FAKULTA DOPRAVNÍ

Bc. Jan Šlégl

Řešení barvení grafů metodou celočíselného programování

Diplomová práce

2017

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
Fakulta dopravní
d ě k a n
Konviktská 20, 110 00 Praha 1

K617..... Ústav logistiky a managementu dopravy

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení studenta (včetně titulů):

Bc. Jan Šlégl

Kód studijního programu a studijní obor studenta:

N 3710 – LO – Logistika, technologie a management dopravy

Název tématu (česky): **Barvení grafů metodou celočíselného programování**

Název tématu (anglicky): Binary Programming Approach towards Graph Colouring Problem

Zásady pro vypracování

Při zpracování diplomové práce se řiďte osnovou uvedenou v následujících bodech:

- problematika barvení grafu, defektivní barvení grafu
- aplikace defektivního barvení grafu v dopravní praxi
- rešeršní část
- formulace problému, omezující podmínky, složitost úlohy
- řešení úlohy pomocí celočíselného lineárního programování - sestavení matematického modelu, softwarová implementace, metodologie testování modelu na vybraných grafech
- řešení úlohy exaktním a heuristickými přístupy s časovým omezením
- interpretace výsledků testování
- porovnání výsledků exaktního a heuristických přístupů navrženého modelu s výstupy algoritmů dalšího autora

ČVUT FD

Rozsah grafických prací: podle pokynů vedoucí diplomové práce

Rozsah průvodní zprávy: minimálně 55 stran textu (včetně obrázků, grafů a tabulek, které jsou součástí průvodní zprávy)

Seznam odborné literatury:

NIEUWOUDT, I. a J. H. VAN VUUREN. Algorithms for a shared resource scheduling problem in which some level of conflict is tolerable

BRADLEY, Stephen P, Arnoldo C HAX a Thomas L MAGNANTI. Applied mathematical programming.

WINSTON, W. L. (1994). Operations research:

Vedoucí diplomové práce: **doc. Ing. Denisa Mocková, Ph.D.**

doc. Ing. Josef Volek, CSc.

Datum zadání diplomové práce: **30. června 2015**
(datum prvního zadání této práce, které musí být nejpozději 10 měsíců před datem prvního předpokládaného odevzdání této práce vyplývajícího ze standardní doby studia)

Datum odevzdání diplomové práce: **30. května 2017**
a) datum prvního předpokládaného odevzdání práce vyplývající ze standardní doby studia a z doporučeného časového plánu studia
b) v případě odkladu odevzdání práce následující datum odevzdání práce vyplývající z doporučeného časového plánu studia

L. S.

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE Fakulta dopravní

doc. Ing. Lukáš Týfa, Ph.D.
vedoucí
Ústavu logistiky a managementu dopravy

prof. Dr. Ing. Miroslav Svítek, dr. h. c.
děkan fakulty

Potvrzuji převzetí zadání diplomové práce.

Bc. Jan Šlégl
jméno a podpis studenta

V Praze dne ..... 21. prosince 2016

PODĚKOVÁNÍ

Tato práce by nebyla vytvořena, nebýt několika lidí, kterým bych tímto chtěl velmi poděkovat. Asi největší zásluhu na současné podobě této diplomové práce má prof. JH van Vuuren z Stellenbosch University, se kterým jsem měl tu čest spolupracovat, a který mi byl velkou inspirací a oporou. Díky pomoci prof. Van Vuurena a několika studentů z výzkumného týmu SuNORE se mi podařilo úspěšně vytvořit matematický model a implementovat ho do potřebného softwaru, ke kterému jsem dostal jako člen tohoto týmu přístup. Dále bych rád poděkoval své rodině a nadaci „Nadání Josefa, Marie a Zdenky Hlávkových“, bez jejichž podpory by má práce na Stellenbosch University nebyla možná. Na závěr bych rád poděkoval vedoucím práce doc. Ing. Denise Mockové, Ph.D. a doc. Ing. Josefu Volkovi, CSc. za jejich přínosné připomínky a rady ohledně podoby této práce.

PROHLÁŠENÍ

Nemám závažný důvod proti užívání tohoto školního díla ve smyslu § 60 Zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon).

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.“

V Praze dne 28. května 2017 …….....

podpis

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

Fakulta dopravní

Barvení grafů metodou celočíselného programování

A Binary Programming Approach towards Graph Colouring Problem

diplomová práce
květen 2017
Bc. Jan Šlégl

ABSTRAKT

Diplomová práce je zaměřena do oblasti aplikací a řešení *maximum degree colouring problem*. Exaktní algoritmické přístupy řešící tento problém jsou náročné z hlediska výpočetního času. V této diplomové práci je řešeno, zda je použití celočíselného programování vhodnou alternativou řešení problému. Zjišťujeme, že tento přístup je schopný exaktního řešení grafů s velikostí do cca 400 hran. Použití modelu z této s přidáním časového omezení je shledáno jako velmi efektivní co do poměru výpočetního času a přesnosti.

Klíčová slova

Defektivní barvení grafu, celočíselné programování, teorie grafů, matematický model

ABSTRACT

This paper is focused on a application and solving the so-called maximum degree colouring problem. Exact algorithmic approaches towards solving this problem exactly are very costly in terms of computation time. We investigate whether a binary programming approach towards solving this problem is a viable. We found that this approach is able to solve the maximum degree colouring problem in a exact way until the approximate size of the graph of 400 edges. The model with time limit approach is found to be very efficient in matter of computation time and accuracy.

Keywords

Defective graph colouring, Integer programming, Graph Theory, Mathematical model

OBSAH

Seznam použitých zkratek 7
1 Úvod 8
2 Klasické barvení grafů 10
2.1 Historie „problému 4 barev“ 10
2.2 Problém klasického barvení vrcholů grafu 11
2.3 maximum degree colouring problem 11
2.4 Základní pojmy používané při barvení grafů 12
3 Rešerše 15
3.1 Exaktní algoritmy 17
3.2 Heuristické algoritmy 19
4 Formulace problému 24
5 Využití klasického a defektivního barvení grafů 27
5.1 klasické barvení grafu - Sestava letových plánů 27
5.2 Přerozdělování přístupu uživatelů k souborům ve sdílené síti 28
5.3 Rekonstrukce sítě metra 31
5.4 Komunikace satelit-pozemní přijímač 35
6 Testované grafy 37
7 Metodologie 41
8 Testování matematického modelu 42
8.1 řešení problému MDCP použitím modelu s časovým omezením 43
8.2 Exaktní metoda pro řešení MDCP 47
8.3 Testování náhodně generovaných grafů 50
8.4 Srovnání s ostatními algoritmy 55
9 Závěr 61
Literatura 65
Seznam obrázků 68
Seznam Tabulek 69

SEZNAM POUŽITÝCH ZKRATEK

AMACOL - The adaptive memory colouring algorithm

CCA - Critical χ_d^{Δ} – colouring algorithm

CDA – Colouring degree algorithm

DIMACS – Centre for Discrete Mathematics and Theoretical Computer Science

ICA – χ_d^{Δ} irredundant–colouring algorithm

MDCP – Maximum degree colouring problem

MFTDMA - Multi-Frequency Time-Division Multiple Access

TABUCOL – tabu search for graph colouring

TDMA - time Division Multiple Access

TO – Time out

TSA – Tabu search algorithm

1 ÚVOD

Teorie grafů je součástí matematické disciplíny operačního výzkumu, ve kterém jsou využívány matematické postupy k řešení úloh z prostředí logistiky, ekonomiky, či úloh procesních. Grafy samotné si lze představit jako matematické struktury, definované množinami vrcholů a hran. Hrany představují spojnici mezi dvěma vrcholy. Graf může znázorňovat například města, která jsou vzájemně propojená komunikacemi. Tyto komunikace vždy začínají a končí v jednom z měst. Teorie grafů zahrnuje velké množství různých úloh, pro které již bylo nalezeno bezpočet různých postupů řešení. Jedním z důvodů studia teorie grafů je množství problémů a praktických úloh, ve kterých mohou být postupy teorie grafů použity a implementovány.

Grafem lze zjednodušeně interpretovat velké množství zadání úloh, které by jinak byly jen velmi obtížně řešitelné. Vrcholy grafu mohou představovat lidské bytosti, vozidla či uživatele počítačové sítě. Hranami grafu pak lze reprezentovat například státní hranice či sociální a jiné abstraktní vazby mezi lidmi, či různými subjekty.

Barvení grafů, kterým se tato práce zabývá, je jednou z často zkoumaných oblastí teorie grafů. V úlohách zabarvovacího typu jsou vrcholům nebo hranám přiřazovány různé barvy. Přiřazování barev je možné pouze za dodržování předem určených omezení a pravidel. Pravidla barvení vrcholů pak tvoří podstatu výsledného zabarvení grafu. Jedno z nejznámějších a často používaných pravidel je omezení, které nedovoluje dvěma sousedním vrcholům zabarvení stejnou barvou. Úloha s tímto pravidlem byla poprvé zmíněna již v první polovině devatenáctého století ve formulaci takzvaného „problému 4 barev“. Tento problém následně zůstával nevyřešen po více než sto let.

Tato práce je věnována problematice barvení grafu obsahující omezení, které dovoluje vrcholu zabarvenému určitou barvou sousedit nanejvýše s b vrcholy zabarvenými toutéž barvou.

Úloha se v literatuře nazývá *maximum degree colouring problem* (dále jen MDCP), nebo éž volně přeloženo do češtiyxn defektivní barvení grafu. Řešící postupy je možné aplikovat na různé praktické úlohy, jejichž příklady jsou blíže popsány v kapitole 5. Za zmínku stojí například aplikace defektivního barvení grafu při řešení úlohy rekonstrukce stanic metra,

která může probíhat jen za přesně stanovených omezujících podmínek tak, aby nebyla narušena dopravní obslužnost města.

Pro řešení problému defektivního barvení grafu již bylo vytvořeno několik exaktních i heuristických algoritmů, z nichž některé se ukázaly jako velmi účinné. Hlavním přínosem této práce by měl být nový způsob, kterým je hledáno řešení úlohy. Zatímco všechna dosavadní řešení jsou dosažena použitím algoritmů, autorovi se nepodařilo najít práci, ve které by bylo defektivní barvení grafu řešeno pouze prostřednictvím celočíselného lineárního programování. V kapitole 4 je představen matematický model, který je možno řešit metodou celočíselného lineárního programování. Hlavní výhodou takového přístupu je možnost použití některých velmi účinných solverů, které na valnou většinu ostatních algoritmů aplikovat nelze. Model lze řešit simplexovou metodou. Jedním z nejlepších softwarů specializovaných na řešení úloh lineárního programování je IBM ILOG CPLEX Optimalization Studio. Matematický model představený v této práci je implementován právě do tohoto solveru.

V kapitole 8 jsou variace modelu testovány na různých skupinách grafů. Tyto grafy se liší počtem vrcholů, hran, hustotou hran či pravidelností struktury grafu. Jsou zaznamenávána a vyhodnocována výsledná zabarvení, přesnost a časová efektivita modelu. Veškeré výsledky jsou podrobně popsány a uvedeny v tabulkách. V poslední části této kapitoly jsou pak tyto výsledky porovnány s výstupy několika jiných algoritmů a je provedeno jejich zhodnocení.

Cílem této práce je přinést nový pohled na způsoby řešení defektivního barvení grafu a vytvořit obecný matematický model, který bude možno jednoduše upravovat a implementovat dle charakteru a požadavků různých zadání.

