Minimální kostra grafu

a její využití

Autor: Lukáš Macek
Obor: Obecná matematika
Ročník: První
Předmět: *Ukázky aplikací matematiky*

Univerzita Karlova v Praze

Matematicko-fyzikální fakulta

2018–2019, letní semestr

Obsah

Realita

1 Něco z historie

Na přelomu 19. a 20. století se začal hojně rozšiřovat elektrický proud do různých koutů světa. Aby se rozsvítily žárovky v každé domácnosti, bylo potřeba postavit elektrické vedení, které by elektrický proud vyráběný v elektrárnách zprostředkovávalo. I Češi si tehdy museli položit zásadní otázku: „Kudy elektrické dráty povedou?“ Samozřejmě je důležité ušetřit co nejvíc materiálu, ale na druhou stranu by bylo vhodné vedení zavést do všech obcí.

S podobným problémem se už ale lidstvo setkalo o něco dříve, když se začala rozšiřovat železniční síť. V té době taky bylo žádoucí, aby každá dvě města daného regionu byla dosažitelná, ovšem s minimálními náklady na výstavbu.

Daný problém lze řešit velmi efektivně s pomocí jednoduchého matematického modelu.

Matematický model

2 Oddíl definic

2.1 Ohodnocený graf

Jednoduchý graf = uspořádaná dvojice množiny vrcholů a množiny hran. Každá hrana je dvouprvkovou množinou, jejímiž dvěma prvky jsou právě prvky množiny vrcholů daného grafu.

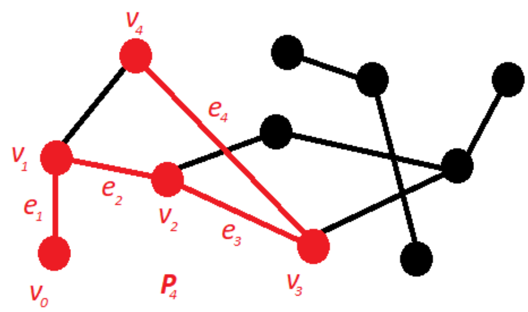
Ohodnocení hran = zobrazení množiny hran do nějaké množiny ohodnocení hran, jejímiž prvky jsou reálná čísla. Ohodnocení každé hrany se jinak nazývá váha dané hrany.

Ohodnocený graf = uspořádaná trojice množiny vrcholů, množiny hran a zobrazení ohodnocení hran.

2.2 Soubor definic pro kostru grafu

Cesta v grafu

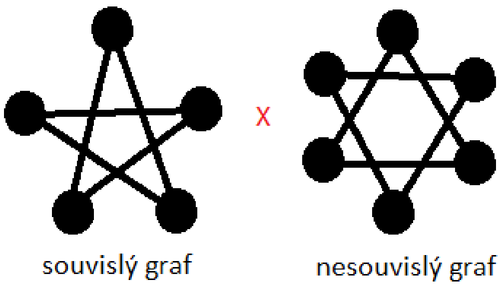
Nechť $G = (V, E)$ je graf. Cestou délky n v grafu G nazýváme libovolnou posloupnost $(v_0, e_1, v_1, e_2, v_2, \dots, v_{n-1}, e_n, v_n)$, kde v_i jsou navzájem různé vrcholy grafu G pro $i \in \{0, 1, \dots, n\}$ a $\forall i \in \{1, \dots, n\} : e_i = \{v_{i-1}, v_i\}, e_i \in E$. Značíme ji P_n.



Souvislost grafu

Graf $G = (V, E)$ je souvislý právě tehdy, když pro každé dva vrcholy $x,\ y \in V$ existuje v grafu G cesta z x do y.

Poznámka: U nesouvislého grafu H nazýváme souvislé části, ze kterých se graf skládá, **komponenty souvislosti** grafu H.

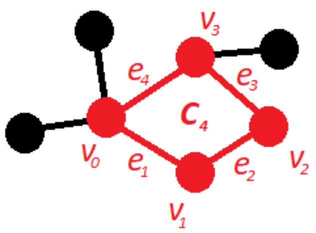


Nesouvislý graf na obrázku obsahuje dvě komponenty souvislosti, každá z nich je cyklus délky 3.

