnice.
- 3) Pro každý vrchol v grafu G udělej:
- a. Pro každý vrchol w grafu G udělej:
 - i. Polož $d := -1$ a $P := \emptyset$.
 - ii. Polož $d := d + 1$.
 - iii. Vytvoř plný strom Tv hloubky d s kořenem v a plný strom Tw hloubky d s kořenem w v grafu G .
 - iv. Pokud obsahují Tw a Tv společný vrchol, pak jdi na další dvojici vrcholů v a w (krok 3, resp. krok 3a). (Pokud už byly všechny dvojice v a w probrány, tak jdi na 4.)
 - v. Najdi největší množinu Q hranově disjunktních cest vedoucích mezi Tw a Tv (tj. uv -cest, kde $u \in V(Tv) \wedge v \in V(Tw)$ a žádné vnitřní vrcholy uv -cest nejsou v $V(Tv)$ ani v $V(Tw)$) a polož $P := Q$.
 - vi. Pokud platí $|P| < 3 * 2^d \wedge |P| < \kappa$, pak polož $\kappa := |P|$, najdi minimální hranový řez R oddělující Tv a Tw (tj. Tv a Tw jsou v navzájem různých komponentách grafu $G \setminus R$) a polož $M := R$.
 - vii. Pokud platí $3 * 2^d < \kappa$, jdi na 2aii. Jinak jdi na další dvojici vrcholů v a w (krok 3, resp. krok 3a). (Pokud už byly všechny dvojice v a w probrány, tak jdi na 4.)
- 4) Vrať κ a M .

V prvním kroku algoritmu 4.4.1 se zjistí, zda je vstupní graf G souvislý, což lze pomocí BFS zjistit v čase $O(n)$, kde n je řád grafu G (pro velikost m a řád n kubického grafu totiž platí $m = \frac{3n}{2}$). Pokud je G nesouvislý, algoritmus vrátí jeho cyklickou hranovou souvislost (která je u každého nesouvislého kubického grafu rovna nule), minimální cyklický hranový řez (prázdnou množinu) a skončí. Pokud je G souvislý, spočítá se nejdříve obvod grafu G , což je horní mez jeho cyklické hranové souvislosti podle věty 4.1.1. Obvod grafu lze spočítat pomocí BFS podle algoritmu 3.3.1 s časovou složitostí $O(n)$, který se aplikuje postupně na všechny vrcholy grafu G (vždy se hledá kružnice nejmenší délky obsahující daný vrchol), takže časová složitost kroku 2 algoritmu 4.4.1 je $O(n^2)$. Cyklický hranový řez M bude nejdříve obsahovat hrany

incidentní s nalezenou kružnicí nejmenší délky v grafu G , ale neobsažené v této kružnici, a v dalších krocích se bude postupně měnit, až se z něj stane minimální cyklický hranový řez grafu G . V kroku tři se postupně vytvářejí pro každou dvojici vrcholů v a w grafu G plné stromy Tv (s kořenem v) a Tw (s kořenem w) hloubky 0 až $\lceil \log_2 \frac{\kappa}{3} \rceil$. Pokud stromy Tv a Tw (pro konkrétní dvojici vrcholů v a w a hloubku stromů d) neobsahují společný vrchol, hledá se největší počet hranově disjunktních cest vedoucích mezi Tw a Tv (tj. uv -cest, kde $u \in V(Tv) \wedge v \in V(Tw)$ a žádné vnitřní vrcholy uv -cest nejsou v $V(Tv)$ ani v $V(Tw)$), které se uloží do množiny Q (krok 3av). Tyto cesty určují hranový řez R oddělující stromy Tv a Tw (který má tedy velikost $|Q|$). Pokud je velikost hranového řezu R menší než $3 * 2^d$ (kde d je hloubka stromů Tv a Tw), pak se jedná podle věty 3.4.2 o cyklický hranový řez, jelikož každý plný strom (v kubickém grafu) hloubky d má aktivitu $3 * 2^d$. Pokud navíc platí $|R| < |M|$, nahradí se cyklický hranový řez M cyklickým hranovým řezem R a aktualizuje se κ (uchovávaná velikost cyklického hranového řezu M).

Pokud je cyklická hranová souvislost κ vstupního grafu G algoritmu 4.4.1 rovna nule, zjistí se to v kroku 1 algoritmu. Pokud je κ větší než nula, použije se věta 4.1.3. Pokud G obsahuje kružnici délky κ (možnost 1 věty 4.1.3), najde se tato kružnice (určující minimální cyklický hranový řez a cyklickou hranovou souvislost grafu G) při počítání obvodu grafu ve druhém kroku algoritmu (je to kružnice v G s nejmenší délkou). Pokud G neobsahuje kružnici délky κ , pak podle věty 4.1.3 existuje v grafu G cyklický hranový řez $\partial(X)$ velikosti κ takový, že obě části grafu G , které navzájem odděluje hranový řez $\partial(X)$, tj. $G[X]$ a $G[V(G) \setminus X]$, obsahují plný strom hloubky $\lceil \log_2 \frac{\kappa+1}{3} \rceil$. Označme tyto dva stromy T_1 a T_2 . Při procházení jednotlivých hranových řezů oddělujících stromy Tv a Tw hloubky $\lceil \log_2 \frac{\kappa+1}{3} \rceil$ se tedy jednou najde hranový řez oddělující T_1 a T_2 velikosti nejvýše κ a podle věty 3.4.2 je tento hranový řez cyklický. Algoritmus 4.4.1 tedy opravdu po konečném počtu kroků vrátí cyklickou hranovou souvislost a minimální cyklický hranový řez vstupního grafu G .