2 KLASICKÉ BARVENÍ GRAFŮ

2.1 HISTORIE „PROBLÉMU 4 BAREV“

Historie klasického barvení grafů je velmi pestrá. Za počátek této matematické disciplíny se považuje úloha, jejímž cílem bylo nalezení minimálního počtu barev nezbytných k zabarvení zemí na politické mapě tak, aby žádné dvě sousedící země nebyly zabarveny toutéž barvou. Tato úloha byla formulována v teorému 4 barev, o kterém se jako první zmiňuje v jedné ze svých přednášek A. F. Mobius (1790-1868) [1]. Teorém 4 barev říká, že libovolná dvourozměrná mapa může být zabarvena použitím čtyř barev tak, že každá dvě ohraničená území sdílející společnou hranici budou zabarvena jinou barvou. Tento problém se také někdy označuje jako Guthrievův problém, podle studenta, který s teorémem přišel a obrátil se s ním právě na profesora Mobiuse. [2]

Vytvořit příklad mapy, která by se skutečně dala zabarvit pouze čtyřmi barvami, jistě není problém. Nicméně dokázat, že na zabarvení libovolné politické mapy postačí čtyři barvy, se ukázalo jako velmi obtížné. Významný geometr Felix Klein (1849-1925) se například nechal slyšet, že jediný důvod, proč tento problém dosud nebyl vyřešen je, že se jím dosud žádný význačný matematik nezabýval. Nicméně ani Klein v řešení problému neuspěl. [3]

V roce 1879 byl v časopise Americal Journal of Mathematics zveřejněn důkaz hypotézy 4 barev, jehož autorem byl A. B. Kempe (1845-1922). Tento důkaz však byl po jedenácti letech zpochybněn P. Headwoodem (1861-1955), který v něm po důkladném zkoumání nalezl chybu. Nicméně Headwood objevil, že tzv. „Kempeho řetěz“ (sekvence zemí, které alternují mezi právě dvěma barvami) postačuje k důkazu, že každá politická mapa může být zabarvena právě pěti barvami. Neboli že chromatické číslo je rovno nanejvýše 5, ale může nabývat hodnot 4 a méně. Toto tvrzení se začalo nazývat jako hypotéza pěti barev. Během konce devatenáctého století bylo nalezeno ještě několik dalších důkazů, avšak všechny byly po čase zpochybněny a vyvráceny. [3]

V roce 1974 autoři Kenneth Appel (1932-2013) a Wolfgang Haken (1928) konečně dokázali pravdivost teorému 4 barev. Jejich důkaz popisoval, že mapu na povrchu jakékoli koule lze zabarvit nanejvýše 4 barvami. Výpočet tohoto důkazu zabral celkem 1200 strojových hodin na superpočítači University of Illinois. Technika řešení spočívala v identifikaci 633 fundamentálních konfigurací map, do kterých mohla být redukována

libovolná jiná mapa, a důkazu, že každá z těchto konfigurací může být redukována do konfigurace jednoduššího typu. Problémem však zůstávalo ověření výsledku. Výpočet totiž z časových důvodů nemohl být nepřetržitý. Následující rok byla v důkazu nalezena chyba algoritmu, která byla později opravena. Publikace byla nakonec vydána v roce 1976 a teorém 4 barev byl od té doby pokládán za vyřešený. V průběhu následujících let byly v algoritmu nalézány další chyby a nesrovnalosti, které se však postupně odstraňovaly dalšími úpravami algoritmu. Profesor Paul Seymour se svou skupinou z Princeton University publikovali v roce 1994 článek, v němž popisují kompletně jiný přístup k řešení tohoto problému. Algoritmus se jeví jako stabilnější a výpočet není tak náročný. Dodnes však není popsán důkaz, ke kterému by nebylo za potřebí výpočetní síly.

2.2 PROBLÉM KLASICKÉHO BARVENÍ VRCHOLŮ GRAFU

Klasické barvení grafu je základní úlohou disciplíny barvení grafů. V úloze je zadaný graf G s množinou hran E a vrcholů V, a množina barev C, kterými mají být jednotlivé vrcholy zabarveny. Cílem úlohy je zabarvení každého vrcholu grafu právě jednou barvou tak, že žádná dvojice vzájemně sousedících vrcholů nesmí být zabarvena stejnou barvou. Výsledek zapisujeme jako množiny vrcholů obarvených stejnou barvou, či graficky znázorníme v grafu. Každá z barev množiny C použitých k zabarvení grafu pak indukuje skupinu stejně zabarvených vrcholů tvořící podgraf grafu G. V případě klasického barvení grafu je tento podgraf vždy diskrétní.

2.3 MAXIMUM DEGREE COLOURING PROBLEM

MCDM je variací klasického problému barvení grafů. V úloze je zadán graf G s množinou hran E a vrcholů V, a množina barev C, kterými mají být jednotlivé vrcholy zabarveny. Cílem úlohy je pak zabarvit všechny vrcholy tak, že každý vrchol $v \in V$ zabarvený barvou $c \in C$ sousedí nanejvýše s b vrcholy zabarvenými toutéž barvou. Každá barva použitá k zabarvení grafu G pak indukuje skupinu vrcholů tvořících podgraf grafu G. Stupeň libovolného vrcholu v podgrafu indukovaném některou z použitých barev je pak vždy menší či roven b. Klasický problém barvení grafu je tedy speciálním případem MDCP s hodnotou $b = 0$. Stupeň každého vrcholu v podgrafu indukovaného některou

z použitých barev pak může nabývat pouze hodnoty 0 a výsledkem je stejně jako v případě klasického barvení grafů diskrétní podgraf.

2.4 ZÁKLADNÍ POJMY POUŽÍVANÉ PŘI BARVENÍ GRAFŮ

Zabarvením grafu $G(V, E)$ rozumíme přiřazení barev $c \in \mathbb{C}$ vrcholům z množiny V, přičemž ke každému vrcholu v je přiřazena právě jedna barva c tak, aby bylo dodrženo stanovené pravidlo barvení $\mathcal{R}$. Množina $\mathbb{C}$ reprezentuje množinu použitelných barev.

Množiny vrcholů zabarvené stejnou barvou nazýváme **barevné třídy** C_i kde $i = 1, \ldots, k$. Každá z těchto k tříd, kde k reprezentuje počet barev použitých k zabarvení grafu G, tedy obsahuje množinu vrcholů. Použití k barev k zabarvení grafu G je nazýváno ***k*-zabarvení grafu** G. V případě defektivního barvení grafů je zabarvení grafu označováno $(\boldsymbol{b}, \boldsymbol{k})$**-zabarvení grafu**.

Stupeň vrcholu $\boldsymbol{st(v_i)}$ označuje počet hran incidujících s vrcholem v_i.

Matice sousednosti neorientovaného grafu $G(V, E)$ je celočíselná matice $A = \left(a_{i,j}\right)_{i,j=1}^{n}$, kde $n = |V|$. Prvky matice mohou nabývat hodnot:

- $a_{i,j} = 0$ pro $i = j$,
- $a_{i,j} = 0$ když neexistuje hrana incidující zároveň s vrcholy v_i a v_j,
- $a_{i,j} = 1$ když existuje hrana incidující zároveň s vrcholy v_i a v_j.

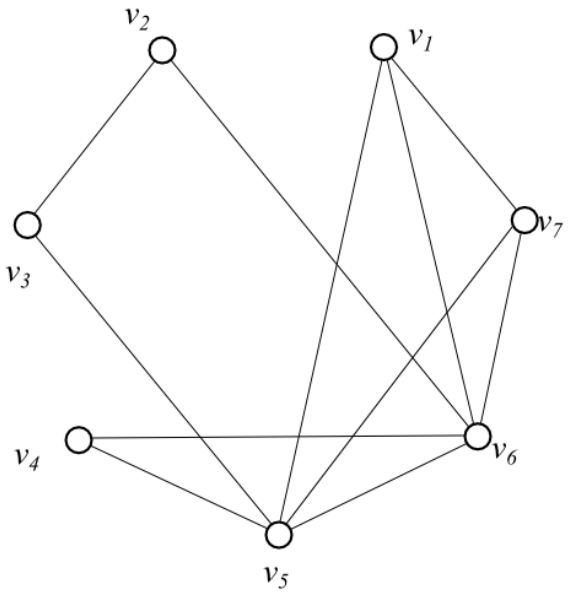
Barevný stupeň vrcholu v z grafu G určuje počet barev, kterými jsou zabarveny vrcholy sousedící s vrcholem v.

Kompletní zabarvení grafu znamená, že každý vrchol $v \in V$ grafu $G(V, E)$ je obarven barvou $c \in C$.

Částečným zabarvením se pak rozumí přiřazení barev $c \in C$ podmnožině vrcholů $S \subset V(G)$.

Klikou se nazývá kompletní graf, který je zároveň podgrafem. Řád kliky pak popisuje počet vrcholů kompletního podgrafu. Na Obrázku 1 je pro příklad zobrazen graf J, který obsahuje hned několik klik. Největší z nich je klika čtvrtého řádu tvořená vrcholy $\{v_1, v_5, v_6, v_7\}$. Tyto čtyři vrcholy a jejich incidující hrany tedy spolu tvoří kompletní graf

o 4 vrcholech. Graf J dále obsahuje 5 klik třetího řádu, a to jmenovitě: $\{v_1, v_5, v_7\}; \{v_1, v_6, v_7\}; \{v_4, v_5, v_6\}; \{v_5, v_6, v_7\}; \{v_1, v_5, v_6\}$. Kliky druhého a prvního řádu není zapotřebí vypisovat.



Obrázek 1 - Graf J – příklad klik v grafu. (Zdroj: autor)

Chromatické číslo grafu představuje nejnižší možný počet barev, kterými lze zabarvit stanovený graf G. Chromatické číslo obecně označujeme jako $\chi(G)$. V případě řešení problému MDCP pak symbolu přidáváme horní index Δ. Chromatické číslo grafu zabarveného dle MDCP je označováno jako $\chi_b^{\Delta}(G)$. Horní index obecně označuje příslušnost chromatického čísla ke specifické úloze zabarvování grafu. Dolní index obsahuje hodnotu b, ke které je chromatické číslo vztaženo.

Δ-chromatická sekvence grafu G je sekvencí chromatických čísel reprezentujících počet barev k, potřebných k χ_b^{Δ} zabarvení grafu G, kde b představuje rostoucí parametr. Sekvence tedy vypadá následovně:

$$\chi_{b_0}^{\Delta}(G), \chi_{b_1}^{\Delta}(G), \chi_{b_2}^{\Delta}(G), \ldots$$

Celá chromatická sekvence je pak sekvencí neklesající. Δ-chromatická sekvence je obecně ukončována, jakmile některý člen klesne na hodnotu $\chi_b^{\Delta}(G) = 1$. První člen chromatické sekvence je tedy roven chromatickému číslu daného grafu v případě použití klasického barvení grafu, zatímco poslední člen této sekvence představuje tentýž graf zabarvený pouze jednou barvou

Řád grafu $\boldsymbol{i}(\boldsymbol{G})$ značí počet vrcholů grafu.

Velikost grafu $\boldsymbol{q}(\boldsymbol{G})$ představuje počet hran grafu**.**

Stupeň barevnosti dosud nezabarveného vrcholu v stanovuje celkový počet barev, kterými jsou obarveny vrcholy s ním sousedící.

Hustota hran $\boldsymbol{h}(\boldsymbol{G})$ značí poměr počtu hran $|E|$ neorientovaného grafu G a počtu hran kompletního grafu stejného řádu.

Hustotu hran neorientovaného grafu G s počtem vrcholů $|V|$ lze vypočítat dle následujícího vzorce:

$$h(G) = \frac{|E_G|}{|E_{K_n}|} \tag{2.1}$$

Jelikož platí že

$$|E| = \binom{|V|}{2} = \frac{|V|(|V|-1)}{2} \tag{2.2}$$

Můžeme vzorec upravit do podoby

$$h(G) = \frac{2 * |E|}{|V|(|V|-1)} \tag{2.3}$$

3 REŠERŠE

K tématu defektivního barvení grafu se vztahuje velké množství odborné literatury. Defektivní barvení grafů bylo poprvé představeno téměř současně v osmdesátých letech minulého století v publikacích autorů Andrews and Jacobson [4], Cowen et al. [5], and Harary and Jones [6]. Cowen a Woodall [5] se ve své práci soustředili především na grafy vložené do povrchů různých těles a popsali kompletní charakteristiku všech k a b takových, že každý planární graf je (k, b)-zabarvitelný. Tedy, že každý graf lze v úloze MDCP zabarvit při stanoveném b zabarvit minimálně k barvami.