Poznámka: Každý souvislý neprázdný graf má právě jednu komponentu souvislosti. (Celý graf.)

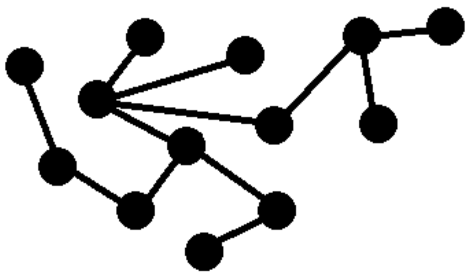
Cyklus v grafu

Nechť $G = (V, E)$ je graf. Cyklem (kružnicí) délky n v grafu G nazýváme libovolnou posloupnost $(v_0, e_1, v_1, e_2, v_2, \dots, v_{n-1}, e_n, v_0)$, kde v_i jsou navzájem různé vrcholy grafu G pro $\{0, 1, \dots,\ n-1\}$ a $\forall i \in \{1, \dots,\ n\} : e_i = \{v_{i-1}, v_i\}, e_i \in E$. Značíme ho C_n.



Strom

Stromem nazýváme každý neprázdný souvislý graf, který neobsahuje cyklus.

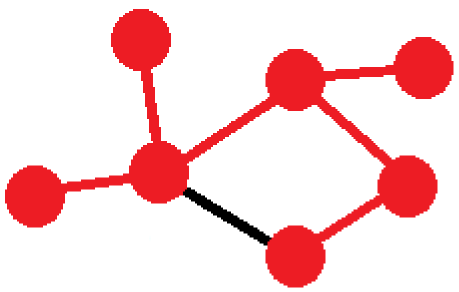


Graf na obrázku je stromem, protože je neprázdný a neobsahuje žádný cyklus.

Poznámka: Lze vidět, že pro něj platí $|E| = |V| - 1$, což platí právě tehdy, když graf je stromem.

Kostra grafu

Nechť $G = (V, E)$ je graf. Kostrou grafu nazveme libovolný strom $T = (V, E')$, kde $E' \subseteq E$.



Na obrázku je jedna ze čtyř koster grafu, který jsme viděli na obrázku u definice cyklů. (Kostra je červeně zvýrazněna.)

Poznámka: Každý strom má jedinou kostru, která obsahuje všechny jeho hrany i vrcholy.

3 Problém hledání minimální kostry grafu

3.1 Potřebné definice

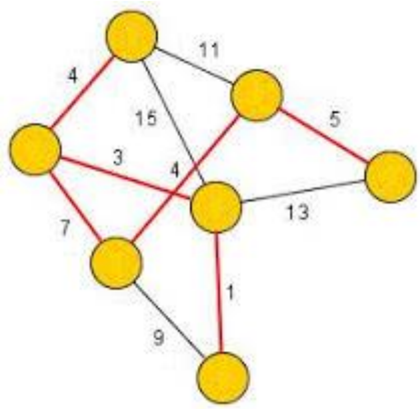
Podgraf

Nechť $G = (V, E)$ je graf. Graf $H = (V', E')$ je podgrafem grafu G právě tehdy, když platí $V' \subseteq V$ a $E' \subseteq E \cap \binom{V'}{2}$.

Poznámka: Zvýrazněná cesta, cyklus i kostra na obrázcích výše jsou podgrafy původního grafu.

Minimální kostra

Nechť $G = (V, E)$ je souvislý ohodnocený graf s nezápornými vahami hran $w(e_1), \dots, w(e_n)$, kde $n = |E|$ a e_i jsou navzájem různé hrany grafu G pro $i \in \{1, \dots, n\}$. Nejmenší kostrou grafu G je takový podgraf $T = (V, E')$ grafu G, který ze všech možných podgrafů grafu G má nejmenší součet hodnot $w(e_i)$ pro všechny i, takové, že $e_i \in E'$.
(Jinými slovy takový podgraf $T = (V, E')$, kde výraz $\sum_{e \in E'} w(e)$ nabývá nejmenší hodnoty.)



Na obrázku je souvislý graf s kladným ohodnocením hran. Červeně jsou znázorněny hrany jeho jediné minimální kostry.

Poznámka: Připomeňme že každá kostra grafu obsahuje všechny jeho vrcholy.