Hledání minimálních hranových řezů mezi stromy Tv a Tw grafu G ve třetím kroku algoritmu 4.4.1 se provádí tak, že se nejdříve všechny vrcholy stromu Tv stáhnou do jednoho vrcholu x a všechny vrcholy stromu Tw se stáhnou do druhého vrcholu y a pak se na vzniklý graf použije Fordův-Fulkersonův algoritmus 2.2.2. Pokud si uvědomíme, že algoritmus 2.2.2 pouze zvyšuje hodnotu toku, každé zvýšení probíhá

v čase $O(n)$ (n je řád grafu G) a maximální počet navzájem hranově disjunktních cest (zvyšování toků) mezi Tv a Tw hloubky d je $3 * 2^d$, kde d je omezeno obvodem grafu G , který je podle věty 4.1.1 přibližně roven $\log_2 n$, pak se pro každou dvojici vrcholů provede přibližně $\log_2 n$ navýšení toků, kde má každé navýšení časovou složitost $O(n)$. Pro každou dvojici vrcholů tedy algoritmus 4.4.1 pracuje v čase $O(n \log n)$ a celková časová složitost algoritmu 4.4.1 je tedy $O(n^3 \log n)$.

Pro vstupní graf G řádu alespoň 243 lze použít pro určení cyklické hranové souvislosti grafu G následující algoritmus 4.4.2, který má lepší časovou složitost než algoritmus 4.4.1.

Věta 4.1.4 [2]

Nechť je G kubický graf řádu n s obvodem g , kde $n \geq 243 \wedge g \geq 5$. Pak G obsahuje 12 hranově disjunktních (neobsahujících společné hrany) plných stromů hloubky dva. Tyto stromy lze najít v čase $O(n)$.

Definice 4.1.8 [2]

Kompletní binární strom T hloubky d je strom T , jenž obsahuje právě $d + 1$ úrovní (tj. nultou až d -tou úroveň) a jehož každý vrchol (kromě vrcholů úrovně d) má právě dvě děti.

Příkladem kompletního binárního stromu hloubky 2 je graf znázorněný na obr. 4.1.1b. Počet vrcholů k -úrovně kompletního binárního stromu T hloubky d je 2^k , a tedy strom T obsahuje celkem $2^{d+1} - 1$ vrcholů.

Věta 4.1.5 [2]

Nechť je G kubický graf s obvodem g , kde $g \geq 13$. G obsahuje nejméně g hranově disjunktních (neobsahujících společné hrany) kompletních binárních stromů hloubky $\lceil \log_2 \frac{g-1}{2} \rceil$. Tyto binární stromy lze najít v čase $O(n)$.

Algoritmus 4.1.2 (Cyklická hranová souvislost kubického grafu; Dvořák, Kára, Král' a Pangrác) [2]

- Vstup: kubický graf G bez smyček řádu alespoň 243.
- Výstup: cyklická hranová souvislost κ a minimální cyklický hranový řez M grafu G .

1) Pokud je G nesouvislý, polož $\kappa := 0$, $M := \emptyset$ a jdi na 7.

- 2) Urči obvod g grafu G a polož $\kappa := g$. Řez M inicializuj jako množinu hran incidentních s kružnicí délky g v G , ale nepatřících do této kružnice.
- 3) Pokud je $2 \leq g \leq 4$, vytvoř množinu A_0 g navzájem různých hran grafu G (chápaných jako podgrafy grafu G).
- 4) Pokud je $5 \leq g \leq 12$, vytvoř množinu A_0 g hranově disjunktních plných stromů grafu G hloubky 2.
- 5) Pokud je $g \geq 13$, vytvoř množinu A_0 g hranově disjunktních kompletních binárních stromů grafu G hloubky $\lceil \log_2 \frac{g-1}{2} \rceil$.
- 6) Pro každý vrchol v grafu G udělej:
 - a. Pro každý prvek A z A_0 udělej:
 - i. Polož $d := -1$ a $P := \emptyset$.
 - ii. Polož $d := d + 1$.
 - iii. Vytvoř plný strom T hloubky d s kořenem v v grafu G .
 - iv. Pokud obsahují T a A společný vrchol, pak jdi na další dvojici v a A (krok 6, resp. krok 6a). (Pokud už byly všechny dvojice v a A probrány, tak jdi na 7.)
 - v. Najdi největší množinu Q hranově disjunktních cest vedoucích mezi T a A (tj. uv -cest, kde $u \in V(T) \wedge v \in V(A)$ a žádné vnitřní vrcholy uv -cest nejsou v $V(T)$ ani v $V(A)$) a polož $P := Q$.
 - vi. Pokud platí $|P| < 3 * 2^d \wedge |P| < \kappa$, pak polož $\kappa := |P|$, najdi minimální hranový řez R oddělující T a A (tj. T a A jsou v navzájem různých komponentách grafu $G \setminus R$) a polož $M := R$.
 - vii. Pokud platí $3 * 2^d < \kappa$, jdi na 6aii. Jinak jdi na další dvojici v a A (krok 6, resp. krok 6a). (Pokud už byly všechny dvojice v a A probrány, tak jdi na 7.)
- 7) Vrať κ a M .

Algoritmus 4.4.2 se liší od algoritmu 4.4.1 (kromě omezení pro vstupní graf) pouze v tom, že se v něm pro každou dvojici vrcholů místo se dvěma úplnými stromy T_v a T_w pracuje s jedním úplným stromem T a prvkem A z A_0 . Druh prvků A množiny A_0 je určen podle obvodu g grafu G . A_0 vždy obsahuje právě g prvků. Pro $2 \leq g \leq 4$ jsou to navzájem různé hrany grafu G , pro $5 \leq g \leq 12$ navzájem hranově disjunktní plné stromy grafu G hloubky 2 (které se najdou v čase $O(n)$ podle věty 4.1.4) a pro $g \geq 13$

navzájem hranově disjunktí kompletní binární stromy grafu G hloubky $\lceil \log_2 \frac{g-1}{2} \rceil$ (které se najdou v čase $O(n)$ podle věty 4.1.5).