Teorémem 4 barev, který byl dokázán autory K. Appel a W. Haken, můžeme stanovit, že každý planární graf je (4,0)-zabarvitelný.

Teorém 1

Pro každý planární graf G platí, že $\chi(G) \leq 4$

V roce 1994 stanovili Frick a Henning [7] následující horní hranici χ_b^{Δ} pro libovolný graf G.

Teorém 2 Frick a Henning *[7]*

Pro libovolný graf G s maximálním stupněm Δ a libovolným $b \in \mathbb{N}_0$ *platí*

$$\chi_b^{\Delta}(G) \leq [(\Delta + 1)/(\mathrm{b} + 1)]$$

Teorém 4 nám umožňuje určit přibližnou horní hranici počtu barev potřebných k (b, k)-zabarvení libovolného grafu G pomocí maximálního stupně grafu G. Množinu barevných tříd nutných k zabarvení libovolného grafu v úloze MDCP lze použitím teorému vyjádřit jako:

$$\mathbb{C} = \left\{1, \dots, \left\lceil \frac{\Delta + 1}{b + 1} \right\rceil \right\} \tag{3.1}$$

Stejní autoři [7] pak také stanovili následující nezbytné podmínky, které musí splňovat každá Δ- *chromatická sekvence* grafu:

***Teorém 3 Frick a Henning** [7]*

Pokud jsou $x_0, x_1, x_2, \ldots$ hodnoty, reprezentující počet barev nezbytné k $\chi_0^\Delta, \chi_1^\Delta, \chi_2^\Delta, \ldots$ zabarvení grafu G, pak pro Δ-chromatickou sekvencí grafu G platí že

a) $x_n = 1$

b) $x_j \leq x_i \leq x_j \left[\frac{j+1}{i+1}\right]$ *pro všechna* $0 \leq i \leq j$

Platnost Teorému 3 byla dokázána, nicméně se zatím v žádné práci nepodařilo dokázat, zda jsou tyto podmínky dostatečné pro stanovení jakékoli Δ-*chromatické sekvence*.

O Δ-*chromatické sekvenci* libovolného grafu pak také můžeme tvrdit, že je sekvencí nerostoucí, viz. Teorém 4.

Teorém 4

Pro libovolný graf G a jakékoli $b \in N_0$ platí, že

$$1 \leq \chi_{b+1}^\Delta(G) \leq \chi_b^\Delta(G) \leq \chi(G)$$

Teorém 4 je využit v kapitole 8, kde je uvedeno několik Δ-*chromatických sekvencí*, které Teorém 4 nesplňují. Z toho lze odvodit nesprávnost těchto sekvencí.

Frick [8] se ve své dizertační práci zabýval mimo jiné barvením kompletních grafů pomocí metody MDCP. V této práci lze také nalézt důkaz pro následující tvrzení:

Teorém 5 Frick a Henning [8]

Pro všechna $b \in \mathbb{N}_0$ a všechna $i \in \mathbb{N}$ platí že

$$\chi_b^\Delta(K_i) = [i/(b+1)]$$

Z tohoto teorému můžeme odvodit, že jakýkoli kompletní graf lze zabarvit použitím jediné barvy pouze v případě, že $b \geq i - 1$.

Isabelle Niewoudt [9] ve své práci provedla důkaz tvrzení popisujícího zabarvení grafů typu kružnice. Následující teorém uvádí, že graf typu kružnice lze v případě klasického barvení grafů zabarvit použitím 2 barev v případě, že graf je tvořen sudým počtem vrcholů. V případě, že je graf tvořen lichým počtem vrcholů, jeho k jeho zabarvení zapotřebí minimálně 3 barev. Teorém dále stanovuje počet barev nutný k zabarvení libovolného grafu typu kružnice pro různé hodnoty b.

Teorém 6 Isabelle Niewoudt [9]

Pro všechna $b \in \mathbb{N}_0$ *a všechna* $i \in \mathbb{N}$ *platí že*

$$\chi_b^{\Delta}(C_i) = \begin{cases} 1 \ \ když\ b \geq 2 \\ 2\ když\ b = 1\ nebo\ když\ (b = 0\ a\ \ i\ \ je\ sudé) \\ 3\ když\ b = 0\ a\ \ i\ \ je\ liché \end{cases}$$

3.1 EXAKTNÍ ALGORITMY

Dle Magaluti [10] byl jeden z prvních exaktních algoritmů pro problém zabarvení vrcholů grafu navrhnutý Brownem [11]. Tento algoritmus obecně pracuje s postupným zabarvováním vrcholů, přičemž se vždy snaží vrchol zabarvit nejprve tou barvou, která již byla použita pro zabarvení některého z předchozích vrcholů. Pokud to není možné, používá algoritmus k zabarvení vrcholu barvu novou.

K zabránění generování velkého množství nadbytečných zabarvení, tedy neoptimálních a redundantních, jsou v algoritmu použity takzvané *„prověřovací techniky"*. Představme si, že vrchol v_k je vybrán ke generování nového částečného zabarvení z již existujícího částečného zabarvení $\mathcal{B}$ grafu G s n vrcholy. Vrchol v_k by mohl být zabarven jednou z dostupných barev $c \in \mathbb{C}$. *Prověřovací techniky* algoritmu sníží počet barev, kterými tento vrchol bude moci být zabarven na základě 3 aspektů. Tím budou eliminována nadbytečná řešení, která jsou pro cíl úlohy nadbytečná. Prvním z aspektů je eliminace redundantních barev dle následujícího teorému:

Teorém 7 Brown *[11]*

Nechť je $\mathbb{C}$ *množinou všech barev. Nechť jsou* $\mathbb{C}_u$ *a* $\mathbb{C}_n$ *barevnými podmnožinami množiny* $\mathbb{C}$*, pro které platí* $\mathbb{C}_u \cup \mathbb{C}_n = \mathbb{C}$*. Podmnožina* $\mathbb{C}_u$ *obsahuje barvy použité při částečném zabarvení* $\mathcal{B}$ *grafu G, kdežto podmnožina* $\mathbb{C}_n$ *obsahuje barvy, které při částečném zabarvení* $\mathcal{B}$ *použity nebyly. Chceme-li generovat nové částečné zabarvení z* $\mathcal{B}$*, zabráníme redundantnímu částečnému zabarvení použitím* $(\mathbb{C} \cup \{\alpha\}) \subseteq \mathbb{C}$*, kde* α *představuje libovolnou barvu z podmnožiny* $\mathbb{C}_n$*.*

Druhý aspekt eliminuje barvy, jejichž použitím by mohlo dojít k porušení pravidla $\mathcal{R}$ pro barvení grafu.

Třetí aspekt *prověřovacích technik* říká, že pokud již bylo nalezeno přípustné zabarvení grafu k barvami, pak v dalších hledaných zabarveních nesmí být použito více než $k-1$ barev.

Pro Brownův algoritmus je pořadí zabarvování vrcholů určeno předem. V tomto pořadí sousedí zabarvovaný vrchol v_j , $j \in 1, \dots, n$ s více vrcholy z $\{v_1, \dots, v_{j-1}\}$, nežli s vrcholy z množiny $\{v_{j+1}, \dots, v_n\}$. Toto pořadí je jedním ze vstupů algoritmu.

Hansen a kolektiv [12] přišli v roce 2005 s takzvanou *Set Packing* formulací jiného základního matematického modelu pro zabarvování vrcholů grafu. Původní model je založen na zabarvování diskrétních podmnožin vrcholů v grafu. Účelová funkce pak minimalizuje celkový počtu diskrétních podmnožin vrcholů. Původní model pracuje s nezávislými množinami vrcholů v grafu, kterým přiřazuje barvy a zaznamenává, zda jsou všechny vrcholy v nezávislé množině obarveny stejnou barvou:

$$\min \sum_{s \in S} x_s \,, \tag{3.2}$$

$$\sum_{s \in S : i \in s} x_s \geq 1 \,, \tag{3.3}$$

$$x_s \in \{0; 1\} \quad s \in S \,.$$

(kde $s \in S$ představuje jednotlivé diskrétní podmnožiny vrcholů grafu G a x_s je binární proměnná, která udává, zda jsou všechny vrcholy množiny s zabarveny pouze jednou barvou, či nikoli)

Set packing formulace tento původní model upravuje do následující podoby:

$$max \sum_{s \in \Omega} (|s| - 1) x_s, \tag{3.4}$$

$$\sum_{s \in \Omega : i \in s} x_s \leq 1 \quad i \in V, \tag{3.5}$$

$$x_s \in \{0,1\} \quad s \in \Omega \,.$$

Kde $\Omega = \{s \in S : |s| \geq 2\}$, tedy množina všech nezávislých množin s velikostí větší než 2. Pokud bychom pak výslednou hodnotu spočtenou modelem označili z, byl by počet barev nutný k zabarvení grafu roven $k = n - z$.

Dvě práce autorů Méndéz Díaz a Zabala [13] pracují s jinou formulací matematického modelu určeného k zabarvení vrcholů grafu. Ta vypadá následovně:

$$min \sum_{k=1}^{n} y_k \ , \tag{3.6}$$

$$\sum_{k=1}^{n} x_{ik} = 1 \quad i \in V \ , \tag{3.7}$$

$$x_{ik} + x_{jk} \leq y_k \quad (i,j) \in E, k = 1, \dots , n \ , \tag{3.8}$$

$$x_{ik} \in \{0;1\} \quad i \in V, k = 1, \dots , n \ ,$$

$$y_k \in \{0;1\} \quad i \in V, k = 1, \dots , n \ .$$

Podstata tohoto modelu je podobná jako podstata modelu představeného v této práci. y_k reprezentuje binární proměnnou, která indikuje, zda byla použita barva k. Proměnná x_{ik} pak udává, zda byl vrchol v_i obarven právě barvou k. Účelová funkce minimalizuje počet barev potřebných k zabarvení grafu. Hlavním nedostatkem tohoto modelu je však výběr barev. Je-li k zabarvení grafu použito k barev z celkového počtu n, je počet možných kombinací barev $\binom{n}{k}$. Méndéz Díaz a Zabala [13] proto k modelu přidávají novou omezující podmínku, která zajišťuje snížení počtu těchto kombinací. Podmínka stanovuje, že nová barva může být použita pouze v případě, že byla použita barva, která ji v pořadí barev předchází:

$$y_k \geq y_{k+1} \quad k = 1, \dots , n-1. \tag{3.9}$$

Touto podmínkou mnohdy dochází k významnému poklesu možných kombinací zabarvení, což samozřejmě ovlivní rychlost výpočtu.

3.2 HEURISTICKÉ ALGORITMY

První heuristické algoritmy pro zabarvování vrcholů grafu byly různé variace hladových algoritmů. Princip těchto algoritmů spočívá v postupném obarvování vrcholů dle předem určeného pravidla pro výběr pořadí vrcholů a barvy použité k zabarvení. Výhoda těchto algoritmů je výpočetní doba, která je však vykoupena sníženou přesností a citlivostí na vstupní údaje.