3.2 Otázka

Jak zaručeně najít minimální kostru libovolného souvislého grafu s kladným ohodnocením hran?

Na tuto otázku nám odpoví následující algoritmy.

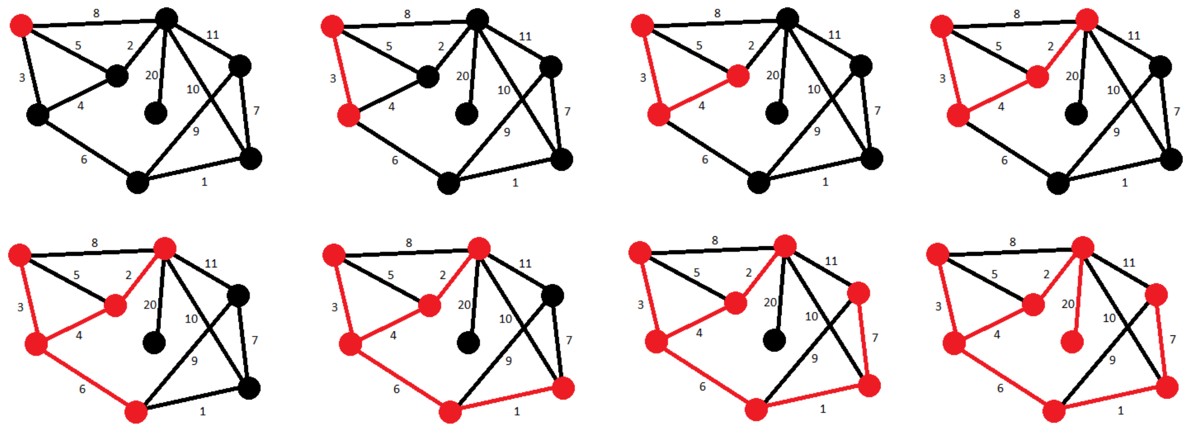
3.3 Algoritmy pro nalezení minimální kostry grafu

3.3.1 Jarníkův algoritmus

- Nechť $G = (V, E)$ je souvislý neprázdný ohodnocený graf. Položme $E_0 = \emptyset$ a $V_0 = \{v\}$, kde v je libovolný vrchol G.
- Dokud existuje alespoň jedna hrana vyhovující níže popsané podmínce, opakujme:
    1. Nalezněme hranu $e_i = \{x_i, y_i\} \in E$ nejmenší možné váhy z takové množiny hran, že $x_i \in V_{i-1}$ a $y_i \; V \setminus V_{i-1}$.
    2. Položme nové množiny $V_i = y_i \cup V_{i-1}$ a $E_i = e_i \cup E_{i-1}$.
- $T = (V_i, E_i) = (V, E_{|V|-1})$ je minimální kostra grafu G.

Poznámka: Může se stát, že máme na výběr, kterou hranu podle algoritmu zvolíme. V takovém případě hranu volíme náhodně, a tedy se výstup může lišit při použití toho stejného algoritmu dvakrát (od začátku) na stejný vstup. Mohutnost množiny hran budou však stejnou a pro oba výstupy bude platit, že výstupní graf je minimální kostrou vstupního.

Znázornění hledání minimální kostry pomocí Jarníkova algoritmu vidíme na obrázku.



Jarníkův algoritmus

3.3.1.1 Důkaz správnosti Jarníkova algoritmu

Je zřejmé, že graf $T = (V, E_i)$ je kostra grafu G.
Nechť $e_1, e_2, \dots, e_{|V|-1}$ jsou hrany T očíslované v tom pořadí, v jakém je algoritmus přidal do $E_{|V-1|}$.
Nechť pro spor není T minimální a nechť T_{min} je minimální kostra G.
Nechť k je nejmenší index takový, že $e_k \notin T_{min}$. Pak vybereme T_{min} tak, že k je co největší.
Přidáme-li do množiny hran grafu T_{min} hranu e_k, nový graf zřejmě obsahuje kružnici; označme ji C.
Nechť V' je množina vrcholů přidaných do V_i v prvních $k-1$ iteracích (tedy po přidání e_{k-1} do E_i).
Kružnice C obsahuje hranu e_k a jednu další hranu e' spojující V' s $V \backslash V'$.