Pokud je cyklická hranová souvislost κ vstupního grafu G algoritmu 4.4.2 rovna nule, zjistí se to v prvním kroku algoritmu, a pokud je κ větší než nula, použije se opět věta 4.1.3. Pokud G obsahuje kružnici délky κ (možnost 1 věty 4.1.3), najde se tato kružnice (určující minimální cyklický hranový řez a cyklickou hranovou souvislost grafu G) při počítání obvodu grafu ve druhém kroku algoritmu (je to kružnice v G s nejmenší délkou). Pokud G neobsahuje kružnici délky κ , pak podle věty 4.1.3 existuje v grafu G cyklický hranový řez $\partial(X)$ velikosti κ takový, že obě části grafu G , které navzájem odděluje hranový řez $\partial(X)$, tj. $G[X]$ a $G[V(G) \setminus X]$, obsahují plný strom hloubky $\lceil \log_2 \frac{\kappa+1}{3} \rceil$. Jeden z grafů $G[X]$ a $G[V(G) \setminus X]$ obsahuje alespoň jeden prvek A množiny A_0 , jelikož existuje prvek A množiny A_0 , který neobsahuje žádnou hranu řezu $\partial(X)$ ($\kappa < g$ a množina A_0 obsahuje g navzájem hranově disjunktích prvků). Označme tento prvek A_1 a bez újmy na obecnosti předpokládejme, že graf $G[X]$ obsahuje A_1 . Plný strom hloubky $\lceil \log_2 \frac{\kappa+1}{3} \rceil$ v grafu $G[V(G) \setminus X]$ označme T_1 . Při procházení jednotlivých hranových řezů oddělujících stromy T hloubky $\lceil \log_2 \frac{\kappa+1}{3} \rceil$ a prvky A z A_0 se tedy najde hranový řez oddělující T_1 a A_1 velikosti nejvýše κ a podle věty 3.4.2 je tento hranový řez cyklický, jelikož každý prvek A z A_0 má aktivitu g ($\kappa < g$). Algoritmus 4.4.2 tedy po konečném počtu kroků vrátí cyklickou hranovou souvislost a minimální cyklický hranový řez vstupního grafu G .

Hledání minimálních hranových řezů mezi stromy T grafu G a prvky A množiny A_0 v šestém kroku algoritmu 4.4.1 se provádí opět tak, že se nejdříve všechny vrcholy stromu T stáhnou do jednoho vrcholu x a všechny vrcholy prvku A se stáhnou do druhého vrcholu y a pak se na vzniklý graf použije Fordův-Fulkersonův algoritmus 2.2.2. Každý prvek A má aktivitu g , a pokud budeme při hledání množiny Q v kroku 6v vycházet z již nalezených cest mezi A a stromem o jedna menší úrovně než T , proběhne navýšení toku Fordova-Fulkersonova algoritmu pro daný prvek A a všechny stromy daného prvku v nejvýše g -krát čili v čase $O(n \log n)$. Jelikož je prvků A celkem g (přibližně $\log_2 n$), je časová složitost algoritmu 4.1.2 $O(n^2 \log^2 n)$.

4.2 Implementace algoritmu pro určení cyklické hranové souvislosti kubického grafu

Jako praktický příklad byl implementován algoritmus 4.1.1 pro určení cyklické hranové souvislosti kubického grafu. Algoritmus byl implementován v programovacím jazyku C. Zdrojový soubor programu `Algoritmus.c` je na přiloženém CD.

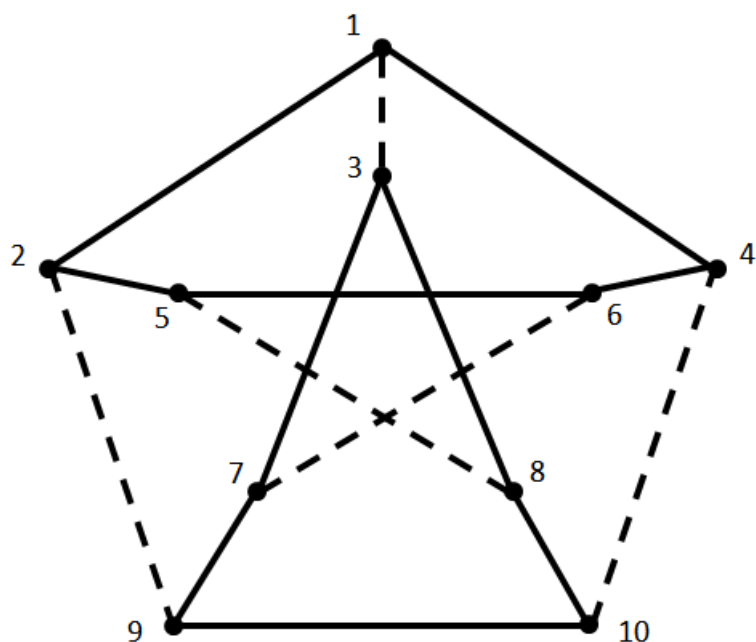
Program přijímá na vstupu grafy, které jsou reprezentovány pro každý vrchol v vstupního grafu G trojicí sousedů vrcholu v v grafu G . Předpokládá se, že tyto trojice budou zapsány v souboru `vstupniSoubor.txt`, který je ve stejné složce jako zdrojový soubor programu. Vrcholy vstupního grafu G musí být očíslovány od 1 do $v(G)$. V souboru `vstupniSoubor.txt` je pak pro $i = 1, \dots, v(G)$ zapsána trojice sousedů (čísel, přiřazených těmto vrcholům při očíslování) vrcholu i . Prvky v každé trojici jsou seřazeny vzestupně a odděleny mezerou. Jednotlivé trojice jsou odděleny mezerou nebo koncem řádku. V souboru `vstupniSoubor1.txt` na přiloženém CD je příklad správně zapsaného vstupního grafu.