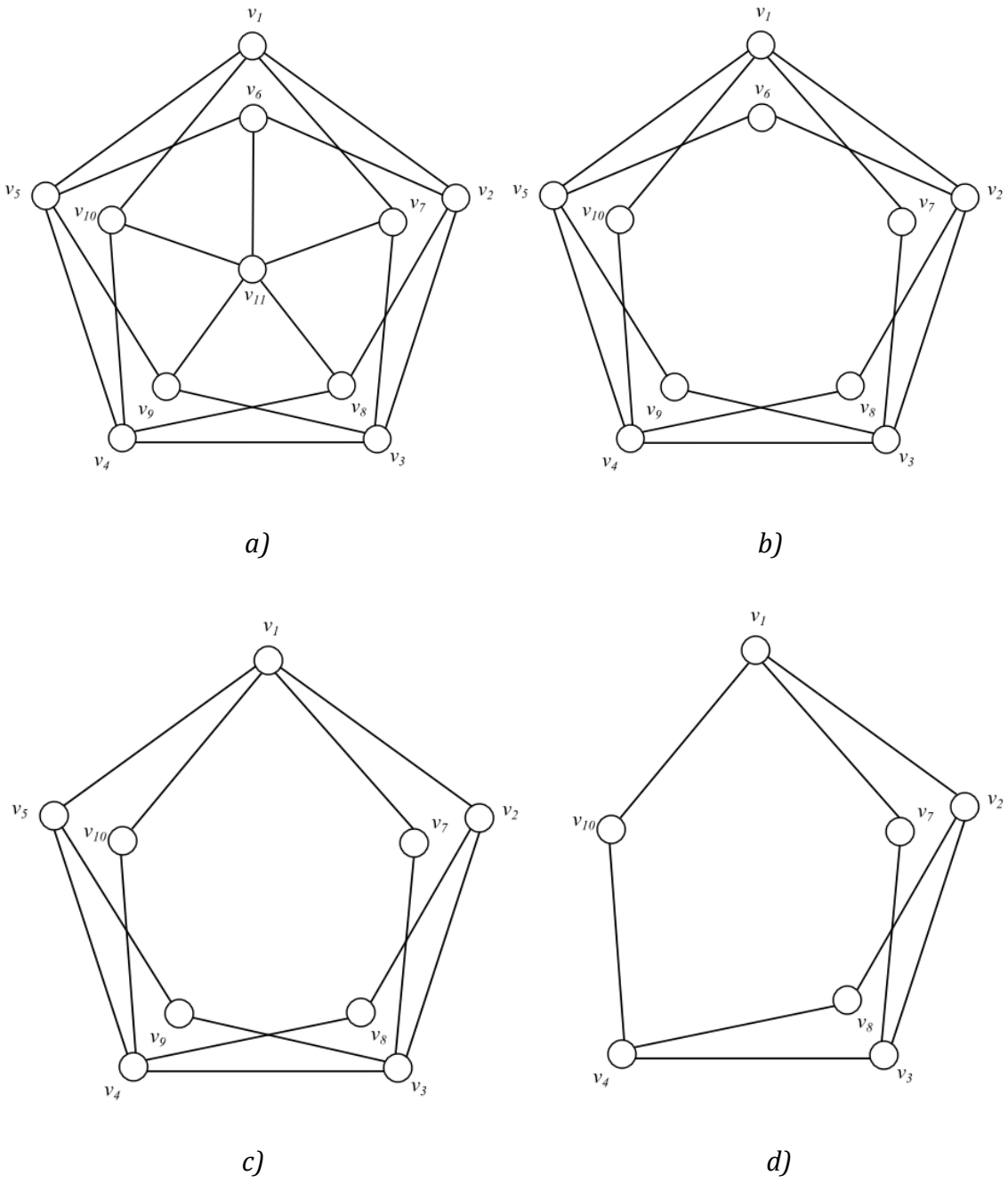
Jedním z nejjednodušších hladových algoritmů je takzvaný hladový sekvenční algoritmus. Pro funkčnost tohoto algoritmu jsou jednotlivé vrcholy grafu G označeny číslicemi

vzestupně a poté dle pořadí zabarvovány. První je tedy zabarven vrchol v_1 použitím barvy první barevné třídy a po něm všechny ostatní vrcholy v_i $(i = 2, \dots, n)$. Pro zabarvování se vždy používá nejnižší přípustná barevná třída.

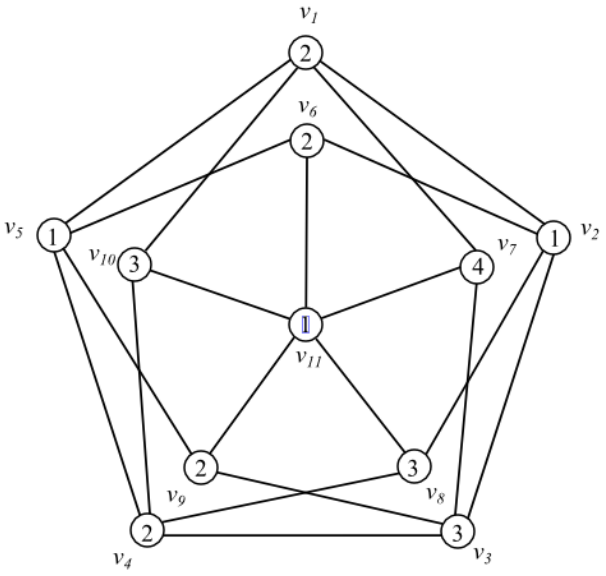
S variací hladového algoritmu přišel v roce 1979 Brélaz [14] Postup je založený na postupném zabarvování vrcholů dle jejich stupně barevnosti. Stupeň barevnosti vrcholu udává počet všech barev, kterými jsou zabarveny vrcholy s ním sousedící. Na začátku každé iterace je vybrán vrchol grafu s nejvyšším stupněm barevnosti vrcholu a následně zabarven první přípustnou barvou. Pokud je však vrchol přeřazen z množiny neobarvených vrcholů do některé z barevných tříd, změní se tím stupně zabarvenosti všech sousedních vrcholů. V následující iteraci je pak vybrán další vrchol z množiny nezabarvených vrcholů, mající nejvyšší stupeň zabarvenosti vrcholu. Proces zabarvování se opakuje do doby, kdy jsou zabarveny všechny vrcholy grafu.

Na Obrázku 2 je ukázán příklad postupu barvení Grotzchova grafu použitím této heuristiky. Graf na Obrázku 2a zobrazuje graf v podobě, kdy všechny vrcholy náleží do množiny dosud nezabarvených vrcholů G_0 . V prvním kroku algoritmu je zvolen vrchol s největším stupněm v_{11}, kdy $st(v_{11}) = 5$, a zařazen do barevné třídy C_1. Na začátku druhého kroku je množina dosud nezabarvených vrcholů ponížena o v_{11} a jsou odstraněny veškeré hrany s ním incidující, zároveň se však zvýší stupeň zabarvenosti všem vrcholům sousedícím s v_{11}. Graf po prvním kroku tedy bude vypadat viz. Obrázek 5b. Nyní mají vrcholy $v_6, \dots, v_{10}$ shodně nejvyšší stupeň zabarvenosti. Můžeme tedy k zabarvení vybrat libovolný z nich. Zvolíme a zabarvíme vrchol v_6. Množina dosud neobarvených vrcholů je ponížena, a na začátku třetího kroku graf vypadá viz. Obrázek 2c. Nyní mají vrcholy v_2 , v_5 mezi vrcholy s nejvyšším stupněm zabarvení nejvyšší stupeň vrcholu $st(v_2, v_5) = 3$. Jeden z těchto dvou vrcholů může být zabarven. Zvolme vrchol v_5. V následujícím kroku je jediný vrchol s nejvyšším stupněm zabarvenosti v_9 a tudíž musí být zabarven jako další. Množina neobarvených vrcholů po těchto dvou krocích je zobrazena na Obrázku 5d.

Tímto postupem je postupně zabarven celý graf, který je včetně přiřazených barev zobrazen na Obrázku 3.



Obrázek 2 - Použití Brélazova algoritmu (Zdroj: autor)



Obrázek 3 – Graf zabarvený Brélazovým algoritmem (Zdroj: autor)

Jiným přístupem k zabarvení vrcholů grafu heuristickou metodou je použití metaheuristických postupů. Pro zabarvování vrcholů grafu se nejčastěji používá takzvaná *local search* metaheuristika. Postup této metody vychází z počátečního zabarvení, které je následně optimalizováno a je hledán nejlepší možný výsledek. Algoritmus samotný se pak většinou skládá ze dvou částí. První část algoritmu stanoví počáteční přípustné řešení (zabarvení grafu). Druhá část algoritmu se pak snaží počáteční zabarvení modifikovat s cílem snížit počet barev použitých k zabarvení vrcholů grafu. Algoritmy se dají rozdělit dle způsobu funkčnosti na ty, které zabarvují celý graf s použitím konstantního počtu barev, a ty, které také zabarvují celý graf nebo jeho části, avšak s použitím proměnného počtu barev.

Jeden z prvních metaheuristických algoritmů byl takzvaný TABUCOL, pracující s procedurou *Tabu search,* představený v práci autorů Hertz a Werra [15]. Tento algoritmus používal konstantní počet barev určených ke kompletnímu zabarvení daného grafu. Počáteční zabarvení grafu však u tohoto algoritmu nemusí nutně splňovat pravidla zabarvení grafu (například že žádná dvojice vzájemně sousedících vrcholů nesmí být zabarvena jednou barvou). V následující části algoritmu jsou vrcholy jeden po druhém postupně přebarvovány, přičemž po každém přebarvení vrcholu je měřen počet konfliktů (nesplnění podmínek zabarvení). Algoritmus však přebarvuje pouze konfliktní vrcholy,

tedy ty, u kterých není splněna podmínka zabarvování. Současně jsou v algoritmu zaznamenávány barvy, kterými již byly jednotlivé vrcholy zabarveny. To z důvodu, aby nedocházelo k replikaci vyšší, než je pevně stanovený počet iterací. Toto je zajišťováno seznamem vrcholů, do kterého jsou zaznamenávány kombinace vrcholů a použitých barev. TABUCOL je mezi metaheuristickými barvícími algoritmy velmi známý a byl několikrát upraven a vylepšen dalšími autory.

Jiným metaheuristickým přístupem je použití metody simulovaného žíhání. Tato metoda patří do stochastických simulačních algoritmů se základy ve fyzice. Johnson et al. [16] použili metodu simulovaného žíhání, ve které jsou srovnávána 3 různá zabarvení, kdy jedno splňuje podmínky zabarvení, a zbylá dvě podmínky splňovat nemusí. Počet barev potřebných k zabarvení grafu zde funguje jako proměnná a algoritmus zabarvuje celý graf. Jakmile je metaheuristikou nalezen přípustný počet barev pro zabarvení grafu, který je nižší než předem stanovená hodnota, je graf zabarven exaktní metodou. Tato kombinace metaheuristiky a exaktního přístupu se ukázala jako velmi konkurenceschopná.

Davis [17] se pokusil vytvořit účinný barvící čistě genetický algoritmus, který by řešení transformoval do permutace vrcholů zabarvovaných sekvenčním hladovým algoritmem. Naneštěstí se však čistě genetické postupy neprokázaly jako dostatečně účinné pro problém zabarvování grafů.

V populačně založených algoritmech jsou jednotlivá řešení, která se vytváří během průběhu algoritmu, ukládána do „paměti“ algoritmu. Obecně tyto algoritmy ukládají informaci o kompletním zabarvení grafu. Algoritmy s adaptivní pamětí jsou však schopny ukládat pouze částečná zabarvení, která se jeví jako perspektivní. Galinier [18] vytvořil algoritmus pojmenovaný AMACOL, pracující s konstantním počtem použitelných barev, který ukládá nezávislé množiny zabarvených vrcholů, získané během zabarvování grafu. Tyto množiny pak jsou použity ke generování nových řešení.

4 FORMULACE PROBLÉMU

Předpokládejme, že chceme zabarvit graf G s počtem vrcholů n a množinou vrcholů $V(G) = \{v_1, \dots, v_n\}$. Stupeň každého vrcholu v_i v grafu G označíme jako $st(v_i)$ a maximální stupeň grafu jako $\Delta = max_i\{st(v_i)\}$.