(Pro ulehčení zápisu předpokládejme, že $w: E \to (0;\ \infty)$ je kladné ohodnocení hran grafu G.)
Srovnejme $w(e')$ s $w(e_k)$:

1) $w(e') < w(e_k)$:
   Spor s tím, že algoritmus zvolil e_k místo e'.
2) $w(e') > w(e_k)$:
   Spor, protože přidáme-li do množiny hran grafu T_{min} hranu e_k a odebereme-li hranu e', dostaneme zase kostru grafu G, ale s menší celkovou vahou než T_{min}, což znamená, že T_{min} nebyla minimální kostrou grafu G.
3) $w(e') = w(e_k)$:
   Přidáme-li do množiny hran grafu T_{min} hranu e_k a odebereme-li odtud hranu e', získáme minimální kostru grafu G, která obsahuje všechny hrany $e_1, e_2, \dots, e_k$, což je spor s tím, že jsme volili T_{min} tak, aby k bylo co největší.

Jelikož žádná jiná možnost nemůže nastat, nastává spor, a tedy T, což je kostra grafu G nalezena Jarníkovým algoritmem, je minimální. $\square$

3.3.1.2 Důsledek Jarníkova algoritmu

Každý souvislý graf s prostým ohodnocením hran má právě jednu minimální kostru.
Tato kostra je jednoznačně určená lineárním uspořádáním hran.
(Poznámka: Obecně však minimální kostra jednoznačně určena není.)

3.3.2 Kruskalův („hladový“) algoritmus

- Nechť $G = (V, E)$ je souvislý ohodnocený graf. Položme $E_0 = \emptyset$.
- Dokud je to možné, opakujme:
  1. Nalezněme hranu $e_i \in E \setminus E_{i-1}$ s nejmenší možnou vahou takovou, že mezi hranami již vybranými (tj. mezi hranami z množiny E_{i-1}) nevytvoří kružnici.
  2. Položme $E_i = E_{i-1} \cup e_i$.
- Graf $T = (V, E_i)$ je minimální kostrou grafu G.

Poznámka: Podobně jako u Jarníkova algoritmu je někdy možné vybrat si z více než jedné hrany. V každém případě bude však na výstupu minimální kostra grafu G.

Poznámka: Pro prázdný graf G je výstupem také prázdný graf, tj. $T = (\emptyset, \emptyset)$.

3.3.2.1 *Důkaz správnosti Kruskalova algoritmu*

Nejprve musíme dokázat, že algoritmus skutečně vytvoří kostru grafu G.

Bez újmy na obecnosti předpokládejme, že je G úplný graf, tj. $E = \binom{V}{2}$. Pokud G úplný není, můžeme hrany, které jsou jeho doplňkem do úplného grafu na $|V|$ vrcholech, nahradit za hrany s neporovnatelně větším ohodnocením.

I. *Důkaz vytvoření kostry:*

Víme, že T je podgrafem grafu G.
Dále T zřejmě neobsahuje kružnici, protože poslední hrana přidaná do kružnice by spojovala ten samý podstrom a ne dva rozdílné, což je podmínkou pro přidání nové hrany.
T nemůže být nesouvislý, protože hrany přidané do T slučující podstromy byly do T přidány.
Protože T je souvislý a neobsahuje kružnici, je zřejmé, že T je strom.
Protože T obsahuje všechny vrcholy grafu G, je T kostra grafu G.

II. *Důkaz minimální kostry:*

Pro spor předpokládejme, že T je kostrou grafu G, ale není minimální. Označme T_{min} minimální kostru grafu G, která má co nejmenší počet hran, které se nenacházejí v T. Index první hrany, která je v T, ale není v T_{min}, označme k. Pak e_k je první hrana, která by byla algoritmem přidána do T z těch, které nejsou v T_{min}. (Hrany grafu T označené $e_1, e_2, \dots, e_{|V|-1}$ jsou seřazené v tom pořadí, v jakém je algoritmus do grafu T přidal.)
Pak zřejmě bude graf T_{min} tvořit kružnici, přidáme-li do jeho množiny hran hranu e_k.
Jelikož T je kostra, a tedy i strom, zřejmě neobsahuje kružnici.
Proto musí nestromový graf T_{min} obsahující navíc hranu e_k obsahovat ještě hranu e', která v množině hran grafu T není obsažena. Označme takový graf $T_{min}^{e_k}$.
Odebereme-li z množiny hran grafu $T_{min}^{e_k}$ hranu e', získáme zase kostru grafu G.
Celkové ohodnocení hran této kostry však nemůže být menší než celkové ohodnocení hran grafu T_{min}, a proto nemůže mít e_k menší ohodnocení než e'.

