FAKULTA
INFORMAČNÍCH
TECHNOLOGIÍ
ČVUT V PRAZE

Zadání bakalářské práce

Název:	Parametrizované algoritmy pro problém Min-Power Symmetric Connectivity
Student:	Daniel Dajbov
Vedoucí:	Mgr. Michal Opler, Ph.D.
Studijní program:	Informatika
Obor / specializace:	Teoretická informatika
Katedra:	Katedra teoretické informatiky
Platnost zadání:	do konce letního semestru 2024/2025

Pokyny pro vypracování

Cílem práce je navrhnout či modifikovat existující parametrizované algoritmy pro problém Min-Power Symmetric Connectivity, jehož motivací je nalezení minimálního celkového přenosového výkonu nutného k zajištění konektivity symetrické bezdrátové komunikační sítě. Tento problém je NP-úplný [3] a byl již uvažován i z pohledu parametrizované složitosti [1]. Nicméně nebyla prozkoumána jeho parametrizovaná složitost vzhledem ke strukturálním parametrům grafu komunikační sítě. V práci se student zaměří na tento typ parametrizace.

Pokyny pro studenta:
1. Provede rešerši literatury na téma problému Min-Power Symmetric Connectivity.
2. Nastuduje existující algoritmus parametrizovaný velikostí vrcholového pokrytí a počtem různých vah pro asymetrickou verzi problému zvanou Min-Power Asymmetric Connectivity [2].
3. Navrhne vlastní či modifikuje výše zmíněný algoritmus pro problém Min-Power Symmetric Connectivity a pokusí se ho rozšířit pro obecnější parametrizaci, např. sousedskou různorodostí či velikostí vrcholového pokrytí bez omezení na počet různých vah.

[1] Matthias Bentert, René van Bevern, André Nichterlein, Rolf Niedermeier, Pavel V. Smirnov: Parameterized Algorithms for Power-Efficiently Connecting Wireless Sensor Networks: Theory and Experiments. INFORMS J. Comput. 34(1): 55-75 (2022)

Elektronicky schválil/a doc. Ing. Jan Janoušek, Ph.D. dne 12. února 2024 v Praze.

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ ČVUT V PRAZE

[2] Matthias Bentert, Roman Haag, Christian Hofer, Tomohiro Koana, André Nichterlein: Parameterized Complexity of Min-Power Asymmetric Connectivity. Theory Comput. Syst. 64(7): 1158-1182 (2020)
[3] Lefteris M. Kirousis, Evangelos Kranakis, Danny Krizanc, Andrzej Pelc: Power consumption in packet radio networks. Theor. Comput. Sci. 243(1-2): 289-305 (2000)

Elektronicky schválil/a doc. Ing. Jan Janoušek, Ph.D. dne 12. února 2024 v Praze.

Bakalářská práce

PARAMETRIZOVANÉ ALGORITMY PRO PROBLÉM MIN-POWER SYMMETRIC CONNECTIVITY

Daniel Dajbov

Fakulta informačních technologií
Katedra teoretické informatiky
Vedoucí: Mgr. Michal Opler, Ph.D.
16. května 2024

Odkaz na tuto práci: Dajbov Daniel. *Parametrizované algoritmy pro problém Min-Power Symmetric Connectivity.* Bakalářská práce. České vysoké učení technické v Praze, Fakulta informačních technologií, 2024.

Obsah

Poděkování iv

Prohlášení v

Abstrakt vi

1 Úvod 1

2 Značení 2
- 2.1 Teorie grafů . . . 2
- 2.2 Definice problému . . . 3
- 2.3 Parametrizovaná složitost . . . 4

3 Známé výsledky 6

4 Vrcholové pokrytí 9
- 4.1 Základní pojmy . . . 9
  - 4.1.1 Algoritmus parametrizovaný vrcholovým pokrytím . . . 9
- 4.2 Algoritmus parametrizovaný vrcholovým pokrytím a počtem různých typů vah . . . 13

5 Sousedská různorodost 18
- 5.1 Základní pojmy . . . 18
- 5.2 Algoritmus parametrizovaný barevnou sousedskou různorodostí 19
- 5.3 Barevná sousedská různorodost a sousedská různorodost . . . 22
- 5.4 Těžkost parametrizace pomocí sousedské různorodosti . . . 24

6 Závěr 26

Seznam obrázků

2.1 Instance problému MinPSC s optimalni kostrou vyznačenou tučně modře a u každého vrcholu je šedě jeho ohodnocení . . . 4

4.1 Rozklad grafu G s vrcholovým pokrytím $M = \{v_1, v_2, v_3\}$ a vahami hran z množiny $Q = \{4, 5, 10, 11, 12\}$ 14

5.1 Bipartitní graf se dvěma partitami o velikosti tři. Modrá hrana symbolizuje váhu hrany 2 a černá hrana symbolizuje váhu hrany 1 23

Chtěl bych poděkovat především svému vedoucímu Mgr. Michalu Oplerovi, Ph.D. za trpělivost a ochotu pomoci při spolupráci se mnou. Zvlášť v posledních týdnech do deadlinu. Dále bych chtěl poděkovat rodině a kamarádům za podporu a za pochopení, že jsem na ně někdy neměl tolik času, kolik bych chtěl nebo kolik by si oni přáli. A nakonec chci poděkovat svému zdraví, které to snad vydrželo až dokonce.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací. Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů, zejména skutečnost, že České vysoké učení technické v Praze má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 citovaného zákona.

V Praze dne 16. května 2024

Abstrakt

V bakalářské práci představíme problém Min Power Symmetric connectivity (MinPSC), který je na obecných grafech NP-těžký. Problém MinPSC má uplatněni v sítích, kde pro daný uzel chceme zmenšit jeho spotřebu energii na co nejmenší úroveň a požadujeme přitom, aby každý uzel mohl komunikovat s ostatními. V této práci zkoumáme problém MinPSC ze pohledu dvou strukturálních parametru. A to z vrcholového pokrytí a sousedské různorodosti.

Klíčová slova komunikační sítě, optimalizace vysílacího výkonu, teorie grafů, parametrizované algoritmy, vrcholové pokrytí, sousedská různorodost

Abstract

In this bachelor thesis, we introduce Min Power Symmetric Connectivity (MinPSC) problem which is NP-complete. The application of the problem is in networks, where we want to reduce a consumption of energy as much as possible. And while it is required to have a connection between each node in the network. In this thesis, we examine the problem MinPSC from the point of view of two structural graph parameters. Vertex cover and neighbourhood diversity.

Keywords communication networks, optimization of transmission power, graph theory, parameterized algorithms, vertex cover, neighbourhood diversity

Kapitola 1

Úvod

Min Power Symmetric Connectivity, zkráceně MinPSC, je problém nalezení minimálního celkového přenosového výkonu nutného k zajištění konektivity symetrické bezdrátové komunikační sítě. V bezdrátových sítích jsou uzly typicky napájeny bateriemi, a proto je důležité efektivně spravovat jejich energii, aby se prodloužila doba jejich provozu. Základní myšlenka spočívá v tom, že každý uzel nastavuje svůj vysílací výkon na minimální možnou úroveň, která je stále dostatečná k udržení komunikace s jeho sousedy. Toto nastavení musí být provedeno tak, aby bylo zajištěno, že celá síť zůstane souvislou. To znamená, že musí existovat alespoň jedna komunikační cesta mezi každými dvěma uzly v síti. Důležitým aspektem je symetrie, což znamená, že pokud uzel A může komunikovat s uzlem B, pak i uzel B musí být schopen komunikovat s uzlem A s použitím stejné nebo nižší úrovně vysílacího výkonu.

Popsaný problém má aplikaci v mnoha situacích, kdy tradiční sítě by byly na implementaci moc drahé nebo někdy až nemožné. Mohlo by se jednat v případě historických budov, kde instalace komunikační sítě pomocí kabelů by byla z vizuálního hlediska nepřijatelný krok. Dále můžeme zmínit situaci, kdy nastala katastrofa, například povodeň. V této situaci, kdy vybudovaná síť je nefunkční, je potřeba zajistit prozatímní síť.

Tato práce má následující strukturu. V kapitole 2 si v kostce zavedeme pojmy, které v této práci budeme používat. Ve kapitole 3 představíme předchozí známe výsledky MinPSC. V kapitole 4 se podíváme na tento problém z pohledu vrcholového pokrytí a v kapitole 5 z pohledu sousedské různorodosti.

Kapitola 2
Značení

Označme množinu přirozených čísel $\mathbb{N}$ jako $\{0, 1, 2, 3, \dots\}$. Zápisem $[d]$, kde $d \in \mathbb{N}$ myslíme množinu přirozených čísel $\{1, 2, \dots, d\}$. Zápisem $\bigcup x$ myslíme množinu $\bigcup x = \{z \mid (\exists y)(z \in y) \wedge (y \in x)\}$, což znamená sjednocení množin obsažených v množině x. Například pro $x = \{\{1, 2\}, \{2, 3, 4\}\}$ by platilo, že $\bigcup x$ je množina $\{1, 2, 3, 4\}$.

2.1 Teorie grafů

Předpokládáme, že jsme obeznámeni s základními pojmy teorie grafů. V této kapitole definujeme pojmy bez hlubšího vysvětleni, které budeme nadále používat. Dle potřeby doporučujeme například následující literaturu [1].

Konečný prostý graf je dvojice $G = (V, E)$, kde $V(G)$ je konečná množina vrcholů a $E(G) \subseteq \binom{V(G)}{2}$ je množina hran. Mějme graf $G = (V, E)$. Značením $V(G)$ myslíme množinu vrcholů grafu G a $E(G)$ množinu hran grafu G. Číslem $n \in \mathbb{N}$ budeme typicky značit velikost množiny vrcholu a číslem $m \in \mathbb{N}$ velikost množiny hran. *Ohodnocený* graf je neorientovaný graf $G = (V, E)$ s váhovou funkci $w : E(G) \to \mathbb{R}$. Pro *orientovaný* graf je dvojice $G = (V, E)$, kde $V(G)$ je konečná množina vrcholů a množina hran $E(G)$ se skládá z hran $e = (u, v) \in E(G)$ takových, že u a v jsou v $V(G)$. Napříč této prací budeme uvažovat konečné prosté grafy. Místo toho, abychom psali konečný prostý graf, budeme psát graf.