Program spočítá podle algoritmu 4.1.1 cyklickou hranovou souvislost vstupního grafu, najde minimální cyklický hranový řez a výsledky zobrazí na standardním výstupu (na obrazovce). Cyklická hranová souvislost vstupního grafu je uložena v proměnné `c` a minimální cyklický hranový řez vstupního grafu je uložen v poli `CyklickyRez`. Podrobnější informace o činnosti programu jsou v komentářích ve zdrojovém souboru `Algoritmus.c`.

Program byl vyzkoušen na tzv. Petersenově grafu, na nesouvislém kubickém grafu, na náhodném kubickém grafu řádu 50 a na náhodném kubickém grafu řádu 100.

Petersenův graf je znázorněný na obr. 4.2.1. Program správně zjistil, že je jeho cyklická hranová souvislost 5, a našel jeho minimální cyklický hranový řez, jehož prvky jsou na obr. 4.2.1 vyznačeny přerušovanými čarami. Petersenův graf je zapsán jako vstupní graf programu v souboru `vstupniSoubor1.txt` na přiloženém CD.

Příkladem nesouvislého kubického grafu je graf A skládající se ze dvou komponent, z nichž každá je úplný graf řádu 4. Program správně zjistil, že je cyklická hranová souvislost grafu A rovna nule, a oznámil, že minimální cyklický hranový řez grafu A neobsahuje žádné prvky. Graf A je zapsán jako vstupní graf programu v souboru `vstupniSoubor2.txt` na přiloženém CD.



Obr. 4.2.1: Petersenův graf

Náhodně vygenerovaný kubický graf B řádu 50, resp. náhodně vygenerovaný kubický graf C řádu 100, je zapsán jako vstupní graf programu v souboru `vstupniSoubor3.txt`, resp. `vstupniSoubor4.txt`, na přiloženém CD. Výsledky programu pro Petersenův graf, graf A , graf B a graf C jsou shrnuty v tab. 4.2.1, ve které $\kappa'_c(G)$ značí cyklickou hranovou souvislost grafu G a C_{min} nalezený minimální cyklický hranový řez. V posledním sloupci tab. 4.2.1 je uveden celkový čas výpočtu pro příslušný vstupní graf.

Graf G	$v(G)$	$\kappa'_c(G)$	C_{min}	Čas
Petersenův	10	5	$\{\{8, 5\}, \{9, 2\}, \{3, 1\}, \{10, 4\}, \{7, 6\}\}$	1,625 s
A	8	0	\emptyset	1,406 s
B	50	3	$\{\{42, 25\}, \{7, 5\}, \{16, 37\}\}$	2,234 s
C	100	3	$\{\{49, 89\}, \{19, 8\}, \{58, 93\}\}$	3,735 s

Tab. 4.2.1: Výsledky programu

Nalezená implementace (vytvořený program) pracuje se stejnou časovou složitostí jako algoritmus 4.1.1, správně počítá cyklickou hranovou souvislost a nalézá minimální cyklický hranový řez vstupního grafu G .

5 Algoritmy pro určení esenciální hranové souvislosti grafu

5.1 Algoritmus pro určení esenciální hranové souvislosti grafu

Definice 5.1.1

Hranový řez $\partial(X)$ grafu G neobsahujícího žádnou smyčku takový, že existují alespoň dvě navzájem různé komponenty A a B grafu $G \setminus \partial(X)$, z nichž každá obsahuje alespoň jednu hranu, se nazývá esenciální hranový řez grafu G .

Definice 5.1.2

Nechť existuje esenciální hranový řez grafu G . Esenciální hranový řez $\partial(X)$ grafu G takový, že žádný esenciální hranový řez grafu G nemá menší velikost než $\partial(X)$, se nazývá minimální esenciální hranový řez grafu G . Velikost minimálního esenciálního hranového řezu grafu G nazýváme esenciální hranová souvislost grafu G a značíme $\kappa_e(G)$.

Definice 5.1.3

Nechť existuje esenciální hranový řez grafu G . Graf G je esenciálně hranově k -souvislý pro každé nezáporné celé k takové, že $k \leq \kappa_e(G)$.

Věta 5.1.1

Nechť existuje esenciální hranový řez grafu G a $k \in \mathbb{N} \cup \{0\}$. Graf G je esenciálně hranově k -souvislý, právě když neobsahuje žádný esenciální hranový řez $\partial(X)$ s vlastností $|\partial(X)| < k$.

Důkaz

„ \Rightarrow “: Nechť je G esenciálně hranově k -souvislý a existuje esenciální hranový řez $\partial(X)$ grafu G s vlastností $|\partial(X)| < k$. Podle definice 5.1.3 platí $k \leq \kappa_e(G)$, a tedy $|\partial(X)| < k \leq \kappa_e(G)$, kde je $\kappa_e(G)$ velikost minimálního esenciálního hranového řezu grafu G . $\partial(X)$ má tedy menší velikost než minimální hranový řez grafu G , což je spor.

„ \Leftarrow “: Pokud G neobsahuje žádný esenciální hranový řez $\partial(X)$ s vlastností $|\partial(X)| < k$, pak pro velikost jeho minimálního esenciálního hranového řezu $\kappa_e(G)$ platí $k \leq \kappa_e(G)$, a G je tedy podle definice 5.1.3 esenciálně hranově k -souvislý.

Definice 5.1.4

Hranový řez $\partial(X)$ grafu G takový, že $|\partial(X)| = 1$, se nazývá most.