Předpokládejme tedy, že ohodnocení e_k je větší než ohodnocení e'.
Pamatujme na to, že algoritmus hrany vybírá v neklesajícím pořadí jejich ohodnocení.
Proto by byla hrana e' testována pro přidání do T (či T_{min}) před hranou e_k.
Tedy $\exists i \in \{1, \dots, k-1\}$: $e' = e_i$. To znamená, že v této fázi průběhu algoritmu je zatím $T = T_{min}$.

Protože přidání hrany e' do T_{min} nevytvoří kružnici, nemůže vytvořit kružnici ani v T, jelikož jsou zatím oba grafy identické. To je ale spor s tím, že e' v grafu T kružnici vytvoří.

Proto jsou ohodnocení hran e' a e_k stejná. Z předchozích úvah také plyne, že graf $T_{min}^{e_k}$, z jehož množiny hran odebereme hranu e' je minimální kostrou grafu G.
Pak má ale tento graf o jednu společnou hranu s T víc než s grafem T_{min}, což je ve sporu s výběrem T_{min} tak, aby měla co nejmenší počet hran, které se nenachází v T.

Protože jiná možnost nemůže nastat, je graf T, který byl nalezen Kruskalovým algoritmem, minimální kostrou grafu G.

□

3.3.3 Borůvkův algoritmus

- Nechť $G = (V, E)$ je souvislý neprázdný graf s prostým ohodnocením hran.
- Vezměme všechny vrcholy grafu G a považujme je za samostatné komponenty.
- Opakuj, dokud nemáme právě jednu komponentu souvislosti:
    - Pro každou komponentu souvislosti proveď:
        - Najdi hranu z E s minimální vahou takovou, že vede z této komponenty souvislosti do jiné, tedy takovou $e_i \in E :\ e_i = \{x, y\} \Leftrightarrow x \in C_i \,\&\, y \notin C_i$, e_i má nejmenší váhu z množiny všech hran, pro které platí předchozí podmínky, kde C_i je komponenta souvislosti, pro kterou takovou hranu hledáme.
        - Spoj touto hranou komponentu souvislosti, ze které hrana vede, s jinou komponentou souvislosti, do které hrana vede, tedy pro $e_i = \{x, y\}, x \in C_i, y \in C_i'$ spoj hranou e_i komponenty souvislosti C_i a C_i'.
- Výsledná komponenta souvislosti bude podgrafem grafu G a zároveň jeho minimální kostrou, tedy $T = (V, E')$, kde E' je množina všech vybraných hran, kterými jsme komponenty souvislosti spojovali.

Poznámka: Díky prostotě funkce ohodnocení hran grafu G spojováním jednotlivých komponent vybranými hranami (s nejmenší možnou vahou) nikdy nevznikne cyklus.

Poznámka: Při každé iteraci vnějšího cyklu se zmenší počet komponent souvislosti alespoň o polovinu. (Pro nalezení minimální kostry může stačit i pouhá jedna iterace.)

3.3.3.1 Důkaz správnosti Borůvkova algoritmu

Protože algoritmus spojuje hranou vždy 2 různé komponenty souvislosti a vykoná to celkově $(|V| - 1)$krát, bude pro T platit, že jeho množina vrcholů je shodná s množinou vrcholů grafu G, a tedy bude mít mohutnost $|V|$, a jeho množina hran bude mít mohutnost $|V| - 1$, což nutně znamená, že T je strom, jelikož je graf T souvislý, protože obsahuje pouze 1 komponentu souvislosti a víme, že G byl souvislý. Proto je T kostra grafu G. (Poznámka: Algoritmus by mohl vytvořit cyklus, pokud by ohodnocení hran nebylo prosté!)