Mějme množinu vrcholu $B \subseteq V(G)$ grafu $G = (V, E)$. Pro graf $G = (V, E)$ definujme *podgraf* jako graf $G' = (V', E')$ takový, že $V(G') \subseteq V(G)$ a $E(G') \subseteq E(G)$. Graf $G[B] = (B, E')$ je *indukovaný* podgraf grafu G, pokud pro každý dva vrcholy $u \in V(G[B])$ a $v \in V(G[B])$ platí, že $\{u, v\} \in E(G[B])$ právě tehdy, když $\{u, v\} \in E(G)$.

Pro graf $G = (V, E)$ definujme *otevřené* sousedství vrcholu v, značíme $N_G(v)$, jako $N_G(v) := \{u \in V(G) \mid \{u, v\} \in E(G)\}$ Dále zadefinujeme *uzavřené* sousedství vrcholu v pro nějaký graf $G = (V, E)$, značíme $N_G[v]$, jako $N_G[v] :=$

$N_G(v) \cup \{v\}$. *Stupeň* vrcholu v, značíme $\deg_G(v)$, definujme jako $\deg_G(v) := |N_G(v)|$. Množinu $I \in V(G)$ pro nějaký graf $G = (V, E)$ nazveme *nezávislou* množinou, pokud pro každé dva vrcholy $u \in I$ a $v \in I$ platí $\{u, v\} \in E(G)$. *Bipartitní* graf je graf $G = (A \cup B, E)$, kde A a B jsou nezávislé množiny a pro každou hranu $\{u, v\} \in E(G)$ platí $\{u, v\} \subseteq A \times B$.

Cesta P délky $m \in \mathbb{N}$, značíme P_m, je graf $(\{0, \ldots, m\}, \{\{i, i+1\} \mid i \in \{0, m-1\}\})$. Pokud pro cestu P_i, kde $i \in \mathbb{N}$, platí, že $u = v_1$ a $v = v_{i+1}$, pak mluvíme o u-v-cestě neboli o cestě z u do v. O grafu $G = (V, E)$ řekneme, že je souvislý, pokud pro každé dva vrcholy $u \in V(G)$ a $v \in V(G)$ existuje u-v-cesta. *Cyklus* C délky $n \in \mathbb{N}$, nebo prostě cyklus, je graf takový, že $C_n = ([n], E(P_n) \cup \{\{1, n\}\}$. Pokud graf $G = (V, E)$ neobsahuje cyklus jako podgraf, řekneme, že je acyklicky.

Strom je graf $G = (V, E)$ takový, že je souvislý a acyklický. Pro strom $T = (V, E)$ definujme *list*, což je vrchol $u \in V(T)$ takový, pro který platí $\deg_T(u) = 1$. *Kostra* K grafu $G = (V, E)$ je podgraf takový, že K je strom a platí $V(K) = V(G)$. Pro orientovaný graf $G = (V, E)$ řekneme, že je *silně souvislý*, pokud pro každé dva vrcholy existuje orientovaná cesta mezi těmi vrcholy.

▶ **Definice 2.1.** *Mějme graf $G = (V, E)$ s váhovou funkcí $w : E \to \mathbb{N}$. Množinu Q definujme jako obor hodnot váhové funkce. Množině Q budeme říkat různé typy vah. Velikost Q budeme značit q.*

Navíc potřebujeme základní tvrzení o stromech, které později v textu využijeme.

▶ **Tvrzení 2.2** (Postupná výstavba stromů [2])**.** *Pro daný graf G a jeho list v jsou následující dvě tvrzení ekvivalentní.*

(i) *G je strom*

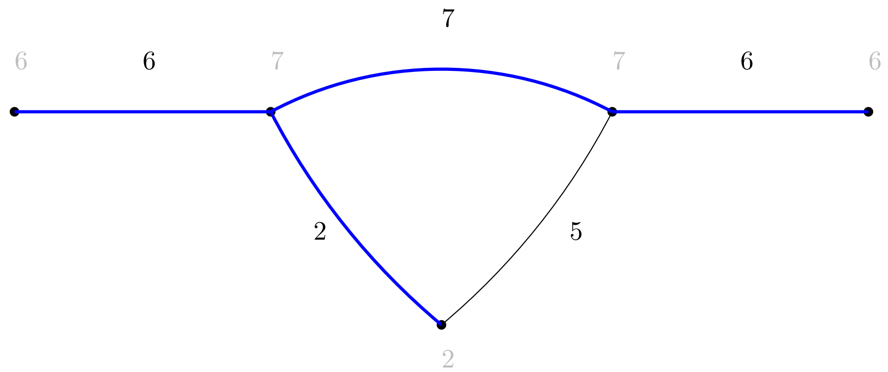
(ii) *$G[G \setminus \{v\}]$ je strom.*

2.2 Definice problému

Hlavní problém, kterým se budeme zabývat, je Min Power Symmetric Connectivity, zkráceně MinPSC. Ve zbytku práce se budeme držet zkrácené verze problému.

▶ **Definice 2.3** (MinPSC [3])**.** *Mějme souvislý ohodnocený graf $G = (V, E)$, který má n vrcholů a m hran. Cílem je najít kostru $K = (V, F)$, $F \subseteq E$, grafu G, která minimalizuje*

$$\sum_{v \in V(G)} \max_{\{u,v\} \in E(K)} w(\{u, v\}).$$



Obrázek 2.1 Instance problému MinPSC s optimalni kostrou vyznačenou tučně modře a u každého vrcholu je šedě jeho ohodnocení

Kostře grafu G, která řeší výše popsaný problém, budeme říkat optimální a hodnotu, kterou kostra nabývá z pohledu účelové funkci, budeme značit $OPT(G)$.

Uvádíme ilustrační příklad, na kterém aplikujeme naši definici.

Na obrázku 2.1 je vidět, že kdybychom chtěli vybírat jenom nejlehčí hrany, a tedy hranu s váhou 7 nevybereme, nemusíme dojít k optimálnímu řešeni. Optimální řešení má cenu 28, ale v případě vybírání nejlehčí hrany bude cena takovéto kostry 29.

Pro naše potřeby budeme chtít využít rozhodovací verzi MinPSC, kde rozhodneme, jestli pro dané k a pro danou instanci I problému MinPSC platí $OPT(I) \leq k$.

Dále budeme potřebovat zavést cenu za kostru, ať už se jedná optimální nebo ne. Intuitivně se jedná o to, kolik celkově zaplatíme za každý vrchol v účelové funkci. Mějme kostru K pro nějaký graf G a funkci $w : K \to \mathbb{N}$, která má předpis

$$w(K) = \sum_{v \in V(G)} \max_{\{u,v\} \in E(K)} w(\{u, v\}).$$

Celková cena je hodnota, kterou nabývá funkce $w(K)$ pro kostru K.

2.3 Parametrizovaná složitost

V této sekci zadefinujme klíčové pojmy z parametrizované složitosti, jejichž definici lze najít v [4]. V odkazu najdeme více detailů.

▶ **Definice 2.4.** Parametrizovaný problém je formální jazyk $L \subseteq \Sigma^* \times \mathbb{N}$, kde Σ je konečná fixní abeceda. Instanci formálního jazyka L budeme značit $(x, k) \in \Sigma^* \times \mathbb{N}$, kde k je parametr.

▶ **Definice 2.5.** O parametrizovanému problému $L \subseteq \Sigma^* \times \mathbb{N}$ řekneme, že je *částečně polynomiální*, anglicky *slice-wise polynomial*, (XP), pokud existuje algoritmus $\mathcal{A}$ a dvě vyčíslitelné funkce f, $g : \mathbb{N} \to \mathbb{N}$ takové, že pro $(x, k) \in \Sigma^* \times \mathbb{N}$ algoritmus $\mathcal{A}$ rozhodne korektně, jestli $(x, k) \in L$ v čase, který je omezen $f(k) \cdot |(x, k)|^{g(k)}$. XP je třída všech částečně polynomiálních parametrizovaných problémů.

▶ **Definice 2.6.** O parametrizovanému problému $L \subseteq \Sigma^* \times \mathbb{N}$ řekneme, že je *fixně parametricky zvládnutelný*, anglicky *fixed-parametr tractable*, (FPT), pokud existuje algoritmus $\mathcal{A}$ a vyčíslitelná funkce $f : \mathbb{N} \to \mathbb{N}$ a konstanta c taková, že pro $(x, k) \in \Sigma^* \times \mathbb{N}$ algoritmus $\mathcal{A}$ rozhodne korektně, jestli $(x, k) \in L$ v čase, který je omezen $f(k) \cdot |(x, k)|^c$. FPT je třída všech fixně parametricky zvládnutelných problémů.

Kapitola 3

Známé výsledky

V této části kapitoly se uvedeme na krátký přehled předchozích výsledků. O problému Min Power Symmetric Connectivity se ví, že je NP-těžký a tedy za běžných předpokladů vylučuje existenci efektivního algoritmu. [5]

Bentert et al. [3] popisují přístupy, jak můžeme získat odhad na spodní ohodnocení vrcholů v každém optimálním řešení. Nabízí se triviální odhad takový, že se podíváme na každý vrchol u a vezmeme nejmenší váhu hrany mezi všemi hranami incidentními s vrcholem u. Více rafinovanější odhad popíšeme následovně. Mějme funkci $\ell : V \to \mathbb{N}$ takovou, že platí $\ell(v)$ je nanejvýš takové, jaké má ohodnocení vrchol v v nějakém optimálním řešení. Definujme předpis pro ℓ následovně

$$\ell(v) = \max_{G' \in C} \min_{u \in V(G')} w(\{u, v\}),$$

kde C je množina souvislých komponent grafu $V(G) \setminus \{v\}$.

Problém MinPSC byla uvažován z hlediska parametru počtu souvislých komponent, kde každá z nich je indukovaná hranami, které musí být ve výsledném řešení. Takový graf budeme označovat jako povinný graf G_l.