Následující věta 5.1.2 plyne z věty 5.1.1 a jsou v ní shrnuty poznatky o esenciální hranové souvislosti grafu.

Věta 5.1.2

Nechť existuje esenciální hranový řez grafu G .

- a) Graf G je esenciálně hranově 0-souvislý.
- b) Graf G je esenciálně hranově 1-souvislý, právě když má právě jedna komponenta grafu G velikost větší nebo rovnu jedné.
- c) Graf G je esenciálně hranově 2-souvislý, právě když G neobsahuje most $\{uv\}$ s $d_G(u) > 1$ a $d_G(v) > 1$ a právě jedna komponenta grafu G má velikost větší nebo rovnu jedné.
- d) Esenciálně hranově 0-souvislý graf G je esenciálně hranově 1-souvislý, právě když má právě jedna komponenta grafu G velikost větší nebo rovnu jedné.
- e) Esenciálně hranově 1-souvislý graf G je esenciálně hranově 2-souvislý, právě když neobsahuje most $\{uv\}$ s $d_G(u) > 1$ a $d_G(v) > 1$.

Nyní je vhodné určit, které grafy mají esenciální hranový řez. To prozradí věty 5.1.3 a 5.1.4.

Věta 5.1.3

Graf G neobsahující smyčky obsahuje alespoň dvě hrany a a b nemající společný vrchol, právě když existuje esenciální hranový řez grafu G .

Důkaz

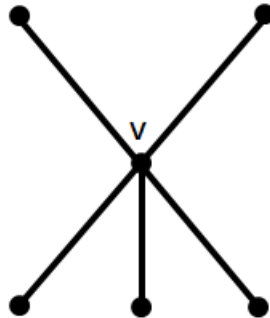
Pokud G obsahuje alespoň dvě hrany a a b nemající společný vrchol, pak např. všechny hrany mající právě jeden společný vrchol s hranou a tvoří esenciální hranový řez grafu G . Pokud naopak existuje esenciální hranový řez $\partial(X)$ grafu G , pak musí existovat hrana a v $G[X]$ a hrana b v $G[V(G) \setminus X]$, které nemají společný vrchol.

Důsledkem věty 5.1.3 je věta 5.1.4.

Definice 5.1.5

Prostý souvislý graf G , jehož každé dvě hrany mají právě jeden společný vrchol v (společný všem hranám grafu G), se nazývá hvězda.

Příklad hvězdy je znázorněný na obr. 5.1.1.



Obr. 5.1.1: Hvězda

Věta 5.1.4

Pro souvislý graf G neobsahující žádnou smyčku platí: Existuje alespoň jeden esenciální hranový řez grafu G , právě když platí $v(G) \geq 4$ a základní prostý graf grafu G není hvězda. Pro nesouvislý graf G neobsahující žádnou smyčku platí: Existuje alespoň jeden esenciální hranový řez grafu G , právě když platí právě jedna z následujících podmínek:

- Existují alespoň dvě navzájem různé komponenty A a B grafu G , z nichž každá má velikost větší nebo rovnu jedné.
- Právě jedna komponenta A grafu G obsahuje alespoň jednu hranu, platí $v(A) \geq 4$ a základní prostý graf grafu A není hvězda.

V následující větě 5.1.5 omezíme shora esenciální hranovou souvislost grafu G obsahujícího esenciální hranový řez.

Věta 5.1.5

Nechť existuje esenciální hranový řez grafu G . Pak platí $\kappa'(G) \leq \kappa_e(G) \leq 2(\Delta(G) - 1)$.

Důkaz

Minimální esenciální hranový řez grafu je jedním z hranových řezů oddělujících dva vrcholy grafu, a tedy $\kappa'(G) \leq \kappa_e(G)$. Víme, že G obsahuje esenciální hranový řez $\partial(X)$, a tedy musí existovat hrana a v $G[X]$ a hrana b v $G[V(G) \setminus X]$. Všechny hrany grafu

G , které mají s hranou a právě jeden společný vrchol, tvoří esenciální hranový řez $\partial(Z)$ grafu G ($a \in E(G[Z])$ a $b \in E(G[V(G) \setminus Z])$) a těchto hran je nejvýše $2(\Delta(G) - 1)$ (odstraní se nejvýše všechny hrany grafu G incidentní s koncovými vrcholy hrany a kromě hrany a). Platí tedy $\kappa_e(G) \leq 2(\Delta(G) - 1)$.

Algoritmus 5.1.1 (Esenciální hranová souvislost grafu)

- Vstup: graf G neobsahující smyčky.
 - Výstup: $\kappa_e(G)$ (pokud je pro G definována), nebo oznámení, že esenciální hranová souvislost grafu G není definována.
- 1) Pokud není esenciální hranová souvislost grafu G definována, jdi na 4.
 - 2) Najdi vrchol v grafu G stupně $\delta(G)$ a polož $\kappa_e(G) := e(G)$.
 - 3) Pro $i = 1 \dots \delta(G)$ udělej pro každou hranu e_i incidentní s vrcholem v v grafu G :
 - a. Pro každou hranu f grafu G nemající s hranou e_i žádný společný vrchol udělej:
 - i. Vytvoř z grafu G graf H stažením koncových vrcholů hrany e_i ve vrchol x a koncových vrcholů hrany f ve vrchol y .
 - ii. Spočítej $p'(x, y)$ v grafu H . Pokud platí $p'(x, y) < \kappa_e(G)$, polož $\kappa_e(G) := p'(x, y)$.
 - 4) Pokud je pro G definována esenciální hranová souvislost, vrať $\kappa_e(G)$. Jinak vrať „Pro G není esenciální hranová souvislost definována“.