Vstupem do modelu jsou:

- počet vrcholů grafu n,
- matice sousednosti A grafu G,
- hodnota veličiny b.

K vytvoření matematického modelu je potřeba formulovat dvě binární proměnné veličiny. Vezmeme-li z matice sousednosti grafu G člen $a_{i,j}$ z řádku i a sloupce j, můžeme definovat binární rozhodovací proměnnou jako:

$$x_{jc} = \begin{cases} 1 & \text{pokud je vrchol } v_j \text{ zabarven barvou } c \\ 0 & \text{v opačném případě} \end{cases}$$

Proměnná x_{jc} zaznamenává, do které barevné třídy vrchol v_j náleží.

Pomocná binární proměnná y_c je definována následovně:

$$y_c = \begin{cases} 1 & \text{pokud je barva } c \text{ použita} \\ 0 & \text{v opačném případě} \end{cases}$$

Pro všechna $j = \{1, \dots, n\}$ a všechna $c \in \mathbb{C}$.

Tato proměnná určuje, zda byla barva c použita k zabarvení alespoň jednoho vrcholu grafu G. y_c je součastí účelové funkce.

Podmínkou MDCP je, aby žádný subgraf indukovaný barevnou třídou neobsahoval vrchol stupně $b + 1$ pro specifické $b \in \mathbb{N}_0$. Účelová funkce pak minimalizuje počet barev nutných k zabarvení grafu:

$$\min Z^{\Delta} = \sum_{c \in C} y_c \,, \tag{4.1}$$

za dodržení následujících omezujících podmínek:

$$\sum_{j=1}^{n} a_{ij}x_{jc} \;\leq\; (d_i - b)(1 - x_{ic}) + b \qquad \text{pro všechna } i \in \{1, \dots, n\} \text{ a } c \in \mathbb{C} \qquad (4.2)$$

$$\sum_{j=1}^{n} x_{jc} \;\leq\; ny_c \qquad \text{pro všechna } c \in \mathbb{C} \qquad (4.3)$$

$$\sum_{c\in\mathbb{C}} x_{jc} \;=\; 1 \qquad \text{pro všechna } j \in \{1, \dots, n\} \qquad (4.4)$$

$$x_{jc}, y_c \;\in\; \{0,1\} \qquad \text{pro všechna } j \in \{1, \dots, n\} \text{ a } c \in \mathbb{C}$$

Podmínka (4.2) zajišťuje, že pokud je vrchol v_i obarven barvou c (tedy pokud $x_{ic} = 1$), pak nanejvýše b sousedních vrcholů v_j je taktéž zabarveno barvou c. Suma v levé straně rovnice postupně prochází všechny vrcholy v grafu. Pokud vrchol v_j sousedí s vrcholem v_i, pak se dle matice sousednosti $a_{ij} = 1$. Pokud je tento vrchol v_j navíc zabarven barvou c, pak se taktéž $x_{jc} = 1$. Součin v sumě $\sum_{j=1}^{n} a_{ij}x_{jc}$ je roven 1 pro všechny vrcholy sousedící s vrcholem v_i a zabarvené identickou barvou jako v_i. Celková suma pak dle nerovnosti musí být nižší nebo rovna pravé straně rovnice. Pravá strana rovnice zajišťuje následující: Pokud je vrchol v_i zabarven barvou c, pak se $x_{ic} = 1$ a součin závorek je nulový. V tom případě se celá pravá strana rovnice rovná b, což požadovaný maximální počet sousedních vrcholů zabarvených stejnou barvou jako vrchol v_i. Pokud ale vrchol v_i barvou c zabarven není, pak je pravá strana rovnice rovna stupni vrcholu v_i. V tomto případě může být barvou c zabarven libovolný počet sousedních vrcholů. Tato podmínka platí pro všechny vrcholy grafu a všechny barvy z množiny barev $\mathbb{C}$.

Omezující podmínka (4.3) slouží ke stanovení použití barvy c. Pokud je alespoň jeden vrchol grafu obarven barvou c, pak je suma v druhé podmínce nenulová. Pravá strana nerovnice pak nabývá pouze hodnoty rovné počtu vrcholů v grafu, tedy n, nebo je nulová. Tato podmínka jen zdánlivě neplní svůj účel. Vezmeme-li jako příklad graf G, ve kterém není žádný z vrcholů zabarven barvou $c \in \mathbb{C}$, pak bude podmínka splněna i v případě, že proměnná y_c značící použití barvy c bude nabývat hodnoty 1. Tedy, že barva c použita byla. Tento případ je však odstraněn účelovou funkcí (4.1), která má za úkol minimalizovat celkový počet použitých barev. Indikátor použité barvy y_c je tedy nenulový jen v případě, že suma v levé straně druhé omezující podmínky je taktéž nenulová.

V opačném případě je účelovou funkcí snížena hodnota y_c na nulu. Podmínka je platná pro všechny barvy z množiny barev $\mathbb{C}$.

Třetí omezující podmínka (4.4) stanovuje, že každý vrchol v_j může být zabarven právě jednou barvou $c \in \mathbb{C}$. Tato podmínka je platná pro všechny vrcholy grafu.

V celém modelu je celkově $(n+1)[(\Delta+1)/(b+1)]$ binárních proměnných a $(n+1)[(\Delta+1)/(b+1)]+n$ netriviálních omezujících podmínek.

Model v této podobě splňuje kritéria lineárnosti a tudíž ho lze řešit metodami celočíselného lineárního programování, jako je například simplexová metoda.

5 VYUŽITÍ KLASICKÉHO A DEFEKTIVNÍHO BARVENÍ GRAFŮ

Disciplína barvení grafů má široké pole působnosti co do využití při řešení úloh z mnoha naprosto odlišných sektorů. Cílem řešení je ve všech případech minimalizace zadané proměnné. Barvení grafů je jedním z nástrojů operačního výzkumu, který se zabývá převážně úlohami s účelovou funkcí, tedy minimalizací či maximalizací dané proměnné. V následujících odstavcích je popsáno několik úloh, ve kterých lze využít klasické či defektivní barvení grafů.

5.1 KLASICKÉ BARVENÍ GRAFU - SESTAVA LETOVÝCH PLÁNŮ

V následující úloze použijeme klasické barvení grafů. Letecká společnost plánuje rozšíření stávající nabídky pravidelných letů. Všechny lety mají pevně stanovený den a čas, ve kterém jsou vykonávány. Letecká společnost dále disponuje daným počtem letadel, kterými je schopna všechny lety pokrýt. Cílem úlohy je zjistit, zda je stávajícím počtem letadel možno pokrýt navýšený počet letů, či zda je zapotřebí pořízení dalších letadel. V případě nutnosti zakoupení dalších letadel je pak zjišťován minimální nezbytný počet strojů.

Předpokládejme, že se všechny lety periodicky opakují. Ke každému z letů musí být vždy přiřazeno právě jedno letadlo. Každý let má dále pevně stanovený časový úsek, za který má být proveden. Předpokládejme dále, že každé z letadel má pevně stanovené časové okno rezervované technické údržbě stroje.

Takto zadanou úlohu je následně potřeba transformovat do podoby grafu. Vrcholy grafu představují časová okna (letů či technické údržby). Barvy, kterými jsou vrcholy zabarvovány, reprezentují jednotlivá letadla. Hrany grafu pak určují vzájemné překrývání jednotlivých časových oken. Pokud se tedy 2 časová okna vzájemně překrývají, jsou příslušné vrcholy navzájem sousedící. Graf je na začátku řešení úlohy částečně zabarven. Vrcholy, reprezentující časová okna určená pro technickou údržbu letadel, zabarvujeme dle příslušnosti k daným letadlům. Úloha následně řeší zabarvení všech dosud nezabarvených vrcholů jako úlohu klasického barvení grafů. V úloze je však zapotřebí předem počítat s maximálním možným počtem barev, které mohou být k zabarvení grafu použity. Výchozí počet barev se bude shodovat se stávajícím počtem dostupných letadel. Důvodem jsou časová okna údržby letadel, kterým musí být přiřazena barva ještě před

začátkem procesu zabarvování. Pokud by tento počet barev k zabarvení všech vrcholů grafu nestačil, je nezbytné navýšit počet barev o jednu a celý proces zabarvení zopakovat. Před samotným zopakováním zabarvování grafu je však nutné celý graf upravit. Do grafu je zapotřebí přidat vrchol odpovídající časovému oknu technické údržby nového stroje, a dle časového umístění okna vytvořit nové hrany představující jeho překrývání se stávajícími časovými okny grafu. Tento proces je pak zapotřebí opakovat až do nalezení přípustného řešení.

Úloha se dá zjednodušit odebráním časových oken určených k údržbě letadel. Bez nich pak vrcholy představují pouze časová okna letů. Význam hran a barev zůstává nezměněn. Proces zabarvování pak není nutno opakovat a výsledkem úlohy je zabarvení grafu, tedy minimální počet letadel nezbytných k pokrytí všech letů. Dalším výstupem úlohy je pak přiřazení letadel k jednotlivým letům.

5.2 Přerozdělování přístupu uživatelů k souborům ve sdílené síti

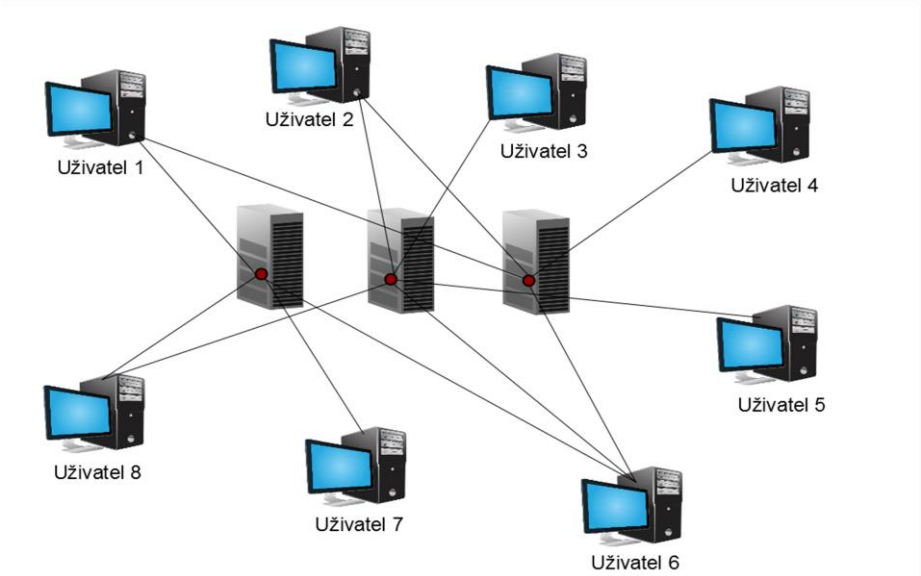
Praktickým příkladem použití MDCP je řešení problematiky simultánního přístupu k souboru ve sdílené počítačové síti. V této úloze se určitý počet uživatelů v jeden časový okamžik snaží dostat k souboru ve sdílené počítačové síti. Má-li však na server, na kterém je soubor uložen, přístup současně více než dva uživatelé, je negativně ovlivněna operační rychlost. Snížení operační rychlost může být redukováno omezením počtu uživatelů, kteří by měli ve stejný časový okamžik na server přístup. Každý uživatel by měl přístup na server v krátkém časovém okně, které by se po prostřídání oken dalších uživatelů opakovalo. Pokud bude toto časové okno mít řád milisekund, může se snížení operační rychlosti serveru stát téměř nepostřehnutelným. Pokud navíc víme, že současný přístup maximálně dvou uživatelů má na operační rychlost pouze zanedbatelný vliv, můžeme sdílet časová okna pro více uživatelů, čímž snížíme nezbytný počet těchto časových intervalů. Pokud by se nám povedlo rozřadit všechny uživatele do co nejmenšího počtu skupin, ve kterých je možný současný přístup k serveru bez významného snížení rychlosti, pak může být vytvořen přístup všech uživatelů k serveru na bázi velmi krátkých, periodicky se opakujících časových intervalů. Při velmi krátkých časech by pak žádný z uživatelů nemusel čekat příliš dlouhou dobu a přístup by se přes svoji diskretizaci mohl jevit jako téměř simultánní.