▶ **Definice 3.1.** Povinný graf G_l *ohodnoceného grafu* $G = (V, E)$ *indukovaný spodním ohodnocením vrcholů* $\ell : V \to \mathbb{N}$ *se skládá ze všech vrcholů grafu* G *a hran* $\{u, v\}$ *takových, že musí být ve výsledném řešení. Tedy pro každou hranu nutně obsaženou v optimálním řešení platí následující*

$$\min\{\ell(u), \ell(v)\} \geq w(\{u, v\}).$$

Povinný graf nemusí být dán odhadem spodních ohodnocení, které jsou popsané výše. Můžeme zadán jako vstup, kdy senzorové sítě neboli graf ztratil spojení kvůli chybám senzorů [6].

Kromě parametrizace byla uvažovaná řešení s pomocí celočíselného lineárního programovaní [7].

Pro nějaké instance problému MinPSC se může stát, že neumíme nalézt optimální řešení v rozumném čase. A proto může být dobrá volba zkusit řešení aproximovat.

Bentert et al. [3] ukazují, že MINPSC-ALB, MIN POWER SYMMETRIC CONNECTIVITY ABOVE LOWER BOUND, což zadefinujeme záhy, je NP-těžké aproximovat s faktorem $o(\log(n))$.

▶ **Definice 3.2** (MINPSC-ALB [3]). *Mějme na vstupu souvislý neorientovaný graf $G = (V, E)$ s váhovou funkcí $w : E \to \mathbb{N}$. Cílem je najít kostru $T = (V, F)$ takovou, že mininalizuje*

$$\sum_{v \in V(T)} \max_{\{u,v\} \in E(T)} w(\{u, v\}) - \sum_{v \in V(G)} \min_{\{u,v\} \in E(G)} w(\{u, v\}).$$

MINPSC-ALB se dívá na MINPSC z jiného pohledu. Snažíme se minimalizovat rozdíl celkové hodnoty kostry a hodnoty, kde bychom pro každý vrchol vybrali hranu, která má nejmenší váhu a je s ním incidentní.

Mohli bychom se neomezovat na neorientované grafy, tedy na problém MINPSC, můžeme se podívat i na orientované grafy, tedy na problém MIN POWER ASYMMETRIC CONNECTIVITY, zkráceně MINPAC. [8]

▶ **Definice 3.3** (MINPAC [8]). *Mějme silně souvislý graf $G = (V, E)$ a váhovou fuknci $w : E \to \mathbb{N}$. Cílem je najít silně souvislý podgraf H grafu G takový, že minimalizuje*

$$\sum_{v \in V(G)} \max_{(v,u) \in E(H)} w((v, u))$$

Oproti MINPSC se liší tím, že instance problému jsou orientované grafy. Tím, že je na vstupu orientovaný graf, hledáme silně souvislý podgraf místo kostry v MINPSC. Dále je rozdíl v účelové funkci. V případě MINPAC se díváme na hrany, které z vrcholu vycházejí. V případě MINPSC nerozlišujeme, které vycházejí a které vcházejí.

Předtím Chen et al. ukázali, že MINPAC je NP-těžký už pro dvě váhy. [9]

Když se omezíme na dvě různé typy vah, existuje (11/6)-aproximační algoritmus, který pracuje v čase $\mathcal{O}(n^2)$. A také existuje i (9/5)-aproximační algoritmus, který je podstatně méně efektivní. [10]

Jako další uvažovanou parametrizaci zmíníme počet silně souvislých komponent, který budeme značit $c \in \mathbb{N}$. Tento parametr získáme tak, že určíme, které hrany budou obsažené v každém optimálním řešení. Takový graf označme G_l. Podobně jak u MINPSC. A poté se podíváme, kolik komponent silné souvislosti graf G_l má. Existuje algoritmus, který je založen na dynamickém programování a pracuje v čase $\mathcal{O}(c^2 \cdot 2^c \cdot n + m + 4^c \cdot 3^{2 \cdot c - 3/2})$. Jedná se o podobný parametr jako v případě MINPSC. [8]

Další parametry, které uvažují Bentert et al. [8], jsou počet různých vah a vrcholové pokrytí nebo počet orientovaných hran nutných k tomu, abychom vytvořili acyklický graf, anglicky feedback edge number. Pro počet různých vah a vrcholové pokrytí lze aplikovat na vstupní graf redukční pravidlo. A tím dosáhnout toho, že vstupní graf bude mít nejvýše $(q + 1)^{2x} + x$ vrcholů, kde x je velikost vrcholového pokrytí a q je počet různých vah. A pro poslední parametr feedback edge number, označme ho g, lze problém MinPAC vyřešit v čase $\mathcal{O}(2^{\mathcal{O}(g)} + n + m)$.

Kapitola 4

Vrcholové pokrytí

V této kapitole si popíšeme XP algoritmus parametrizovaný vrcholovým pokrytím. Pokud parametr vrcholové pokrytí zesílíme tím, že přidáme počet různých typů vah q, dostaneme FPT algoritmus.

Předtím než si popíšeme algoritmy v této kapitole, zavedeme si značení, kterého se budeme držet v této podkapitole.

4.1 Základní pojmy

▶ **Definice 4.1** (Vrcholové pokrytí)**.** *Mějme graf $G = (V, E)$. Vrcholovým pokrytím nazveme podmnožinu vrcholů M takovou, že pro každou hranu $e = \{u, v\}$ platí, že $u \in M$ nebo $v \in M$.*

V této kapitole budeme vrcholové pokrytí značit M a velikost vrcholového pokrytí d. Dále se podíváme na vrcholy mimo vrcholové pokrytí, které můžeme rozložit do omezeného počtu množin na základě sousedství z M.

▶ **Definice 4.2** ([11])**.** *Mějme graf $G = (V, E)$ a jeho vrcholové pokrytí M. Nechť máme $I = V(G) \setminus M$, která je nezávislá množina. Vytvořme rozklad I, který budeme značit $\mathcal{F}$, a to tak, že pro každou $X \subseteq M$ definujme*

$$I_X := \{v \in I \mid \{v, x\} \in E(G) \text{ pro } x \in X \wedge \{v, x\} \notin E(G) \text{ pro } x \notin X\}.$$

4.1.1 Algoritmus parametrizovaný vrcholovým pokrytím

▶ **Tvrzení 4.3.** *Mějme graf G, jeho vrcholové pokrytí M, velikost vrcholového pokrytí d. Pro každou kostru T grafu G platí, že počet vrcholů n, které nejsou listy v T, tj. jsou vnitřní vrcholy v T, je nejvýše $d + k \leq 2 \cdot d - 1$, kde k je počet vnitřních vrcholů v T mimo M.*

Důkaz. Zvolme nějakou kostru grafu G a označme si ji T. Pro počet vrcholů platí $|V(T)| = d + k + \ell$, kde k je počet vrcholů, které jsou vnitřní vrcholy v T a jsou mimo M, a ℓ je počet vrcholů, které jsou listy v T a leží mimo M. Dále z věty o charakterizaci stromů platí $|V(T)| = |E(T)| + 1 \geq 2 \cdot k + \ell + 1$. Tato nerovnost platí, protože každý vnitřní vrchol v T má alespoň dvě hrany. A takových vrcholů v T je alespoň k. A každý list má právě jednu hranu, kterých je v T alespoň ℓ. Takovýmto způsobem započítáme každou hranu nejvýše jednou. Z výše uvedeného vztahu dostáváme nerovnost $d + k + \ell \geq 2 \cdot k + \ell + 1$, jejíž úpravou dostáváme $k \leq d - 1$. K oběma stranám stačí přičíst d a důkaz je dokončen. □

K tomu, abychom popsali algoritmus, je zapotřebí zadefinovat pojem množina nejtěžších listů.

▶ **Definice 4.4.** *Mějme neorientovaný graf $G = (V, E)$, $w : E \to \mathbb{N}$ a kostru K grafu G. Vezmeme si množinu hran $E' \subseteq E(G)$ takovou, že každá hrana $\{u, v\} \in E'$ nastavuje ohodnocení vrcholům v kostře K. Množině L budeme říkat množině nejtěžších listů, pokud platí*

(i) $L \subseteq \bigcup E'$

(ii) Pro každý $\ell \in L$ platí, že je list v kostře K a nastavuje ohodnocení vrcholu, se kterým je v kostře K incidentní.

(iii) pro každé dva různé vrcholy $u \in L$, $v \in L$ neexistuje vnitřní vrchol, kterého by v kostře K sdílely.

(iv) neexistuje množina $L' \subseteq \bigcup E'$ taková, že L' splňuje (ii) a (iii) bod této definice a přitom $|L| < |L'|$

Myšlenka je taková, že chceme najít takovou množinu L, která zaručí, že hodnocení vnitřních vrcholů nějaké K se nebude měnit. Dále chceme najít pro každý list $\ell \in L$ najít jeho odpovídající vnitřní vrchol, pokud existuje. Nechceme ale brát pro daný vnitřní vrchol všechny takové listy, které nastavují ohodnocení. Stačí nám jeden.

Je vidět z (iii) bodu definice, že mohutnost množiny L je nejvýše $2 \cdot d - 1$, protože počet vnitřních vrcholů může být nejvýše $2 \cdot d - 1$.

Před tím než se pustíme do algoritmu, vysvětlíme si intuici, která za ním stojí. Nejdříve si vybereme vnitřní vrcholy, pro něž najdeme nejtěžší listy. Pak se zbývá vypořádat s ostatními listy, což se dá hladově vyřešit za polynomiální čas.