V prvním kroku algoritmu 5.1.1 se zjistí, zda je pro vstupní graf G esenciální hranová souvislost definována. Pokud ne, pak se přejde na krok 4 a algoritmus skončí. Pokud ano, přejde se na krok 3. Necht' je $\partial(X)$ minimální esenciální hranový řez grafu G . Existují dvě hrany a a b (nemající společný vrchol) grafu G , které odděluje hranový řez $\partial(X)$, tj. $a \in E(G[X])$ a $b \in E(G[V(G) \setminus X])$. Stačí tedy určit velikost minimálního hranového řezu $\partial(Y)$ grafu G oddělujícího hrany a a b (tj. hranového řezu $\partial(Y)$ grafu G takového, že $a \in E(G[Y])$, $b \in E(G[V(G) \setminus Y])$ a pro každý hranový řez $\partial(Z)$ grafu G takový, že $a \in E(G[Z])$ a $b \in E(G[V(G) \setminus Z])$, platí $|\partial(Y)| \leq |\partial(Z)|$), jelikož $|\partial(X)| = |\partial(Y)|$. Necht' pro daný vrchol v grafu G je M_v množina všech hran incidentních s vrcholem v v grafu G . Alespoň jedna hrana z M_v není prvkem minimálního esenciálního hranového řezu $\partial(X)$ grafu G (v opačném případě by byl $\partial(X) \setminus M_v$ esenciální hranový řez grafu G takový, že $|\partial(X) \setminus M_v| < |\partial(X)|$, což je spor). V algoritmu 5.1.1 se tedy

vybere vrchol v stupně $\delta(G)$ a jednu z hran incidentních s vrcholem v můžeme bez újmy na obecnosti považovat za hranu a . Hrana b je pak jedna z hran grafu G , které nemají s hranou a žádný společný vrchol. Ve třetím kroku algoritmu 5.1.1 se počítá pro každou hranu e_i incidentní s vrcholem v (mezi nimiž je i hrana a) největší počet hranově disjunktních cest vedoucích mezi hranou e_i a každou hranou f nemající s e_i žádný společný vrchol (mezi hranami f je tedy i hrana b), tj. uv -cest, kde u je koncový vrchol hrany e_i , v je koncový vrchol hrany f a žádný z vnitřních vrcholů uv -cest není koncovým vrcholem hrany e_i ani hrany f . V jednom okamžiku se tedy musí spočítat maximální počet hranově disjunktních cest vedoucích mezi hranou a a b , který je roven velikosti minimálního hranového řezu oddělujícího a a b , a tedy i minimálního esenciálního hranového řezu grafu G . Esenciální hranová souvislost bude na konci algoritmu uložena v proměnné $\kappa_e(G)$, která je inicializována jako $e(G)$ a ve třetím kroku algoritmu 5.5.1 postupně snižována až na esenciální hranovou souvislost grafu G . Pro zjištění největšího počtu hranově disjunktních cest vedoucích mezi hranou e_i a hranou f se v kroku 3a algoritmu 5.1.1 stažením hrany e_i do vrcholu x a stažením hrany f do vrcholu y vytvoří graf H a v něm se spočítá lokální hranová souvislost $p'(x, y)$ podle algoritmu 3.1.1. Nejlepší dosud známá varianta algoritmu 3.1.1 má časovou složitost $O(mn)$ a v algoritmu 5.1.1 se použije pro každou hranu nejvýše $(e(G) - 1)$ -krát. Časová složitost algoritmu 5.1.1 je tedy $O(\delta(G)m^2n)$, kde je $\delta(G)$ nejmenší stupeň vrcholu grafu G .

5.2 Algoritmus pro určení esenciální hranové souvislosti kubického grafu

Následující věta 5.2.1 je důsledkem věty 5.1.4.

Věta 5.2.1

Jediný kubický graf G neobsahující smyčky, který nemá esenciální hranový řez, je graf K_2^3 .

Následující věta 5.2.2 plyne z věty 5.1.2 a jsou v ní shrnuty poznatky o esenciální hranové souvislosti kubického grafu.

Věta 5.2.2

Nechť je G kubický graf neobsahující smyčky a různý od K_2^3 .

- a) Graf G je esenciálně hranově 0-souvislý.
- b) Graf G je esenciálně hranově 1-souvislý, právě když je souvislý.
- c) Graf G je esenciálně hranově 2-souvislý, právě když je souvislý a neobsahuje most.
- d) Esenciálně hranově 1-souvislý graf G je esenciálně hranově 2-souvislý, právě když neobsahuje most.

Následující věta 5.2.3 plyne z věty 5.1.5.

Věta 5.2.3

Nechť existuje esenciální hranový řez grafu G . Pokud je G nesouvislý kubický graf, pak $\kappa_e(G) = 0$. Pokud je G souvislý kubický graf, pak $\kappa_e(G) \in \{1, 2, 3, 4\}$.

Věta 5.2.4

Nechť existuje esenciální hranový řez grafu G a G je hranově k -souvislý graf, kde $k \in \mathbb{N} \cup \{0\}$. Graf G je esenciálně hranově $(k + 1)$ -souvislý, právě když je každý hranový řez $\partial(X)$ grafu G , pro nějž platí $|\partial(X)| = k$, triviální.

Důkaz

„ \Rightarrow “: Jestliže je graf G esenciálně hranově $(k + 1)$ -souvislý, pak podle věty 5.1.1 neexistuje žádný esenciální hranový řez grafu G velikosti menší než $k + 1$, a každý hranový řez $\partial(X)$ grafu G , pro nějž platí $|\partial(X)| = k$, je tedy triviální.

„ \Leftarrow “: Jestliže je každý hranový řez $\partial(X)$ grafu G , pro nějž platí $|\partial(X)| = k$, triviální, pak neexistuje esenciální hranový řez grafu G velikosti menší než $k + 1$, a tedy G je podle věty 5.1.1 esenciálně hranově $(k + 1)$ -souvislý.