Tento přístup časových diskretizovaných oken pro skupiny uživatelů je znám jako *Time Division Multiple Access* (TDMA), a je běžně používán v mobilních sítích, pevných telefonních linkách a lokálních počítačových sítích. Uživatelé mohou v sítích s tímto přístupem například sdílet stejnou frekvenci, ve které vysílají v krátkých přidělených časových úsecích. Díky tomu mohou sdílet méně přenosových médií, než kdyby každý využíval vlastní frekvenci.

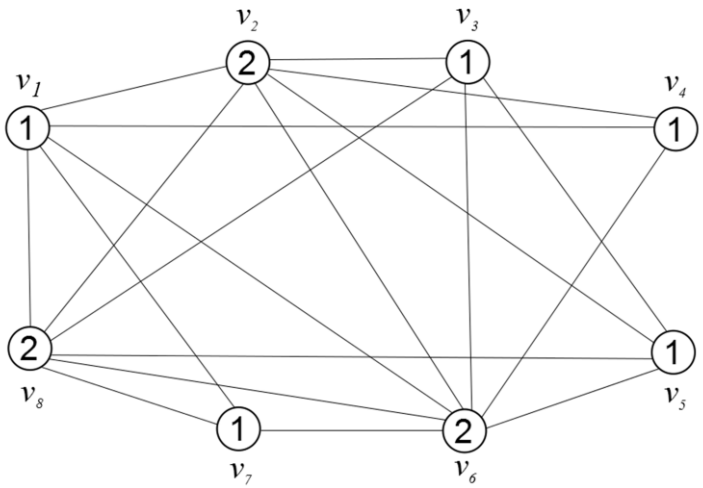
Problematiku přerozdělování sdílených zdrojů, jejíž příklad je uveden na předchozích řádcích, je možné řešit pomocí MDCP. Použití této metody je popsáno na následujícím příkladu.

Uvažujme počítačovou síť, ve které je 8 různých uživatelů. Tito uživatelé požadují přístup k některým ze tří serverů, jak je znázorněno na schématu Obrázku 4. Toto schéma lze znázornit pomocí grafu v takové podobě, aby bylo možno problém řešit jako MDCP. Graf G na Obrázku 5 obsahuje celkem 8 vrcholů, z nichž každý reprezentuje právě jednoho z uživatelů počítačové sítě. Jednotlivé servery však vrcholy reprezentovány nejsou. Mezi 2 vrcholy existuje hrana právě tehdy, pokud oba uživatelé, kteří jsou vrcholy reprezentováni, požadují přístup minimálně k jednomu sdílenému serveru. Z Obrázku 4 můžeme pozorovat, že uživatel 1 a uživatel 8 požadují přístup ke stejnému serveru. Vrcholy v_1 a v_8 jsou tedy v grafu navzájem sousedící.

Dále předpokládejme, že pokud bude mít k jednomu serveru přístup současně více jak jeden uživatel, bude způsobeno zpomalení celé sítě. Toto zpomalení je pro uživatele akceptovatelné v případě, že k jednomu serveru budou mít přístup maximálně 3 uživatelé. Pokud by k jednomu sdílenému serveru mělo současně přístup 4 a více uživatelů, zpomalení by bylo neakceptovatelné a cíl úlohy by nebyl splněn.



Obrázek 4 Graf reprezentujícíc počítačovou síť (Zdroj: autor)



Obrázek 5 – Schéma počítačové sítě (Zdroj: autor)

Efektivní rozvržení přístupu jednotlivých uživatelů ke všem serverům může být řešeno jako úloha zabarvení grafu G nejmenším možným počtem barev tak, že žádný vrchol zabarvený libovolnou barvou nebude sousedit s více než $b = 2$ vrcholy, které jsou obarveny toutéž barvou. Barva v této úloze reprezentuje časový interval pro přístup k serveru. Cílem úlohy je minimalizovat počet časových intervalů a tím zrychlit střídání přístupů jednotlivých skupin uživatelů. Podmínku lze také formulovat tak, že žádný podgraf indukovaný jednou z barevných tříd nebude obsahovat vrchol stupně vyššího než $b = 2$. Takto formulovaný problém můžeme řešit některou z metod defektivního barvení grafů. Přípustné zabarvení grafu je zobrazeno na Obrázku 5. K zabarvení grafu G jsou dle barvícího pravidla potřeba minimálně 2 barvy. Jelikož graf G obsahuje jako podgraf kliku čtvrtého řádu, je zřejmé, že zabarvení musí být optimální. Dle tohoto zabarvení lze tedy vyřešit problém současného přístupu uživatelů k serverům rozdělením těchto uživatelů do 2 skupin, kdy uživatelé 1, 3, 4, 5 a 7 budou mít přístup k serverům v jeden časový interval, kdežto uživatelé 2 a 6 budou sdílet druhý časový interval.

5.3 Rekonstrukce sítě metra

Jiným příkladem použití MDCP je úloha zmíněná v práci *On a graph colouring problem arising form discrete tomography* [19]. Úloha je zaměřena na rekonstrukci stanic sítě metra. Předpokládejme, že rekonstrukce jedné stanice trvá právě 1 měsíc. Z důvodu omezení vlivu rekonstrukce stanic na provoz metra je požadováno, aby na žádné lince metra nebylo ve stejný měsíc opravováno více než b stanic. Kvůli úspoře nákladů a minimalizaci vlivu rekonstrukce na omezení dopravy ve městě je cílem provést opravy stanic v co nejkratším čase. V práci Bentz [19] je úloha popsána následovně:

Síť metra je reprezentována grafem $G = (V, E)$, kde vrcholy představují jednotlivé stanice. Každá z linek metra vytváří subgraf grafu G typu řetěz. Každý takový řetěz označíme P_i. Nechť je h_i^j počet stanic z linky metra P_i, které budou opravovány ve stejný měsíc j. Formulace problému je následující: Hledáme rozdělení vrcholů z množiny V do podmnožin $V^1, V^2, \dots, V^k$ tak, že platí

$$\left| P_i \bigcap V^j \right| = h_i^j \tag{5.1}$$

Tedy, že pro každou linku metra bude maximální počet stanic opravovaných ve stejný měsíc roven h_i^j. Toto číslo by mělo reprezentovat počet stanic, jejichž simultánní rekonstrukce nebude narušovat dopravu ve městě více, než je povoleno.

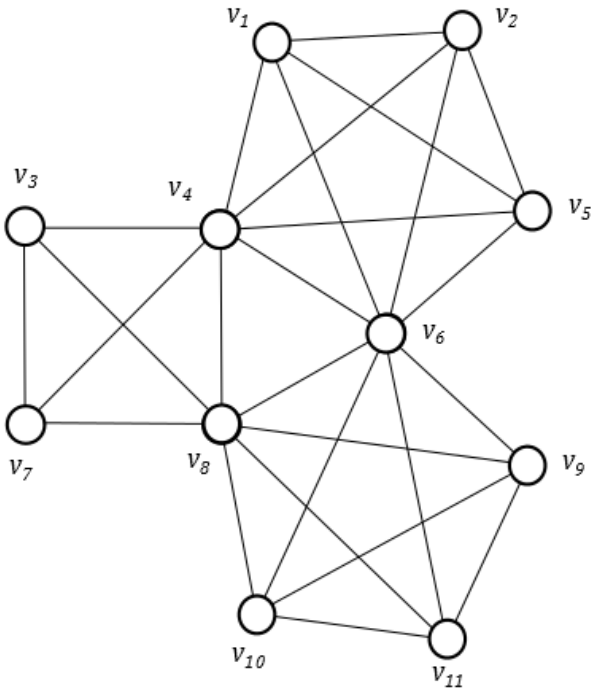
Tento problém může být řešen i metodou MDCP. K tomu je však zapotřebí odlišná formulace problému. Aby bylo možno tento přístup použít, je zapotřebí vytvořit nový graf. Vrcholy nového grafu stále reprezentují jednotlivé stanice metra. Barvy, jež použijeme k zabarvení vrcholů, představují měsíce, v nichž budou stanice rekonstruovány. Změna oproti výše popsanému řešení ale nastává ve významu hran grafu. Hrany v předchozí formulaci z práce [19] představují fyzické spojení mezi jednotlivými stanicemi. Pro řešení pomocí MDCP je však nutné, aby hrany reprezentovaly příslušnost stanic k linkám metra. Každá stanice tedy v grafu sousedí se všemi stanicemi téže linky. Graf reprezentující síť metra je tvořen několika kompletními subgrafy. Počet kompletních subgrafů pak odpovídá počtu linek v síti metra. Jednotlivé subgrafy jsou pak vzájemně propojeny ve vrcholech představujících přestupní stanice. Cílem úlohy je minimalizovat počet měsíců, během kterých bude možné rekonstruovat všechny stanice sítě metra.

Zadání úlohy dále stanovuje maximální počet stanic náležících jedné lince, které mohou být opravovány ve stejný měsíc. V prvním přístupu řešení byl tento počet označen jako h_i^j. V případě řešení úlohy způsobem MDCP je tento počet stanic označen jako b. Hodnota b tedy stanovuje maximální počet vrcholů sousedících s libovolným vrcholem $v \in V$, které mohou být zabarveny identickou barvou jako vrchol v. Hledáme tedy takové zabarvení, ve kterém žádná z barevných tříd neobsahuje vrchol řádu vyššího než b. Celkový počet barevných tříd pak udává minimální počet měsíců, potřebný k rekonstrukci celé sítě metra.

Praktický příklad použití je pro rekonstrukci stanic metra v Praze. Celkem 11 stanic všech pražských linek metra bylo vybráno k celkové rekonstrukci. Rekonstrukce se týká 4 či 5 stanic každé ze tří linek, a to včetně všech 3 přestupních stanic. Zjednodušené schéma rekonstruovaných stanic lze nalézt na Obrázku 6.

Obrázek 6 - Schéma linek metra v Praze (Zdroj: [35])

Stanice v centru Prahy nyní převedeme do podoby grafu, ve kterém, jak již bylo popsáno, existuje hrana pouze mezi vrcholy, které představují stanice stejné linky. Na Obrázku 6 je vidět, že graf linek metra je tvořen 3 vzájemně propojenými klikami. Každá ze 3 klik reprezentuje jednu z linek metra. Vrcholy v_4, v_6 a v_8, ve kterých dochází k propojení jednotlivých klik, představují 3 přestupní stanice Můstek, Florenc a Muzeum. V Tabulce 1 jsou přiřazeny jednotlivým vrcholům odpovídající stanice metra



Obrázek 7 Graf stanic metra (Zdroj: autor)

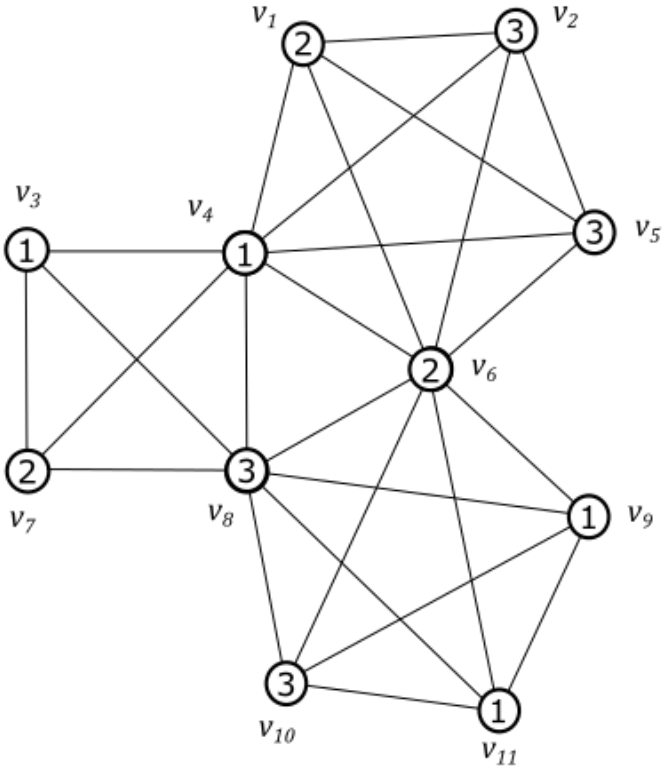
Tabulka 1 Přiřazení stanic k vrcholům grafu (zdroj: autor)

Vrchol	Stanice	Linka
v_1	Národní třída	B
v_2	Náměstí republiky	B
v_3	Staroměstská	A
v_4	Můstek	A/B
v_5	Křižíkova	B
v_6	Florenc	B/C
v_7	Náměstí míru	A
v_8	Muzeum	A/C
v_9	Vltavská	C
v_{10}	I.P. Pavlova	C
v_{11}	Hlavní nádraží	C

V rámci zadání úlohy je dále požadováno, aby na žádné z linek nebyla současně prováděna rekonstrukce více než 2 stanic. V případě řešení metodou MDCP to znamená, že konstanta $b = 1$.