Teď si popíšeme algoritmus. Mějme graf $G = (V, E)$. Z lemmatu 4.3 víme, že každá kostra má nanejvýš $2 \cdot d - 1$ vnitřních vrcholů. Budeme postupně zkoušet pro každé i z množiny $\{1, \ldots, 2d - 1\}$ všechny volby vnitřních vrcholů z množiny vrcholů grafu G. Na i zvolených vrcholů zkonstruujeme všechny možné kostry. Libovolnou z nich si označme K. Ke kostře K existuje nanejvýš i listů, který ovlivňují ohodnocení i vnitřních vrcholů. Budou vytvářet množinu nejtěžších listů. K této kostře K postupně zvolíme pro každé j z množiny $\{1, \ldots, i\}$ všechny volby j nejtěžších listů, které se budeme pokoušet připojit k vnitřním vrcholům kostry K. Tímto krokem dosáhneme toho, že ohodnocení pro množinu vnitřních vrcholů i a listů z množiny nejtěžších listů se nezmění. Zbývají jen vrcholy, které nejsou ani vnitřní vrcholy a ani nepatří do množiny nejtěžších listů. Zvolme libovolný takový vrchol u. Pokud kostra K neobsahuje žádné vrcholy, a tedy žádné vnitřní vrcholy grafu G, vybereme pro u hranu s nejmenší váhou mezi všemi hranami takový, které jsou s vrcholem u incidentní. Jinak o něm víme, že potřebujeme zvolit jednu hranu e s ním, tj. s vrcholem u, incidentní a také incidentní s vrcholem v ze zvolených vnitřních vrcholů kostry K. Pro vrchol v musí platit, že existuje taková hrana s ním a s u incidentní, která má menší váhu než ohodnocení přiřazené vrcholu v. Vrchol u přípojíme k takovému v, který splňuje vlastnost v předchozí větě a váha hrany mezi nimi je nejmenší možná. Protože výběrem hrany e ohodnocení vrcholu v nemůžeme ovlivnit, můžeme tím ovlivnit jen samotný list. Stačí tedy vybrat hranu e s nejmenší váhou. Ve výsledku dostaneme kostru, kterou porovnáme dle definice problému s dosavadním výsledkem, a vybereme z dvou koster tu, která má menší výslednou váhu. Na konci algoritmu dostaneme optimální kostru.

Algorithm 1 pro MinPSC parametrizovaný vrcholovým pokrytím(VP)

```
function MinPSC(G = (V, E), w : V → ℕ, d)
    K ← ∅
    Pro každé i ∈ {0, ..., 2d − 1}:
        Pro každou kostru K′ na i vrcholech:
            Pro každé j ∈ {0, ..., i} :
                Pro každou volbu nejtěžších listů L o velikosti j:
                    Pro každý nejtěžší list z L připoj k odpovídajícímu
                     vnitřnímu vrcholu K′.
                    Pro každý vrchol u ∈ V(G) \ (L ∪ V(K′)):
                        Vyber vnitřní vrchol v, kterému u nezvýší jeho
                         ohodnocení a zároveň váha je nejmenší možná.
                    Pokud K = ∅ nebo w(K′) < w(K):
                        K ← K′
    vrať K
end function
```

▶ **Tvrzení 4.5.** *Mějme ohodnocený graf $G = (V, E)$s vrcholovým pokrytí M o velikosti d. Výše popsaný algoritmus běží v čase $|V(G)|^{\mathcal{O}(d)} \cdot |E(G)|^{\mathcal{O}(d)}$.*

Důkaz. Volíme množinu vnitřních vrcholů I, kterých je nejvýše $2 \cdot d - 1$ dle lemmatu 4.3. Počet voleb všech vnitřních vrcholů je nejvýše $\sum_{i=1}^{2d-1} \binom{|V(G)|}{i} \leq (2d-1) \cdot |V(G)|^{2d-1}$. Konstrukce kostry K na $|I|$ vrcholech nás stojí $\binom{|E(G[I])|}{|I|-1} \leq |E(G[I])|^{|I|-1} \leq |E(G[I])|^{2d-2} \leq |E(G)|^{2d-2}$. Poté je potřeba ke kostře K vybrat množinu nejtěžších listů L pro nějakou fixní množinu I, což stojí nanejvýš

$$\sum_{i=0}^{|I|} \binom{|V(G)| - |I|}{i} \leq \sum_{i=0}^{|I|} \binom{|V(G)|}{i} \leq |I| \cdot |V(G)|^{|I|} \leq (2d-1) \cdot (|V(G)|)^{2d-1}.$$

Pro každý nejtěžší list najdeme jeho odpovídající vnitřní vrchol, což popíšeme takto $\prod_{i=0}^{|L|-1} (|I| - i) \leq (|I|)! \leq (2d-1)^{2d-1}$.

Zbývá se jenom podívat na vrcholy, které nepatří do kostry K a nejsou nejtěžšími listy. Pro každý takový vrchol se podíváme k jakému vnitřnímu vrcholu v z konstruované kostry můžeme s nejmenší váhou připojit a přitom nezvýšit ohodnocení vnitřní vrcholu v. Celkem to bude $(|V(G)| - |I| - |L|) \cdot |E(G)| \leq |V(G)| \cdot |E(G)|$. Celková složitost je nanejvýš

$$(2d-1) \cdot |V(G)|^{2d-1} \cdot |E(G)|^{2d-2} \cdot |V(G)|^{2d-1} \cdot (2d-1)^{2d-1} \cdot |V(G)| \cdot |E(G)| \\ \leq |V(G)|^{\mathcal{O}(d)} \cdot |E(G)|^{\mathcal{O}(d)}.$$

□

▶ **Tvrzení 4.6.** *Mějme graf $G = (V, E)$, vrcholové pokrytí M a velikost vrcholového pokrytí d. Výše popsaný algoritmus je korektní.*

Důkaz. Mějme optimální řešení H. Nejdříve volíme postupně všechny volby vnitřních vrcholů, kterých je dle lemmatu 4.3 nejvýše $2d - 1$, na kterých vytvoříme kostru. Pro danou volbu vnitřních vrcholů existuje množina nejtěžších listů taková, že každý vnitřní vrchol je sousedem s nejvýše jedním vrcholem z množiny nejtěžších listů. V tomto odstavci zkoušíme všechny možné kombinace. Tím pádem mezi všemi kombinacemi je kombinace, kde jsme zvolili vnitřní vrcholy a nejtěžší listy takové, které jsou v optimálním řešení H.

V posledním kroku algoritmu vybíráme hranu pro každý vrchol u, který jsme neoznačili jako vnitřní vrchol a ani jako nejtěžší list. Pro u hledáme hranu takovou, kterou napojím na zbytek grafu. Můžu napojit na list kostře H nebo na vnitřní vrchol.

Pokud se napojuji na vnitřní vrchol, hledám takový vnitřní vrchol, kterému nezvedne ohodnocení a přitom váha hrany je nejmenší možná. Pro spor předpokládejme, že jsme se mýlili. Existují dva případy, které postupně rozebereme.

Případ 1 Existuje list ℓ v H takový, že se napojí do vnitřního vrcholu, kterému nezvedne ohodnocení, ale váha hrany není nejmenší možná. Náš algoritmus pro každý takový ℓ vezme nejmenší váhu hrany takovou, kde nezvedne vnitřnímu vrcholu ohodnocení. Kromě takových listů ℓ žádný jiný vrchol náš algoritmus neovlivní. Tím vytvoří náš algoritmus řešení H', které je ale lepší než optimální, což je spor.

Případ 2 Existuje list ℓ takový, že se napojí do vnitřního vrcholu, kterému zvedne ohodnocení a váha hrany je nejmenší možná, což je spor s tím, jak jsme postupovali. V předchozím kroku jsme zajistili, že se ohodnocení vnitřních vrcholů nezvýší.

Pokud se ale napojuji na list, což poznám tak, že algoritmus žádnou hranu nevybral pro u, protože by jinak zvýšil ohodnocení jinému vrcholu. V tomhle případě vybereme minimální hranu pro vrchol u, což je korektní, protože každý vrchol má ohodnocení, které se rovná minimální váze hrany mezi hranami, se kterými je incidentní. Jako příklad uvedeme graf se dvěma vrcholy, který je spojen hranou. □

Protože existuje algoritmus, který běží v čase $|V(G)|^{\mathcal{O}(d)} \cdot |E(G)|^{\mathcal{O}(d)}$, patří MinPSC parametrizovaný vrcholovým pokrytím do třídy XP rozhodovacích problémů.

4.2 Algoritmus parametrizovaný vrcholovým pokrytím a počtem různých typů vah

Na začátku si zadefinujeme, jak budeme vytvářet rozklad. Na vstupním grafu vytvoříme třídy na vrcholech mimo vrcholové pokrytí podle sousedů, což jsou vrcholy z vrcholového pokrytí. Navíc nás zajímá pro každý vrchol jeho sousedství ve vrcholovém pokrytí a jaké jsou váhy hran.

Následující definice odráží naše požadavky na faktorizaci. Poznamenejme, že pokud mezi dvěma vrcholy nějakého grafu G neexistuje hrana, doplníme hranu mezi ně nekonečné váhy.

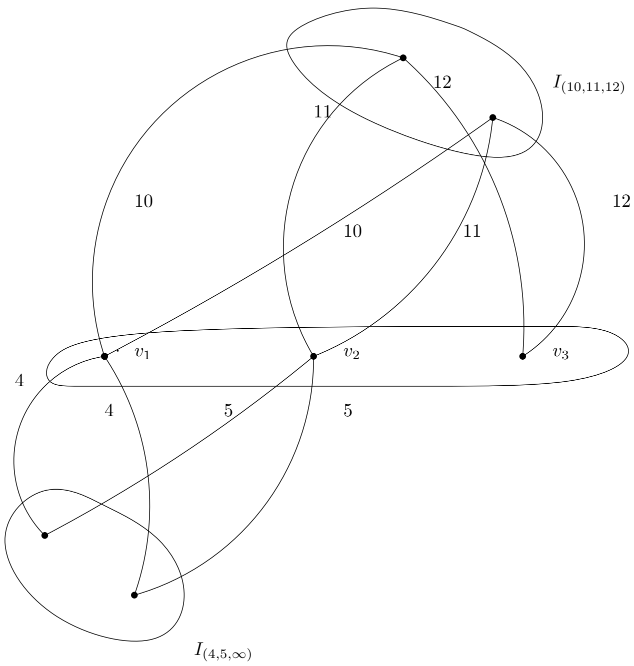
▶ **Definice 4.7.** *Mějme graf $G = (V, E)$, vrcholové pokrytí $M = \{v_1, \ldots, v_d\}$ velikosti d a různé typy vah $Q = \{p_1, \ldots, p_q\}$ a označne $p_{q+1} = +\infty$. Nechť $I = V(G) \setminus M$, která je nezávislá množina. Vytvořme rozklad I, který budeme značit $\mathcal{F}$, a to tak, že pro každé $r_1, \ldots, r_d \in [q+1]$ definujeme*

$$I_{\{(r_1,\ldots,r_d)\}} := \{v \in I \mid \forall i \in [d] : w(\{u, v_i\}) = p_{r_i}\}.$$

Příklad takového rozkladu uvádíme na obrázku 4.1.