Z věty 5.2.4 plyne následující věta 5.2.5.

Věta 5.2.5

Nechť je G kubický graf neobsahující smyčky a různý od K_2^3 . G je hranově 3-souvislý, právě když je G esenciálně hranově 3-souvislý.

Důkaz

„ \Rightarrow “: Pokud je G hranově 3-souvislý, pak je podle věty 5.1.5 také esenciálně hranově 3-souvislý.

„ \Leftarrow “: Nechť je G esenciálně hranově 3-souvislý a není hranově 3-souvislý. Pak nastává právě jedna z následujících tří možností:

- 1) Platí $\kappa'(G) = 0$. V tomto případě je G nesouvislý, což je spor s tím, že je G esenciálně hranově 3-souvislý.
- 2) Platí $\kappa'(G) = 1$. V tomto případě G obsahuje most a to je spor s bodem c věty 5.2.2 a tím, že je G esenciálně hranově 3-souvislý.
- 3) Platí $\kappa'(G) = 2$. Jelikož je graf G esenciálně hranově 3-souvislý, musí být v tomto případě všechny hranové řezy grafu G velikosti 2 podle věty 5.2.4 triviální. To však znamená, že G obsahuje alespoň jeden vrchol stupně dva, což je spor s tím, že je G kubický graf.

Následující věta 5.2.6 je obecně známá. Zde ji uvádíme s jejím důkazem.

Věta 5.2.6

Nechť je G je kubický graf neobsahující smyčky, různý od K_2^3 a obsahující cyklický hranový řez. Graf G je esenciálně hranově 4-souvislý, právě když je G cyklicky hranově 4-souvislý (tj. velikost minimálního cyklického hranového řezu grafu G je alespoň 4).

Důkaz

„ \Rightarrow “: Jestliže je graf G esenciálně hranově 4-souvislý, pak podle věty 5.1.1 neexistuje esenciální hranový řez grafu G velikosti menší než 4, a tedy ani cyklický hranový řez grafu G velikosti menší než 4. Graf G je tedy cyklicky hranově 4-souvislý.

„ \Leftarrow “: Nechť je graf G cyklicky hranově 4-souvislý, ale není esenciálně hranově 4-souvislý, tj. podle věty 5.1.1 existuje esenciální hranový řez $\partial(X)$ grafu G velikosti menší než 4. Dokažme nyní sporem, že oba grafy $G[X]$ a $G[V(G) \setminus X]$ neobsahují vrchol stupně jedna. Nechť tedy graf $G[X]$ obsahuje vrchol stupně jedna. Jelikož je G kubický graf, znamená to, že dvě hrany e a f esenciálního hranového řezu $\partial(X)$ jsou incidentní s vrcholem v . Nechť je d hrana incidentní s vrcholem v obsažená v $G[X]$. Hranový řez $(\partial(X) \setminus \{e, f\}) \cup \{d\}$ je esenciální hranový řez grafu G ($G[X]$ obsahuje maximálně jeden vrchol stupně jedna, jelikož je G kubický graf a $\partial(X)$ má nejvýše tři prvky, a proto graf $G[X] \setminus d$ obsahuje alespoň jednu hranu a $(\partial(X) \setminus \{e, f\}) \cup \{d\}$ je tedy skutečně esenciální hranový řez grafu G) menší velikosti než $\partial(X)$, což je spor. Obdobně ani graf $G[V(G) \setminus X]$ neobsahuje vrchol stupně 1. Oba grafy $G[X]$ a $G[V(G) \setminus X]$ nejsou tedy stromy (ani lesy) a obsahují tedy kružnici. $\partial(X)$ je tedy cyklický hranový řez grafu G velikosti menší než 4, což je spor s tím, že je G cyklicky hranově 4-souvislý.

Algoritmus 5.2.1 (Esenciální hranová souvislost kubického grafu)

- Vstup: kubický graf G neobsahující smyčky.
 - Výstup: $\kappa_e(G)$ (pokud je pro G definována), nebo oznámení, že esenciální hranová souvislost grafu G není definována.
- 1) Pokud platí $v(G) < 243$, použij algoritmus 5.1.1 a jdi na 8.
 - 2) Pokud není esenciální hranová souvislost grafu G definována, jdi na 8.
 - 3) Pokud je G nesouvislý, polož $\kappa_e(G) := 0$ a jdi na 8.
 - 4) Pokud není G hranově 2-souvislý, polož $\kappa_e(G) := 1$ a jdi na 8.
 - 5) Pokud není G hranově 3-souvislý, polož $\kappa_e(G) := 2$ a jdi na 8.
 - 6) Pokud není G cyklicky hranově 4-souvislý, polož $\kappa_e(G) := 3$ a jdi na 8.
 - 7) Polož $\kappa_e(G) := 4$.
 - 8) Pokud je pro G definována esenciální hranová souvislost, vrať $\kappa_e(G)$. Jinak vrať „Pro G není esenciální hranová souvislost definována“.