Zadání úlohy pro řešení matematickým modelem MDCP je v podobě dvou vstupů. Prvním vstupem je hodnota b, druhým pak incidenční matice grafu. Výstupem modelu je zabarvený graf. Řešení úlohy je zobrazeno na Obrázku 8. Číselné hodnoty vepsané v jednotlivých vrcholech představují 3 barvy, kterými lze graf zabarvit za současného splnění všech omezujících podmínek úlohy.

Graf je zabarven právě 3 barvami, což znamená, že k rekonstrukci stanic bude zapotřebí 3 měsíců. V prvním měsíci budou rekonstruovány stanice Staroměstská, Můstek, Hlavní nádraží a Vltavská. V druhém měsíci proběhne rekonstrukce stanic Národní třída, Náměstí Míru a Florenc. Třetí měsíc pak budou rekonstruovány zbývající stanice Náměstí republiky, Křižíkova, Muzeum a I.P. Pavlova.



Obrázek 8 Výsledné zabarvení grafu metodou MDCP (Zdroj: autor)

Zabarvení grafu bylo dosaženo exaktní metodou celočíselného matematického modelu představeného v této práci, tudíž je optimální. Je známo, že při omezení $b = 1$ nemůže být žádná klika pátého řádu zabarvena méně než 3 barvami. Graf představující síť stanic metra obsahuje dvě kliky pátého řádu.

5.4 KOMUNIKACE SATELIT-POZEMNÍ PŘIJÍMAČ

Posledním příklad použití MDCP je pro případ řešení problematiky komunikace pozemních terminálů se satelitem představený v práci *Quasi-optimal bandwidth allocation for multi-spot MFTDMA satellites* [20]. Představme si rozsáhlý satelitní systém tvořený satelitem a množstvím pozemních přijímačů, kterým řídící jednotka přiřazuje jednotlivé frekvence rozdělené do časových slotů. Tento způsob komunikace je obdobný, jako v příkladu 6.1. Zde je způsob přerozdělování poněkud odlišný. Nazývá se MFTDMA (Multi-Frequency Time-Division Multiple Access), neboli mnohonásobný multifrekvenční přístup s časovým rozdělením. Zaměřit konkrétním signálem vysílaným satelitem pouze jeden specifický přijímač je technicky velmi náročné. Navíc jsou takovéto vysílače finančně velmi nákladné. Proto se většinou užívají vysílače s vyšším rozptylem, které jsou však z hlediska pořizovacích nákladů mnohem výhodnější. Nevýhodou použití těchto

vysílačů však je, že v případě koncentrace více přijímačů na poměrně malé ploše bude signál určený pro konkrétní přijímač obdržen také v přijímačích přilehlých. Pro všechny ostatní přijímače však tento signál bude nežádoucím šumem, a bude zapotřebí ho odfiltrovat. Pokud je však součet rušivých signálů příliš velký, přijímač již dále není schopný rozlišit signál, který bude určen přímo pro něj. V práci [20] se zkoumá způsob řešení, kterak přiřadit co nejmenší počet frekvencí vysílaných satelitem jednotlivým pozemním přijímačům tak, aby každý přijímač byl schopen rozlišit přijímaný signál od šumu způsobeného rozptylem signálu.

Pro potřeby řešení pomocí *MDCP* bychom úlohu mohli interpretovat následovně. Vrcholy grafu $G(V, E)$ by představovaly jednotlivé pozemní přijímače, které by byly ve spojení s jedním satelitem. Vzájemné rušení by pak bylo předpokládáno jako symetrické. To znamená, že pokud by přijímač v_1 zachytával šum způsobený vysíláním signálu pro přijímač v_2, pak by i opačně přijímač v_2 zachytával šum způsobený signálem pro v_1. Každé 2 vrcholy by pak byly přilehlé v případě, že odpovídající přijímače by se nacházely v takové blízkosti, která by způsobovala vzájemný šum v důsledku rozptylu vysílaného signálu stejné frekvence. Z důvodu rozlišení signálu od šumu by pak musela platit podmínka, že pozemní přijímač p nesmí být v okruhu vysílání pro více než d dalších přijímačů, které přijímají signál stejné frekvence. Cílem úlohy by pak bylo stanovit nejnižší možný počet barev nutných k zabarvení grafu a reprezentujících vysílané frekvence tak, aby byla splněna předchozí podmínka a každý vrchol byl sousední maximálně s d vrcholy zabarvenými stejnou barvou.

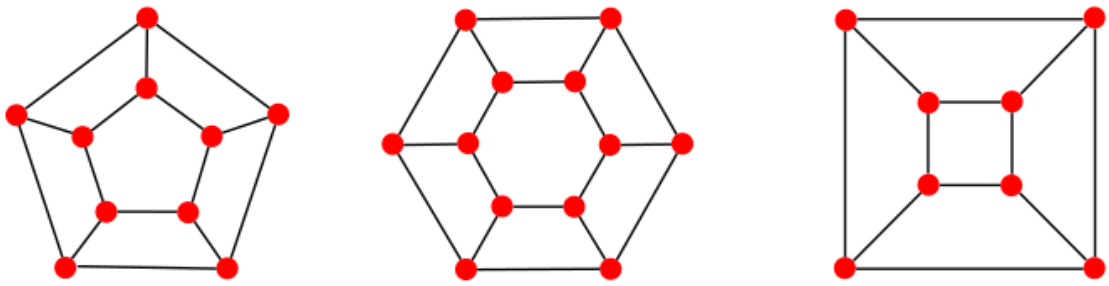
V [20] byla úloha řešena poněkud jiným přístupem. Úloha zde byla formulována jako *fractional colouring problem* a řešena pomocí lineárního a celočíselného programování. Je zde také popsán příklad s 8 přijímači, pro který bylo získáno téměř optimální řešení a dále pokus se souborem 32 přijímačů.

6 TESTOVANÉ GRAFY

Vzhledem k záměru porovnání efektivity algoritmů prezentovaných v této práci a algoritmů představených v práci Isabelle Niewoudt [9] byly k testování vybrány podobné skupiny grafů. Testováno bylo celkem 31 grafů s různým počtem vrcholů, hran a hustotou hran. První skupiny grafů byly vybrány z knihy Atlas of graphs [21]. Pojmenování těchto skupin grafů zůstalo zachováno stejné jako v [21].

Kubické grafy

Grafy označené C_i představují takzvané kubické, nebo také trivalentní grafy řádu i. Graf G je nazýván kubickým právě tehdy, když každý z jeho vrcholů v má stupeň vrcholu roven $st(v) = 3$. Kubické grafy pak mají pouze sudý počet vrcholů [22]. V závislosti na stupni vrcholu $st(v) = 3$ pak vyplývá, že nejmenším kubickým grafem je kompletní graf K_4.

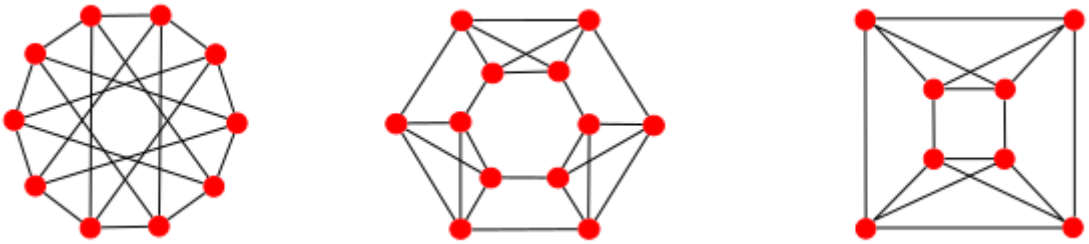


Obrázek 9 - Kubické grafy řádu 8,10 a12 (Zdroj: autor)

Pro testování algoritmů byly použity kubické grafy řádů $i = \{8,10,12\}$.

Kvartické grafy

Grafy označené Q_i představují takzvané kvartické grafy řádu i. Graf G je nazýván kvartickým právě tehdy, když každý z jeho vrcholů v má stupeň vrcholu roven $st(v) = 4$. Pro testování algoritmů byly použity kubické grafy řádů $i = \{8,10,12\}$. Graf K_5 je jediný kompletní graf, který je zároveň kvartickým grafem. [23]



Obrázek 10 - Kvartické grafy řádu 8,10 a 12 (Zdroj: autor)

Kružnicové grafy

Testované grafy z Atlas of graphs [21] se pro zkoušení algoritmů ukázaly jako příliš jednoduché. Proto bylo upuštěno od testování dalších grafů z této práce. Místo toho bylo vytvořeno pět grafů typu "*kružnice*". Tyto grafy jsou unikátní tím, že tvoří takzvané Hamiltonovské kružnice. Takový graf definujeme jako souvislý pravidelný graf druhého stupně. [1] Graf je tedy stejně jako předchozí dvě skupiny grafů souvislý a neobsahuje žádné visící vrcholy. K testování byly vybrány grafy jak se sudým, tak s lichým počtem

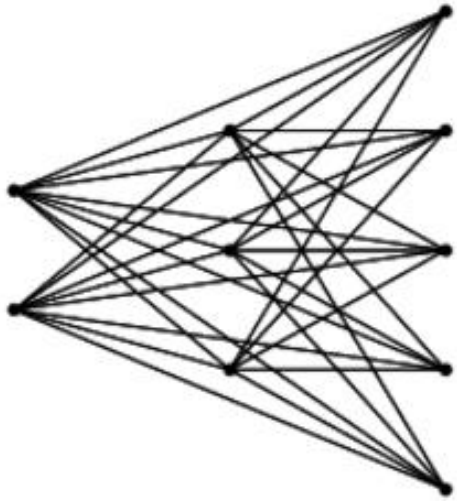
Obrázek 11 - Příklady kružnicových grafů (Zdroj: [37])

vrcholů. V této práci byly testovány grafy řádů $i = \{19,28,29,49,50\}$.