Poznamenejme, že třídy popisované ve výše uvedené definici mohou být i prázdné. Na obrázku by to byla třída například $I_{(5,5,5)}$.

Nejdříve nastíníme myšlenku a poté si to popíšeme formálněji. Začneme pozorováním.



Obrázek 4.1 Rozklad grafu G s vrcholovým pokrytím $M = \{v_1, v_2, v_3\}$ a vahami hran z množiny $Q = \{4, 5, 10, 11, 12\}$

.

▶ **Pozorování 4.8.** *Mějme graf $G = (V, E)$. Každý vrchol grafu G má nejvýše tolik různých ohodnocení, kolik je počet různých typu vah.*

Na základě toho pozorování vyzkoušíme všechny volby ohodnocení pro vrcholy z vrcholového pokrytí. Tím dosáhneme toho, že ohodnocení těchto vrcholů se nebude měnit v dalších krocích. Poté budeme aplikovat redukční pravidlo, dokud je to možné. Myšlenka redukčního pravidla je taková, že na konci budeme mít graf, který má počet vrcholů omezený vůči vrcholovému pokrytí a počtu různých typu vah.

Později se budeme dívat na hrany takové, které nezvednou ohodnocení vrcholů z vrcholového pokrytí. Zavedeme si takovou množinu, aby to bylo přehlednější.

▶ **Definice 4.9.** *Mějme graf $G = (V, E)$, vrcholové pokrytí M, různé typy vah Q a f jakožto ohodnocení vrcholů ve vrcholovém pokrytí, tj. $f : M \to Q$ a faktorizaci $\mathcal{F}$. Množinou R_u, kde $u \in F$ pro nějakou třídu $F \in \mathcal{F}$, označíme hrany splňující*

$$\{\{u, v\} \mid v \in M \wedge w(\{u, v\}) \leq f(v)\}.$$

Teď to popíšeme formálněji.

Pro orientované grafy a problém MinPAC existuje redukční pravidlo, které využívá vrcholové pokrytí a počet různých typu vah [8]. Jenomže je pro orientovaný graf.

My v této kapitole vezmeme zmíněné redukční pravidlo pro orientovaný graf a modifikujeme ho pro neorientovaný graf.

Nejprve si zadefinujeme pozorování, které později použijeme.

▶ **Pozorování 4.10.** *Mějme souvislý graf G. Pokud existují vrcholy $u \in V(G)$ a $v \in V(G)$ takové, že $N_G(u) \subseteq N_G(v)$, pak platí, že indukovaný graf G bez vrcholu u, tj. $G[V(G) \setminus \{u\}]$, je souvislý.*

▶ **Redukční pravidlo 4.11.** *Mějme neorientovaný graf G, jeho vrcholové pokrytí M a jeho faktorizaci $\mathcal{F}$. Pro libovolnou třídu $F \in \mathcal{F}$ o velikosti alespoň dva smažeme libovolný vrchol z této třídy.*

Co stojí za povšimnutí, je, že když odstraňujeme vrchol z nějaké třídy, odstraňovaný vrchol se stává listem v konstruované optimální kostře. A za každý list zaplatíme minimální váhu hrany, se kterou je incidentní.

Dále si popíšeme, jak s použitím redukčního pravidla získáme FPT algoritmus. Budeme redukční pravidlo aplikovat, dokud je možné. Kdybychom se podívali na optimální řešení před a po aplikaci redukčního pravidla, zjistíme, že za vrchol, který odstraňujeme, záplatíme minimální váhu hrany takové, že je incidentní s odstraňovaným vrcholem a vrcholem, kde jsme uhodli ohodnocení.

▶ **Tvrzení 4.12.** *Mějme graf G, jeho vrcholové pokrytí M velikosti d a počet různých typů vah q. Redukční pravidlo běží v čase $\mathcal{O}((q+1)^d) \cdot |V(G)| \cdot |E(G)|$.*

Důkaz. Mějme na vstupu instanci I problému MinPSC a proměnnou $c = 0$, do které při každém použití redukčního pravidla budeme ukládat hodnotu vrcholu, který chceme odstranit z instance I. Dokážeme, že potom, co skončíme s aplikováním redukčních pravidel, zbude instance I' problému MinPSC taková, že počet vrcholů I' je nanejvýš $(q+1)^d$, kde q počet typů vah. V instanci mohou nastat z pohledu redukčního pravidla dva případy.

Pokud je počet vrcholů instance I nanejvýš $(q + 1)^d$, vrátím $c = 0$ a instanci I.

Pokud je počet vrcholů instance I alespoň $(q + 1)^d$, aplikujeme redukční pravidlo 4.11, dokud v každé třídě nezbude jeden vrchol. Aplikace probíhá tak, že si připravíme pole D o velikosti $(q + 1)^d$, kde každé políčko odpovídá tomu, jestli je tam už nějaký vrchol z grafu G, anebo ne. Pro každý vrchol $u \in V(G)$ se podíváme, jestli ve třídě, kam vrchol u patří, je nějaký už prvek. Pokud tam není, nastavíme pro tuto třídu hodnotu True. Pokud tam ale je, vytvoříme indukovaný graf $G[V(G) \setminus \{u\}]$. Podíváme jaká váha je minimální v množině R_u. Označme si ji min_u a nastavíme $c = c + min_u$. □

Protože v algoritmu používáme parametr počet různých typu vah, potřebujeme prozkoumat, jestli každá váha se dá omezit. Jinak po skončení algoritmu nelze spolehnout na to, že instance problému MinPSC je omezené v parametrech počtu různých typu vah a vrcholovém pokrytí. Na to, jestli se to dá, odpovídáme kladně a to využitím techniky popsanou v článku.

▶ **Tvrzení 4.13** ([12]). *Pro vstupní instanci $(G = (V, E), w)$ MinPSC, kde $m := |E|$, můžeme spočítat v polynomiálním čase instanci $(G, \hat{w})$ MinPSC takovou, že*

(i) váha nejtěžší hrany v oboru hodnot $\hat{w}$ je omezena nanejvýš $2^{4m^3} \cdot (2m + 1)^m(m + 2)$,

(ii) souvislý podgraf $T = (V, F)$ grafu G je optimální řešení pro (G, w) právě tehdy, když T je optimální řešení pro $(G, \hat{w})$.

▶ **Věta 4.14.** *Mějme ohodnocený graf $G = (V, E)$, vrcholové pokrytí M o velikosti d. Výše popsaný algoritmus pracuje v čase $\mathcal{O}(q^d \cdot (q+1)^d \cdot |V(G)| \cdot |E(G)|)$.*

Důkaz. Abychom zkusili všechny volby pro vrcholy z vrcholového pokrytí grafu G, bude nás to stát q^d. A poté aplikujeme redukční pravidlo, což zvládneme za $(q + 1)^d \cdot |V(G)| \cdot |E(G)|$. A nakonec vyřešíme vzniklou instanci I' velikosti $\mathcal{O}((q + 1)^d + d)$, kterou si označme $n_{I'}$, tj. počet vrcholů. Můžeme použít hrubou sílu, což zvládneme za $\mathcal{O}({n_{I'}}^{n_{I'}-1})$. □

▶ **Tvrzení 4.15.** *Mějme ohodnocený graf $G = (V, E)$, vrcholové pokrytí M o velikosti d. Výše popsaný algoritmus je korektní.*

Důkaz. Mějme optimální řešení H pro graf G. Pro vrcholy z vrcholového pokrytí vybíráme všechny možné volby ohodnocení. Tím zkusíme i ohodnocení odpovídající optimální řešení H.

V dalším kroku algoritmu se díváme na každý vrchol s ve třídě $F \in \mathcal{F}$ o velikosti dva. U vrcholu s volíme, kterou hranou ho připojíme ke zbytku grafu G. Hranu volíme z množiny R_s. Jedná se o podobný krok, jak jsme to dělali v algoritmu 4.1.1.

Nejdříve se zamyslíme, jestli potřebujeme pro vrchol s volit váhu hrany z množiny R_s. Určitě musíme volit hranu z takovéto množině. Kdybychom volili hranu mimo takovouto množinu, zvýšíme ohodnocení vrcholu ve vrcholovém pokrytí, což je spor s tím, že jsme správně nastavili ohodnocení vrcholům ve vrcholovém pokrytí, které jsou v optimálním řešení H.

Označme dva různé libovolné vrcholy s a s' třídy F. Takové dva různé vrcholy existují, protože F obsahuje alespoň dva vrcholy. Předpokládejme, že pro vrcholy s a s' platí $\max_{v \in N_H(s)}(w(\{v, s\}) \leq \max_{v \in N_H(s')}(w(\{v, s'\}))$. Kdyby to neplatilo, upravíme graf tak, že prohodíme všechny incidentní hrany vrcholu s a s'. Formálněji, jen vrcholy s a s' prohodíme a množinu hran změníme a to tak, že

$$E(H) = \{\{v, v'\} \mid \{v, v'\} \in E(H) \wedge \{v, v'\} \cap \{s, s'\} = \emptyset\} \cup \\ \{\{v, s\} \mid \{v, s'\} \in E(H)\} \cup \{\{v, s'\} \mid \{v, s\} \in E(H)\}.$$

Dále můžeme předpokládat, že $N_H(s) \subseteq N_H(s')$. Pokud by to neplatilo, lze pro vrchol s' doplnit hrany s vrcholy $N_H(s) \setminus N_H(s')$. Touto operací nezměníme cenu H, protože za vrchol s' zaplatí už maximální hrana z předpokladu. Pro libovolný vrchol v'' z množiny $N_H(s) \setminus N_H(s')$ platí, že jeho ohodnocení je alespoň $w(\{v'', s\})$. A přidáním hrany $\{v'', s'\}$ se ohodnocení nezmění, protože $w(\{v'', s'\}) = w(\{v'', s\})$. Ostatní vrcholy nijak neměníme.