Protože hrany vybíráme od té s nejmenší vahou, nemůže se nikdy stát, že by existovala hrana s menší vahou než ta, kterou jsme přidali mi, s tím, že by splňovala nutné podmínky, tedy že by oba její vrcholy byly z jiné komponenty souvislosti. Jinými slovy by taková hrana nutně vytvořila v našem grafu cyklus. Z toho plyne, že výsledná komponenta souvislosti, a tedy graf T, nalezen Borůvkovým algoritmem, je minimální kostrou grafu G.

□

Závěr

4 Využití

Uvažujme problém s elektrifikací určité oblasti, který byl popsán výše. Vytvoříme-li si souvislý ohodnocený graf $G = (V, E)$, kde každá obec z dané oblasti je zastoupena vrcholem $v_i \in V$ s tím, že každý úsek spojující dvě různé obce, na kterém by bylo možné elektrické vedení vybudovat, je zastoupen hranou $e_i \in E$ s ohodnocením $w(e_i)$ takovým, že $w: E \to \mathbb{R}$ je funkce, která každé hraně $e_i \in E$ přiřadí takové přirozené číslo, které odpovídá zaokrouhlené číselné hodnotě ceny výstavby případného elektrického vedení na tomto úseku v korunách.

Aplikujeme-li pak na tento graf G jeden z výše uvedených algoritmů, pro který máme splněny všechny předpoklady, získáme minimální kostru grafu G.

5 Interpretace výsledků

Hrany minimální kostry grafu G sestaveného podle popisu výše jsou pak v našem daném regionu právě ty úseky, kudy bychom měli elektrické vedení vést, pokud chceme, aby všechny obce z regionu byli elektrickým vedením propojeny a abychom co nejvíc ušetřili.

Podobně můžeme tento postup aplikovat na jakýkoliv problém stejného typu.

6 Citace

6.1 Inspirace

Na toto téma jsem přišel díky přednáškám pana prof. RNDr. Loebla, CSc. v kurzu diskrétní matematiky, který byl veden minulý semestr (2018–2019, ZS) na MFF UK v Praze.
Pomocí těchto poznámek jsem zhotovil některé uvedené definice a důkaz Jarníkova algoritmu.

6.2 Články z webu

NEZNÁMÝ. *Počátky elektrifikace* [online]. [cit. 22. 3. 2019]. Dostupný na WWW: https://www.cez.cz/edee/content/file/static/encyklopedie/encyklopedie-energetiky/05/pocatky_3.html

JIROVSKÝ, Lukáš. *Cesta a souvislost grafu* [online]. [cit. 22. 3. 2019]. Dostupný na WWW: https://teorie-grafu.cz/zakladni-pojmy/cesta-a-souvislost-grafu.php#souvislost

JIROVSKÝ, Lukáš. *Hledání minimální kostry grafu* [online]. [cit. 22. 3. 2019]. Dostupný na WWW: https://teorie-grafu.cz/vybrane-problemy/minimalni-kostra.php

NEZNÁMÝ. *Problém minimální kostry* [online]. [cit. 6.4.2019]. Dostupný na WWW: https://mj.ucw.cz/vyuka/0809/ads1/6-kostry-booklet.pdf

NEZNÁMÝ. *Kruskalův algoritmus* [online]. [cit. 6.4.2019]. Dostupný na WWW: https://cs.wikipedia.org/wiki/Kruskal%C5%AFv_algoritmus#D%C5%AFkaz_spr%C3%A1vnosti

NEZNÁMÝ. *Algoritmy pro hledání kostry* [online]. [cit. 22. 3. 2019]. Dostupný na WWW: https://cs.wikipedia.org/wiki/Kostra_grafu

ALGORITMY.NET. *Borůvkův algoritmus* [online]. [cit. 23.3.2019]. Dostupný na WWW: https://www.algoritmy.net/article/1396/Boruvkuv-algoritmus

6.3 Obrázky

NEZNÁMÝ. *Týden.cz* [online]. [cit. 23.3.2019]. Dostupný na WWW: https://www.tyden.cz/rubriky/zahranici/rusko-a-okoli/ukrajina-opravila-elektricke-vedeni-na-krym-proud-ale-neposila_367512.html

SKLENIČKA, Petr. *Linuxsoft.cz* [online]. [cit. 23.3.2019]. Dostupný na WWW: https://www.linuxsoft.cz/article.php?id_article=1797