Pokud pro vstupní graf G algoritmu 5.2.1 platí $v(G) < 243$, pak se použije pro určení esenciální hranové souvislosti grafu G algoritmus 5.1.1 s časovou složitostí $O(n^3)$, kde je n řád grafu G (jelikož u kubického grafu G platí $\delta(G) = 3$ a $m = \frac{3n}{2}$, kde m značí velikost grafu G). Pokud platí $v(G) \geq 243$, zjistí se v kroku 2, zda G obsahuje esenciální hranový řez (tj. zda není izomorfní s K_2^3). Pokud je G izomorfní s K_2^3 , algoritmus přejde na krok 8 a skončí. Pokud G není izomorfní s K_2^3 , zjistí se v kroku 3, zda je souvislý (např. pomocí BFS v čase $O(n)$). Pokud je G nesouvislý, platí $\kappa_e(G) = 0$ podle bodu b věty 5.2.2. Pokud je G souvislý, zjistí se v kroku 4, zda je G hranově 2-souvislý (např. pomocí algoritmu 3.2.1 v čase $O(n^2)$). Pokud souvislý graf G není hranově 2-souvislý (tj. obsahuje most), platí $\kappa_e(G) = 1$ podle bodu c věty 5.2.2. Pokud je G hranově 2-souvislý, zjistí se v kroku 5, zda je hranově 3-souvislý (použije se výsledek z kroku 4). Pokud hranově 2-souvislý graf G není hranově 3-souvislý, platí $\kappa_e(G) = 2$ podle věty 5.2.5. Pokud je G hranově 3-souvislý, zjistí se v kroku 6, zda je cyklicky hranově 4-souvislý (podle algoritmu 4.1.2 v čase $O(n^2 \log^2 n)$). Pokud hranově 3-souvislý graf G není cyklicky hranově 4-souvislý, platí $\kappa_e(G) = 3$ podle věty 5.2.6. Pokud je G cyklicky hranově 4-souvislý, je G podle věty 5.2.6 esenciálně hranově 4-souvislý a platí $\kappa_e(G) = 4$, jelikož $\kappa_e(G)$ může být maximálně 4 podle věty 5.2.3. Algoritmus 5.2.1 tedy vrátí esenciální hranovou souvislost vstupního grafu G v čase $O(n^2 \log^2 n)$.

6 Závěr

Prvním cílem této diplomové práce bylo podat přehled o algoritmech pro určení hranové souvislosti grafu. Byly vybrány pouze dosud nejlepší známé algoritmy pro určení hranové souvislosti grafu a testování hranové 2-souvislosti a hranové 3-souvislosti grafu. Dále byl popsán dosud nejlepší známý algoritmus pro určení cyklické hranové souvislosti grafu a jeho část pro malé grafy byla implementována, což byl druhý cíl této diplomové práce.

Navržení efektivního algoritmu pro testování esenciální hranové souvislosti ve vhodné třídě grafů bylo třetím cílem diplomové práce. Byl nalezen algoritmus pro určení esenciální hranové souvislosti grafu v čase $O(\delta(G)m^2n)$ a také algoritmus pro určení esenciální hranové souvislosti kubického grafu v čase $O(n^2 \log^2 n)$.

Časová složitost nalezeného algoritmu pro určení esenciální hranové souvislosti kubického grafu je stejná jako časová složitost uvedeného algoritmu pro určení cyklické hranové souvislosti kubického grafu, a proto je otevřeným problémem, zda nelze časovou složitost algoritmu pro určení esenciální hranové souvislosti kubického grafu ještě snížit.

Seznam použité literatury a zdrojů

- 1) BONDY, J. A. a U. S. R. MURTY. *Graph theory*. New York: Springer, c2008.
- 2) Z. Dvořák, J. Kára, D. Král' a O. Pangrác. An Algorithm for Cyclic Edge Connectivity of Cubic Graphs: Extended Abstract. *Lecture Notes in Computer Science* [online]. 2004, **33**(3111), 236-247 [cit. 2016-05-13]. Dostupné z: http://link.springer.com/chapter/10.1007%2F978-3-540-27810-8_21
- 3) A.-H. Esfahanian. *Connectivity Algorithms* [online]. [cit. 2016-05-13]. Dostupné z: http://www.cse.msu.edu/~cse835/Papers/Graph_connectivity_revised.pdf
- 4) WEST, D. B. *Introduction to graph theory*. 2nd ed. Upper Saddle River, N.J.: Prentice Hall, c2001. ISBN 0130144002.
- 5) A. Frank. On the Edge-Connectivity Algorithm of Nagamochi and Ibaraki. *EGRES Quick-Proof* [online]. 2009, **4**(1), 1-3 [cit. 2016-05-13]. Dostupné z: <http://www.cs.elte.hu/~frank/cikkek/FrankR2.pdf>
- 6) Z. Ryjáček. *Teorie grafů, diskrétní optimalizace a výpočetní složitost 1* [online]. Plzeň, 2014 [cit. 2016-05-13]. Dostupné z: <http://www.cam.zcu.cz/~ryjacek/students/ps/TGD1.pdf>
- 7) J. M. Schmidt. A simple test on 2-vertex- and 2-edge-connectivity. *Information Processing Letters* [online]. 2013, **113**(7), 241-244 [cit. 2016-05-13]. Dostupné z: <http://www.sciencedirect.com/science/article/pii/S0020019013000288>
- 8) K. Mehlhorn, A. Neumann a J. M. Schmidt. Certifying 3-Edge-Connectivity. *Lecture Notes in Computer Science* [online]. 2013, **42**(8165), 358-369 [cit. 2016-05-13]. Dostupné z: http://link.springer.com/chapter/10.1007/978-3-642-45043-3_31

Přílohy

Přílohou diplomové práce je CD, které obsahuje následující soubory:

- DP_Kraus.docx – plný text diplomové práce ve formátu *.docx.
- DP_Kraus.pdf – plný text diplomové práce ve formátu *.pdf.
- Algoritmus.c – zdrojový soubor programu pro určení cyklické hranové souvislosti kubického grafu.
- VstupniSoubor.txt – prázdný soubor určený pro vstupní data programu.
- VstupniSoubor1.txt – Petersenův graf.
- VstupniSoubor2.txt – nesouvislý kubický graf.
- VstupniSoubor3.txt – kubický graf řádu 50.
- VstupniSoubor4.txt – kubický graf řádu 100.