Z pozorování 4.10 plyne, že $H[V(G) \setminus \{s\}]$ je souvislý. Je acyklický, protože H je acyklický a odstraněním vrcholu acykličnost nezměníme.

Podívejme se na ohodnocení vrcholu s. Označme min_s minimální váhu hrany, která je incidentní s vrcholem s a přitom tato hrana patří do množiny R_s. Odstraněním vrcholu s z grafu G dostáváme graf G'. $H[V(G) \setminus \{s\}]$ je řešení pro graf G' a celkově zaplatíme maximálně $OPT(G) - min_s$, protože u zaplatí alespoň min_s. Tedy dostáváme $OPT(G) \geq OPT(G') + min_s$.

Mějme graf $G' = G[V(G) \setminus \{s\}]$. Mějme optimální kostru H' grafu G'. Ke kostře H' přidáme vrchol s a přidáme takovou hranu, která je v R_s a je nejmenší možnou váhu. Tím, že jsme přidali hranu, jsme nemohli zvednout ohodnocení vrcholu u, který je incidentní s touto hranou. Vrchol u je ve vrcholovém pokrytí, kde každý vrchol má pevně dané ohodnocení. Váhu nově přidané hrany si označme min_s. Tedy vrchol s je list a z věty o trhání listu plyne, že takový graf je strom. Protože obsahuje všechny vrcholu grafu G, jedná se o kostru. A tedy i o řešení, za který zaplatíme nejvýše $OPT(G) \leq OPT(G') + min_s$

Označme min_s minimální váhu hrany takové, která je v R_s. Poté spojením obou nerovností dostáváme

$$OPT(G) = OPT(G') + min_s. \qquad \square$$

Kapitola 5

Sousedská různorodost

V této kapitole si představíme dva parametry. A to sousedskou různorodost a barevnou sousedskou různorodost. Pro barevnou sousedskou různorodost ukážeme, že existuje algoritmus, který běží v čase FPT. Pro sousedskou různorodost dokážeme, že kdybychom chtěli použít tento parametr pro vyřešení MinPSC, zůstává problém MinPSC NP-těžkým.

5.1 Základní pojmy

V této podkapitole si zadefinujeme pojmy, které budeme dále v této kapitole používat.

▶ **Definice 5.1** (Sousedská různorodost)**.** *Mějme graf $G = (V, E)$. Pro dva vrcholy u,v grafu G řekneme, že patří do stejné třídy sousedské různorodosti, pokud $N_G[u] = N_G[v]$. Počet těchto tříd grafu G budeme značit $nd(G)$.*

▶ **Definice 5.2** (Hranově obarvený graf)**.** *Mějme graf $G = (V, E)$ s hranami rozloženými na $E_1, \dots E_k$ tříd pro $k \in \mathbb{N}$ a vytvoříme graf $G' = (V, E' = \bigcup_{i=1}^{k} E_i)$. Grafu G' budeme říkat hranově obarvený graf.*

V definici proměnná k představuje počet barev, která odpovídá počtu různých typu vah neboli oboru hodnot funkce w.

Dále potřebujeme vědět, co to znamená být sousedem podle i-té barvy.

▶ **Definice 5.3** (Barevní sousedí)**.** *Mějme hranově obarvený graf $G = (V, E' = \bigcup_{i=1}^{k} E_i)$ pro počet barev $k \in \mathbb{N}$. Množinu $N_G^j[u]$ vrcholů pro nějaký vrchol $u \in V(G)$, kde $j \in [k]$,definujme jako $N_G^j[u] = \{v \in V(G) \mid \{u, v\} \in E_j(G)\} \cup \{u\}$.*

Sémanticky má předchozí definice význam takový, že množina $N_G^i[u]$ pro nějaký vrchol u grafu G je tvořena sousedy vrcholu u, kteří mají stejnou i-tou váhu hrany, která spojuje daného souseda a vrchol u.

Nakonec zadefinujeme barevnou sousedskou různorodost.

▶ **Definice 5.4** (Barevná sousedská různorodost)**.** *Mějme graf $G' = (V, E' = \bigcup_{i=1}^{k} E_i)$ pro počet barev $k \in \mathbb{N}$. Pro dva vrcholy u, v grafu G' řekneme, že patří do stejné třídy barevné sousedské různorodosti, pokud pro každou barvu $i \in [k]$ platí $N^i_{G'}[u] = N^i_{G'}[v]$.*

Poznamenejme, že pro daný graf G budeme značit $cnd(G)$ velikost barevné sousedské různorodosti, anglicky coloured neighbourhood diversity, tedy počet tříd v barevné sousedské různorodosti.

5.2 Algoritmus parametrizovaný barevnou sousedskou různorodostí

Předtím, než popíšeme redukční pravidlo a dokážeme jeho vlastnosti, popíšeme si hlavní myšlenku, která za ním stojí. Podobně jak v redukčním pravidle pro vrcholové pokrytí 4.11, budeme postupovat i pro obarvenou sousedskou různorodost, kde nejdříve vezmeme libovolný vrchol z každé třídy barevné sousedské různorodosti, u kterého uhodneme ohodnocení. A pak v každé třídě definovanou barevnou sousedskou různorodosti se budeme snažit zredukovat počet vrcholů na jeden. Na všechny zredukované vrcholy krom jednoho se budeme dívat jako na listy v zkonstruované optimální kostře.

Začneme pozorováním.

▶ **Pozorování 5.5.** *Mějme graf $G = (V, E)$. Každý vrchol $u \in V(G)$ má tolik ohodnocení, kolik je tříd barevné sousedské různorodosti, tj. $cnd(G)$.*

V prvním kroku algoritmu z každé třídy barevné sousedské různorodosti vyberme jeden vrchol, u kterého uhodneme jeho ohodnocení. Těch je na základě pozorování 5.5 nejvýše $cnd(G)$. Dále se podíváme hrany, které jsou vybranými vrcholy incidentní. Pokud váha takové hrany je vyšší než váha přiražena vrcholu v předchozím kroku, takovou hranu odstraníme. Poté aplikujeme následující redukční pravidlo.

Následující definice je technická. Chceme všechny vrcholy, které jsme vybrali, seskupit do množiny.

▶ **Definice 5.6.** *Mějme graf $G = (V, E)$ s vrcholy, kterým jsme uhodli ohodnocení, a třídy barevné sousedské různorodosti grafu G. Dále si označme množinu U, do které budou patřit všechny vrcholy, které jsme z každé třídy barevné sousedské různorodosti vybrali.*

Dále bychom chtěli každému vrcholu v přiřadit takový vrchol u, který leží ve stejné barevné sousedské třídě a platí $u \in U$. Takovouto pomocnou funkci označíme $f : V \to V$. Někdy budeme používat i zobecněnou verzi, kdy máme více vrcholů než jeden. Pro neprázdnou množinu vrcholu V definujme $f(V) = \{f(v) \mid v \in V\}$.

▶ **Redukční pravidlo 5.7.** *Mějme graf* $G' = (V, E = \bigcup_{i=1}^{k} E_i)$*, kde* k *je počet barev, s barevnou sousedskou různorodostí* $cnd(G')$*. Pro libovolnou třídu* N *barevné sousedské různorodosti takové, že obsahuje alespoň dva vrcholy, smažeme libovolný vrchol* u *ze třídy* N *takový, že* $u \notin U$*. Ve výsledku dostaneme graf* $G'' = [V(G') \setminus \{u\}]$*.*

▶ **Tvrzení 5.8.** *Mějme graf* $G' = (V, E = \bigcup_{i=1}^{k} E_i)$ *a množinu* U *splňující definici 5.6. Dále mějme graf* G'' *indukovaný množinou vrcholů* $V(G') \setminus \{u\}$*, kde* $u \in V(G')$*,* $u \notin U$ *a* $u \in N$ *pro nějakou třídu barevné sousedské různorodosti velikosti alespoň dva, což je vrchol, který chceme v redukčním pravidle 5.7 odstranit. Označme* $u' \in N$*, který je různý od* u *a* $u' \in U$*. Vztah mezi optimálním řešením grafu* G' *a optimálním řešením grafu* G'' *lze popsat následovně*

$$OPT(G') = OPT(G'') + \min_{v \in f(N_{G'}(u))} (w(\{v, u\})).$$

Předtím než přejdeme na důkaz, všimneme si, že nám stačí pracovat jen s vrcholy z množiny U, protože pro každý vrchol v existuje vrchol $v' \in U$, který leží ve stejné barevné sousedské různorodosti a pro který zároveň platí $w(\{u, v\}) = w(\{u, v'\})$. A poznamenejme, že když se budeme dívat na vrchol sousedy vrcholu u, nás budou zajímat takové vrcholy z U, které jsou ve stejné třídě barevné různorodosti jako sousedi vrcholu u.

Důkaz. Nejdříve dokážeme nerovnost

$$OPT(G') \geq OPT(G'') + \min_{v \in f(N_{G'}(u))} (w(\{v, u\})).$$

Označme si H' jako optimální řešení grafu G'. Bez újmy na obecnosti můžeme předpokládat následující nerovnost.

$$\max_{v \in N_{H'}(u)} (w(\{v, u\})) \leq \max_{v \in f(N_{H'}(u'))} (w(\{v, u'\})).$$

Tato nerovnost říká, že v optimálním řešení H' bude mít vrchol u nanejvýš tak velké ohodnocení, jaké má vrchol u'. Kdyby to neplatilo, vezmeme si množinu všech incidentních hran s vrcholem u v optimálním řešení H' a množinu všech incidentních hran s vrcholem u' v optimálním řešení H'. Tyto dvě množiny prohodíme. Kdybychom to chtěli formálně zapsat, jednalo by se o stejnou myšlenku jako v případě redukčního pravidla 4.11 na bázi vrcholového pokrytí a počtu různých typů vah.

Dále bez újmy na obecnosti můžeme předpokládat, že platí $N_{H'}(u) \subseteq N_{H'}(u')$. Pokud by to neplatilo, lze pro vrchol u' doplnit hrany s vrcholy z množiny $N_{H'}(u) \setminus N_{H'}(u')$. Tímto krokem se $OPT(G')$ nezmění, protože vrcholu u' je přiřazeno ohodnocení, které jsme předpokládali, tedy $\max_{v \in N_{H'}(u)}(w(\{v, u'\}))$. Pro libovolný vrchol v'' z množiny $N_H(u) \setminus N_H(u')$ platí, že jeho ohodnocení je alespoň $w(\{v'', u\})$. A přidáním hrany $\{v'', u'\}$ se ohodnocení nezmění, protože $w(\{v'', u'\}) = w(\{v'', u\})$. Ostatní vrcholy nijak neměníme.

Z pozorování 4.10 plyne, že $H'[V'(G) \setminus u]$ je souvislý. Je zároveň i acyklický. Tudíž se jedná o řešení neboli kostru grafu G''. Celkové ohodnocení takto vzniklého řešení je nejvýše $OPT(G') - \min_{v \in f(N_{H'}(u))}(w(\{v, u\}))$, protože u je ze souvislosti řešení H' ohodnocen alespoň $\min_{v \in N_{G'}(u)}(w(\{v, u\}))$. Z výše uvedeného dostáváme následující nerovnost

$$OPT(G') \geq OPT(G'') + \min_{v \in f(N_{G'}(u))}(w(\{v, u\})).$$

Teď dokážeme opačnou nerovnost.

$$OPT(G') \leq OPT(G'') + \min_{v \in f(N_{G'}(u))}(w(\{v, u\})).$$

Označme si H'' jako optimální řešení grafu G''. Ke grafu H'' připojíme vrchol u s tím, že potřebujeme vybrat hranu. Budeme vybírat hranu nejmenší možné váhy mezi vrcholy z U. Protože obsahuje všechny vrcholy grafu G', jedná se o kostru. A tedy i o řešení, který bude ohodnoceno nanejvýš

$$OPT(G') \leq OPT(G'') + \min_{v \in f(N_{G'}(u))}(w(\{v, u\}))$$

Spojením obou nerovností dostáváme požadovanou rovnost. □

▶ **Tvrzení 5.9.** *Redukční pravidlo lze aplikovat v čase $\mathcal{O}(n \cdot cnd(G))$.*

Důkaz. Mějme na vstupu instanci G a $cnd(G)$ a $c = 0$, do kterého budeme ukládat minimální hodnotu vrcholu, který chceme smazat. Dokážeme, že po aplikování redukčního pravidla dokážeme vytvořit graf G' s tím, že počet vrcholů instance G' je nanejvýš $cnd(G)$. Rozlišíme tři případy.

Pokud bychom měli $V(G) < cnd(G)$, znamenalo by to, že existují třídy barevné různorodosti, kteří nemají žádný vrchol. Takovýto graf jsme museli dostat na vstupu. Z redukčního pravidla jsme ho získat nemohli, protože předpokládá, že ve třídě sousedské různorodosti existují alespoň dva vrcholy a smaže vždy jeden vrchol. Pokud takový graf dostaneme na vstupu, dostáváme spor s definici barevné různorodosti, která nepovoluje prázdné třídy.

Když máme $|V(G)| = cnd(G)$, vrátíme G s tím, že $c = 0$.

Když máme $|V(G)| > cnd(G)$, pak aplikujeme redukční pravidlo, dokud v každé třídě barevné sousedské různorodosti nezbude pouze jeden vrchol z U. Aplikace probíhá tak, že si připravíme pole D o velikosti $cnd(G)$, kde každé políčko má následující význam. „Je tam nějaký vrchol z grafu G, anebo ne?"Tedy pokud je tam nějaký vrchol, tak je tam hodnota True. Pokud je tam není žádný vrchol, je tam hodnota False. Pro každý vrchol $u \in V(G)$ se podíváme, jestli ve třídě barevné různorodosti, kam vrchol u patří, už je nějaký jiný vrchol. Pokud není, tak nastavíme pro tuto třídu barevné různorodosti hodnotu True. Pokud ano, vytvoříme graf $G[V(G) \setminus u]$ a nastavíme $c = c + \min_{v \in f(N_G(u))}(w\{u, v\})$.

Časově zaplatíme $\mathcal{O}(n \cdot cnd(G))$.

Ve výsledku dostáváme graf G', který má v každé třídě barevné různorodosti jeden vrchol, a platí následující vztah $OPT(G) = OPT(G') + c$. □

Poznamenejme, že není potřeba nějakým způsobem omezovat váhy hran, jak jsme to dělali s vrcholovým pokrytím 4.11. V samotné definici barevné sousedské různorodosti jsou zohledněny váhy, neboli barvy, hran, což ukazuje následující pozorování. U sousedské různorodosti stačí, aby třída měla stejné sousedy, ale váhy na hranách mohou mít libovolné, což u barevné sousedské různorodosti je jinak.

▶ **Pozorování 5.10.** *Mějme graf G, typ vah q grafu G a barevnou sousedskou různorodost $cnd(G)$. Typy vah q a barevná sousedská různorodost jsou v následujícím vztahu $q \leq cnd^2(G) + cnd(G)$.*

Důkaz. Pro každou třídu barevné různorodosti platí, že má nejvýše $cnd(G)$ tříd barevné různorodosti, se kterými sousedí. A pro každé dvě třídy sousední barevné různorodosti platí, že mají jeden typ vah.

V rámci jedné třídy barevné různorodosti je nejvýše jeden typ vah. Z toho plyne, že počet různých typů vah je nanejvýš $q \leq cnd^2(G) + cnd(G)$. □

5.3 Barevná sousedská různorodost a sousedská různorodost

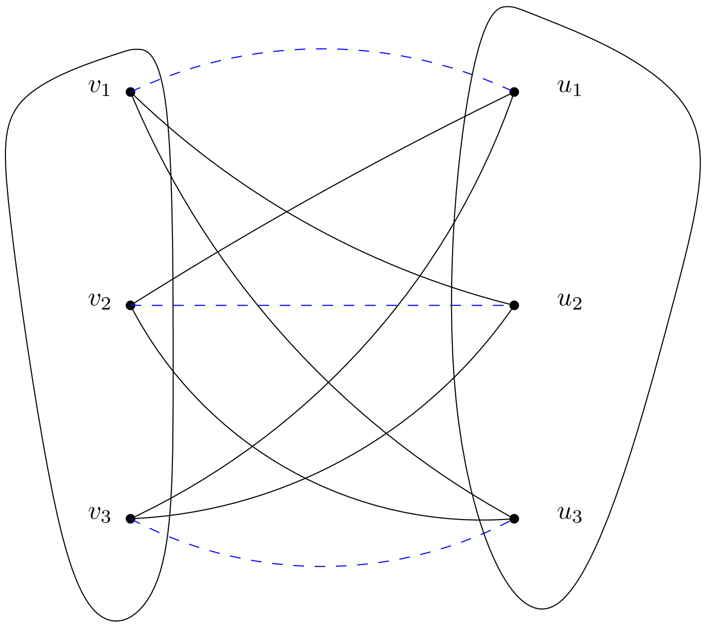
V této podkapitole si ukážeme, že tříd barevné sousedské různorodosti je alespoň tolik, kolik je tříd sousedské různorodosti. Dále si ukážeme graf, kde počet tříd barevné sousedské různorodosti je úměrný počtu vrcholů, ale počet tříd sousedské různorodosti je 2.

▶ **Tvrzení 5.11.** *Mějme souvislý graf $G = (V, E)$ se sousedskou různorodostí $nd(G)$ a s barevnou sousedskou různorodostí $cnd(G)$. Pro barevnou sousedskou různorodost a sousedskou různorodost platí následující nerovnost $nd(G) \leq cnd(G)$.*

Důkaz. Pokud každá třída barevné sousedské různorodosti grafu G je podmnožinou nějaké třídy sousedské různorodosti grafu G, tak nerovnost, kterou chceme dokázat, platí.

Předpokládejme pro spor, že existuje třída barevné sousedské různorodosti F grafu G taková, že je podmnožinou alespoň dvou různých tříd sousedské různorodosti. Dvě různé takové třídy si označme F_1', F_2' a dokážeme, že se muselo jednat o stejnou třídu. Vezměme si dva vrcholy $u \in F_1'$ a $v \in F_2'$. Vrcholy u a v mají stejné sousedství, protože platí $u \in F$ a $v \in F$. Tím pádem musí být ve stejné třídě sousedské různorodosti, což je spor s tím, že jsme předpokládali, že jsou ve dvou různých třídách sousedské různorodosti. □

Můžeme se ptát, na kolik se mohou lišit sousedská různorodost a barevná verze. Následující lemma ukazuje, že i když sousedská různorodost je konstantní, barevná sousedská různorodost může mít až tolik tříd barevné sousedské různorodosti, kolik je počet vrcholů v grafu.



Obrázek 5.1 Bipartitní graf se dvěma partitami o velikosti tři. Modrá hrana symbolizuje váhu hrany 2 a černá hrana symbolizuje váhu hrany 1

▶ **Tvrzení 5.12.** *Pro každý $n \in \mathbb{N}$ existuje ohodnocený graf $G = (V, E)$ takový, že $|V(G)| = 2 \cdot n$, a zároveň platí, že $nd(G) \leq 2$ a $cnd(G) = n$.*

Důkaz. Pro lepší pochopení demonstrujeme nejdříve nakreslení takového grafu 5.1 a poté si ho popíšeme formálněji.

Vezmeme si dvě disjunktní množiny vrcholy o velikosti n. Vrcholy jedné množiny si postupně označíme $v_1, \ldots, v_k$ a druhé množiny sousedské různorodosti označíme $u_1, \ldots, u_k$, kde $k \in \mathbb{N}$ nám udává počet prvků v první třídě a počet prvků ve druhé třídě. Vytvoříme si mezi těmito množinami bipartitní graf $K_{k,k}$ s následující váhovou funkcí

$$w(\{u_i, v_j\}) = \begin{cases} 2 & \text{pokud } i = j \\ 1 & \text{jinak.} \end{cases}$$

Takto zkonstruovaný graf má sousedskou různorodost 2, ale barevnou sousedskou různorodost potřebujeme zjistit. Třída barevné sousedské různorodosti patří právě do jedné sousedské různorodosti. Pro třídu sousedské různorodosti $v_1, \ldots, v_k$ platí, že každé dva různé vrcholy v_i, v_j, kde $i, j \in [k], i \neq j$, platí, že $N^2(v_i) = \{u_i\} \neq \{u_j\} = N^2(v_j)$. Z toho plyne, že $cnd(G) \geq k$, protože by mohla nastat situace, kdy jedna ze tříd barevné různorodosti by měla vrcholy v_j, u_j. Taková situace by znamenala existenci třídy barevné různorodosti, která má jeden vrchol v jedné třídě sousedské různorodosti a druhý vrchol v jiné sousedské různorodosti, což je ve sporu s tím, že každá třída barevné sousedské různorodosti je podmnožinou nějaké třídy sousedské různorodosti, jak jsme ukázali v důkazu lemmatu 5.11. Tedy každá barevné sousedské různorodosti má jeden vrchol. V takovém grafu máme $2 \cdot k$ vrcholů, tedy máme $2 \cdot k$ tříd barevné sousedské různorodosti. □

5.4 Těžkost parametrizace pomocí sousedské různorodosti

Využijme polynomiální redukci z problému MinPSC na MinPSC s konstantním počtem tříd sousedské různorodosti. Hlavní myšlenka je, že libovolnou instanci problému MinPSC lze zúplnit, tj. vytvořit instanci grafu instanci úplného grafu. Tím dosáhneme toho, že počet tříd sousedské různorodosti bude konstantní. V tomto případě konkrétně se bude rovnat jedna. Následující definice formalizuje to, čemu budeme říkat zúplnění.

▶ **Definice 5.13.** *Každý graf $G = (V, E)$ s váhovou funkcí $w : E \to \mathbb{N}$ lze převést, neboli zúplnit, na úplný graf $G' = (V, E')$ s váhovou funkcí $w' : E' \to \mathbb{N}$, kde $E' = \binom{V}{2}$. Mějme $e' \in E'$, $u \in V$ a $v \in V$. Váhová funkce w' je definována následovně*

$$w'(e') = \begin{cases} w(e') & e' \in E(G) \\ \max_{\{u,v\} \in E(G)}(w(\{u, v\}) \cdot |V(G)| & e' \notin E(G). \end{cases}$$

Zúplnění libovolné instance MinPSC sice vytvoří nová řešení, ale optimální zachová, což říká následující pozorování.

▶ **Tvrzení 5.14.** *Mějme hranově souvislý ohodnocený graf G a jeho zúplnění G'. Pro dané grafy platí $OPT(G) = OPT(G')$.*

Důkaz. Mějme optimální řešení H grafu G a optimální řešení H' grafu G'.

H je přípustné řešení zúplněného grafu G, tj. G'. A tedy i platí $OPT(G) \geq OPT(G')$.

A teď to dokážeme opačnou nerovnost, tj. $OPT(G) \leq OPT(G')$. Kdyby optimální řešení H' neobsahovalo žádnou nově přidanou hranu, pak by H' bylo řešením G a $OPT(G) \leq OPT(G')$. Předpokládejme, že H' obsahuje alespoň jednu nově přidanou hranu. Dále si označme Δ váhu hrany takové, která je maximální, tj. $\max_{\{u,v\} \in E(G)} w(\{u, v\})$. Dále víme, že platí následující

$$OPT(G) \leq \Delta \cdot |V(G)| \leq \Delta \cdot |V(G)| \cdot k \leq OPT(G'),$$

kde k je počet nově přidaných hran, které jsou H'. O k víme, že je celé kladné číslo. A tím to máme dokázáno. □

Před tím, než dokážeme redukci, potřebujeme zadefinovat pojem polynomiální redukce neboli Karpova redukce.

▶ **Definice 5.15** (Karpova redukce [13])**.** *Pro rozhodovací problém S a S' mějme funkci $f : S \to S'$ vyčíslitelnou v polynomiálním čase. Funkce f se nazývá Karpova redukce, pokud pro každý x platí $x \in S$ právě tehdy, když $f(x) \in S'$.*

▶ **Věta 5.16.** *MinPSC je NP-úplný i pro grafy s sousedskou různorodosti velikosti 1.*

▶ **Tvrzení 5.17.** *Výše popsaná redukce odpovídá Karpově redukci f z MinPSC do MinPSC s tím, že pro každé $I \in$ MinPSC platí, že $nd(f(I)) = 1$.*

Důkaz. V rámci zúplnění přidáváme neexistující hrany mezi dvěma vrcholy v grafu G s váhou popsanou v definici 5.13 a ostatní hrany zachováme. Tím pádem dostáváme novou instanci MinPSC. Operace přidávání je lineárně závislá na počtu hran, což se stihne v polynomiálním čase. A potom, co zúplníme graf G, vznikne jedna klika, což odpovídá jedné třídě sousedské různorodosti. A přitom z Lemmatu 5.14 plyne, že optimalita mezi původní instancí a nově vzniklou se zachová. □

Kapitola 6

Závěr

Cílem práce bylo prozkoumat problém MinPSC a to z pohledu dvou parametru, vrcholové pokryti a sousedské různorodost.

V kapitole 4 jsme navrhli vzhledem k vrcholovému pokrytí XP algoritmus a s přidanou parametrizací pomocí počtu různých vah jsme vytvořili FPT algoritmus.

Do budoucna se nabízí prozkoumat, jestli bychom dokázali jen pomocí vrcholového pokrytí navrhnout FPT algoritmus, anebo vyvrátit existenci takového algoritmu, tedy ukázat W[1]-těžkost.

V kapitole 5 jsme ukázali, že kdybychom si chtěli pomoci sousedskou různorodostí, zůstává problém MinPSC NP-úplný. Pokud začneme nejen zohledňovat sousedy, jak to je v případě sousedské různorodosti, ale i váhy hran, což odpovídá barevné sousedské různorodosti, dostáváme FPT algoritmus na jednu stranu. Na druhou stranu počet tříd v barevné sousedské různorodosti může být až úměrný počtu vrcholů v grafu, ačkoliv má graf pouze dvě třídy sousedské různorodosti.

Bibliografie

1. MAREŠ, Martin; VALLA, Tomáš. *Průvodce labyrintem algoritmů*. 2. vyd. CZ.NIC, 2022. ISBN 978-80-88168-63-8.
2. MATOUŠEK, Jiří; NEŠETŘIL, Jaroslav. *Kapitoly z diskrétní matematiky*. 5. vyd. Karolinum, 2022. ISBN 978-80-246-5084-5.
3. BENTERT, Matthias; BEVERN, René van; NICHTERLEIN, André; NIEDERMEIER, Rolf; SMIRNOV, Pavel V. Parameterized Algorithms for Power-Efficiently Connecting Wireless Sensor Networks: Theory and Experiments. *INFORMS J. Comput.* 2022, roč. 34, č. 1, s. 55–75. Dostupné z DOI: `10.1287/IJOC.2020.1045`.
4. CYGAN, Marek; FOMIN, Fedor V.; KOWALIK, Lukasz; LOKSHTANOV, Daniel; PILIPCZUK, Marcin; PILIPCZUK, Michal; SAURABH, Saket; MARX, Daniel. *Parameterized Algorithms*. Springer Publishing Company, Incorporated, 2015. ISBN 3319212745.
5. KIROUSIS, Lefteris M.; KRANAKIS, Evangelos; KRIZANC, Danny; PELC, Andrzej. Power consumption in packet radio networks. *Theor. Comput. Sci.* 2000, roč. 243, č. 1-2, s. 289–305. Dostupné z DOI: `10.1016/S0304-3975(98)00223-0`.
6. RODOPLU, Volkan; MENG, Teresa H. Minimum energy mobile wireless networks. *IEEE J. Sel. Areas Commun.* 1999, roč. 17, č. 8, s. 1333–1344. Dostupné z DOI: `10.1109/49.779917`.
7. ALTHAUS, Ernst; CĂLINESCU, Gruia; MANDOIU, Ion I.; PRASAD, Sushil K.; TCHERVENSKI, N.; ZELIKOVSKY, Alexander. Power Efficient Range Assignment for Symmetric Connectivity in Static Ad Hoc Wireless Networks. *Wirel. Networks.* 2006, roč. 12, č. 3, s. 287–299. Dostupné z DOI: `10.1007/S11276-005-5275-X`.
8. BENTERT, Matthias; HAAG, Roman; HOFER, Christian; KOANA, Tomohiro; NICHTERLEIN, André. Parameterized Complexity of Min-Power Asymmetric Connectivity. *Theory Comput. Syst.* 2020, roč. 64, č. 7, s. 1158–1182. Dostupné z DOI: `10.1007/S00224-020-09981-W`.

9. CHEN, Wen-Tsuen; HUANG, Nen-Fu. The strongly connecting problem on multihop packet radio networks. *IEEE Trans. Commun.* 1989, roč. 37, č. 3, s. 293–295. Dostupné z DOI: 10.1109/26.20105.

10. CARMI, Paz; KATZ, Matthew J. Power Assignment in Radio Networks with Two Power Levels. *Algorithmica*. 2007, roč. 47, č. 2, s. 183–201. Dostupné z DOI: 10.1007/S00453-006-1230-1.

11. KNOP, Dušan. *Structural properties of graphs and efficient algorithms: Problems Between Parameters*. Ke Karlovu 2027/3, Praha 2, Nové Město, 121 16 Praha 2, 2018. Disertace. Univerzita Karlova, Matematicko-fyzikální fakulta.

12. BENTERT, Matthias; BEVERN, René van; FLUSCHNIK, Till; NICHTERLEIN, André; NIEDERMEIER, Rolf. Polynomial-time data reduction for weighted problems beyond additive goal functions. *Discrete Applied Mathematics*. 2023, roč. 328, s. 117–133. ISSN 0166-218X. Dostupné z DOI: 10.1016/j.dam.2022.11.018.

13. GOLDREICH, Oded. *Computational Complexity: A Conceptual Perspective*. Cambridge University Press, 2008. ISBN 9781139472746.