Kompletní multipartitní grafy

Další testovanou skupinou již komplexnějších grafů byly kompletní multipartitní grafy. Vrcholová množina V kompletního multipartitního grafu se skládá z několika disjunktních vrcholových podmnožin $V_i, i = 1,2, \dots, n$ a hranová množina E pak obsahuje všechny hrany, jejichž koncové vrcholy přísluší do různých disjunktních podmnožin V_i a V_j , kde $i \neq j$. Dále platí, že žádné dva vrcholy množiny V_i nejsou navzájem spojeny

hranou. Grafy značíme $K(n \times m)$, kde K značíme graf obsahující n partitních skupin



Obrázek 12 - Kompletní multipartitní graf (zdroj: [40])

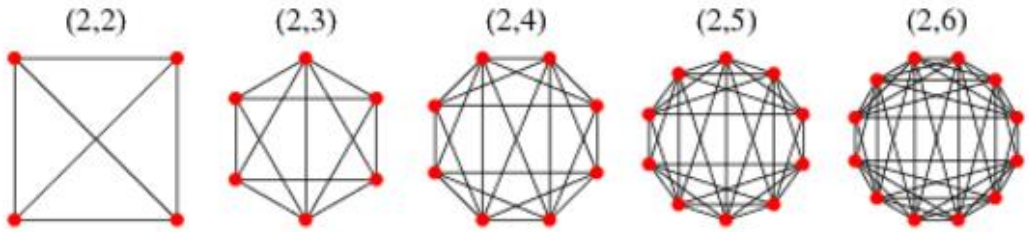
vrcholů, z nichž každá obsahuje m vrcholů.

Benchmark grafy

Následující skupiny grafů byly získány z „The second implemetation DIMACS challenge" [24]. Tato publikace obsahuje množství neorientovaných grafů a zabývá se právě barvením grafů a hledáním maximální kliky v grafu. Jedenáct grafů bylo vybráno z DIMACS za účelem testování algoritmů na komplexních grafech. Tyto grafy pochází z následujících skupin grafů:

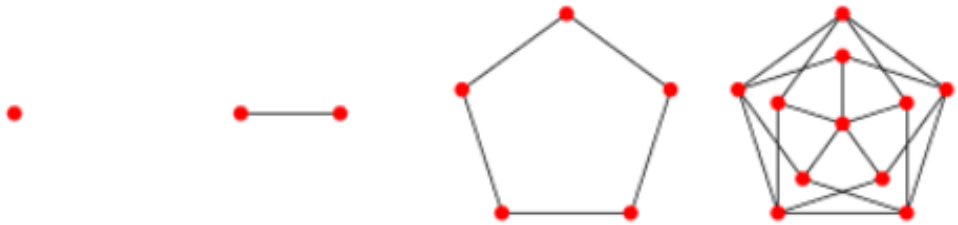
- **Knižní grafy -** jsou zajímavou skupinou grafů vycházející ze světové literatury. Každý vrchol zde představuje knižní postavu. Dva vrcholy jsou poté spojeny hranou, pokud se dané postavy v knize setkaly. Knižní grafy použité v této práci vychází ze tří děl světové literatury: David Copperfield od Charlese Dickense ($david$), Huckleberry Finn od Marka Twaina ($huck$) a Les Mis'erables od Victora Huga ($jean$). Každý z těchto grafů obsahuje více než 70 vrcholů.
- **Queen grafy –** představují grafy vycházející z šachovnice a možnostmi pohybu figurky královny. Je-li dána $n \times n$ šachovnice, queen graph je grafem s n^2 vrcholy, kde každý z vrcholů představuje jedno hrací pole šachovnice. Dva vrcholy jsou přilehlé právě tehdy, pokud hrací pole, které představují, leží ve stejném řádku, sloupci či diagonále. Odtud pochází pojmenování této skupiny grafů, jelikož vertikální, horizontální či diagonální směry reprezentují možnosti pohybu

figurkou královny. V této práci jsou použity tři různě velké grafy z této skupiny. Po vzoru pojmenování z DIMACS jsou grafy označeny $queen(n\_n)$, kde n značí počet hracích polí v šachovnicovém sloupci či řádku. [25]



Obrázek 13 - Queen grafy (Zdroj: [25])

- **Mycielski graf –** jsou speciálním druhem grafů, sestrojených Janem Mycielski. Tyto grafy jsou zkonstruovány tak, že neobsahují žádné kliky s řádem $i = 3$. Pojmenování těchto grafů odpovídá názvosloví dle DIMACS.



Obrázek 14 - Mycielski grafy (Zdroj: [38])

Poslední testovanou skupinou grafů byly grafy náhodné. Matice sousednosti náhodných grafů byly generovány pomocí matematického softwaru Mathematica. Každá skupina náhodných grafů obsahovala celkem 5 grafů s téměř totožnou hustotou hran p a pevně daným počtem vrcholů. Celkem bylo vytvořeno 15 skupin náhodných grafů kombinací tří různých hustot grafů $p = \{0{,}3; 0{,}5; 0{,}7\}$ a pěti různých počtů vrcholů $n = \{25; 40; 55; 70; 85\}$. Celkem tedy bylo generováno a testováno 75 grafů. Každý náhodný graf byl označen $G_{n,p}$, tedy jako graf s n vrcholy, přičemž každá možná hrana grafu je v něm obsažena s pravděpodobností p.

7 METODOLOGIE

Veškeré výpočty byly provedeny na stolním počítači Dell 9020 s procesorem Intel® Core™ i7-4790 CPU @ 3.6 GHz, operační pamětí RAM 8.00 GB a operačním systémem 64-bit Win7. Matematický model byl vytvořen a řešen v softwaru IBM ILOG CPLEX Optimization Studio verze 12.5. První pokusy o řešení modelu byly provedeny v softwaru LINGO, nicméně neúspěšně. Matematický model nebylo možné v programu LINGO řešit pro žádné grafy s počtem vrcholů větším než $n > 20$. Bylo zjištěno, že problém je s největší pravděpodobností softwarový, jelikož identický matematický model se stejným zápisem v novější verzi LINGO fungoval. Tato verze bohužel nebyla k účelům diplomové práce dostupná, tudíž byl použit pouze ILOG CPLEX.

Model samotný byl navrhnut k výpočtu $\chi_b^{\Delta}(G)$ daného grafu G za stanovený časový interval pro stanovenou hodnotu b. K získání chromatické sekvence bylo potřeba model doplnit o skript takzvaný Flow control. Skript lze zevrubně popsat následovně:

Po obdržení řešení zadané úlohy se hodnota b zvýší na $b + 1$ a celý model je spuštěn znovu. Tento postup je opakován do doby, kdy chromatické číslo jakožto výsledek matematického modelu, je rovno $\chi_b^{\Delta}(G) = 1$. Pro každé dílčí řešení byl zaznamenám počet iterací a čas výpočtu.

Výsledky výpočtů jsou srovnány s výsledky algoritmů z dizertační práce Isabelle Niewoudt [9]. Pouhé srovnání však není dostačující, jelikož tyto výsledky byly získány na jiném zařízení, konkrétně na laptopu Intel Core 2 Duo T7200 CPU @ 2.0 GHz s operačním systémem Windows XP Professional SP2. Rozdíl mezi těmito procesory je významný ve více faktorech, a je obtížné výsledky objektivně srovnávat. Prvním zkoušeným postupem bylo získání co největšího počtu informací o obou procesorech a srovnání různých dostupných benchmark testů. Zohledněním výsledků benchmark testů a informací ohledně fungování a parametrů procesorů pak byl stanoven koeficient, dle kterého byly upraveny časy výpočtů. Takové srovnání je pouze informativního charakteru, protože nemůže být dostatečně podloženo. Nakonec se podařilo kontaktovat paní doktorku Niewoudt, která poskytla zdrojové kódy svých algoritmů a stejně tak i použitá data. Její algoritmy tak mohly být spuštěny na stejném zařízení jako modely této práce a testovány v novější verzi softwaru Mathematica.

8 TESTOVÁNÍ MATEMATICKÉHO MODELU

Model popsaný v kapitole 4 byl testován pro všechny uvedené benchmark grafy a náhodně generované grafy. Každý graf byl testován jak exaktním způsobem, tak použitím modelu s časovým omezením. Matematický celočíselný model zůstal pro oba případy stejný. Jediným rozdílem v těchto dvou přístupech je časový limit výpočtu jednotlivých χ_b^{Δ}. Pro oba případy je použit jako řešitel *IBM ILOGs off –the-shelf solver suite*. Pro každý graf pak je zjišťována kompletní Δ-chromatická sekvence. Řešící algoritmus postupuje následovně: pro každý graf je zjišťován nejprve první člen Δ-chromatická sekvence, tedy případ χ_b^{Δ} kde $b = 0$. Pokud je tento člen nalezen, algoritmus zvýšuje hodnotu koeficientu na $b + 1$ a spustil znovu celý výpočet. Tento postup se opakoval až do okamžiku, kdy počet barev potřebný k zabarvení celého grafu klesl na hodnotu rovné $\sum_{c=1}^{n} y_c = 1$. Poté byl celý algoritmus ukončen.

V exaktním přístupu je pro výpočet Δ-chromatické sekvence určen časový limit (dále jen ***TO***) 12 hodin. Pokud v tomto časovém intervalu nebyla sekvence spočtena, zaznamenaly se pouze ty členy sekvence, pro které byla nalezena optimální hodnota. Za posledním členem s vypočtenou optimální hodnotou pak bylo uvedeno překročení časového limitu v podobě (TO) a další členy sekvence již nebyly počítány.

Pro řešení grafu pomocí modelu s časovým omezením pak byly stanoveny časové limity výpočtů jednotlivých členů Δ-chromatické sekvence $\chi_b^{\Delta}(G)$. Po uplynutí časového limitu byl zaznamenán nejnižší vypočtený počet barev potřebných k zabarvení grafu pro stanovenou hodnotu b, poté se tato hodnota zvýšila na $b + 1$ a celý proces byl zopakován. Algoritmus byl ukončen v případě, kdy $\sum_{c=1}^{n} y_c = 1$. V zájmu srovnání efektivity byly pro model s časovým omezením zvoleny tři různé časové limity.

V následujících podkapitolách jsou popsány výsledky použití modelu s časovým omezením, exaktní metody řešení a následně použití těchto metod pro řešení skupiny náhodně generovaných grafů. Skupina náhodně generovaných grafů se totiž od ostatních ve složitosti grafů a výsledcích významně liší.

8.1 ŘEŠENÍ PROBLÉMU MDCP POUŽITÍM MODELU S ČASOVÝM OMEZENÍM

Heuristický přístup pro získání *Δ-chromatické sekvence* grafu byl vytvořen pomocí stanovení časového limitu pro výpočet každého $\chi_b^{\Delta}(G)$ v sekvenci. Tímto způsobem byly vytvořeny tři přístupy, které se liší pouze časovým limitem, po kterém jsou také pojmenovány: *Timelim-1s*, *Timelim-5s* a *Timelim-15s*. Použitý matematický model je pro všechny 3 přístupy stejný. Všechny výsledky těchto algoritmů jsou uvedeny v Tabulce 2. V té je z důvodu přehlednosti uvedeno pouze prvních 10 členů každé sekvence. V tabulce jsou kromě sekvencí, vypočtených pro dané grafy, uvedeny také časy výpočtů $\Delta-$ *chromatických sekvencí*, řády grafů a velikost grafů. Celé sekvence jsou k nalezení v příloze.

Heuristická metoda nebyla použita pro celkem patnáct testovaných grafů z Atlas of Graphs [21]. Důvodem je jejich malý řád, velikost a s tím spojený nízký výpočetní čas, který pro žádné $\chi_b^{\Delta}(G)$ nepřekročil nejmenší stanovený časový limit 1s. Výpočetní čas celých sekvencí pak u žádného grafu nepřekročil 2s. Naopak, několik celých sekvencí grafů bylo spočteno v čase nižším než 1s. Výsledky těchto grafů jsou tudíž zmíněny pouze v kapitole 8.2.

Složitost řešení benchmark grafů a kompletních multipartitních grafů je srovnatelná, proto budou výsledky těchto dvou skupin grafů popisovány současně. Dva benchmark a dva kompletní multipartitní grafy (*myciel4*, *myciel5*, *K4x7*, *K4x11*) byly taktéž spočteny pouze exaktní metodou z důvodu krátkého času výpočtu stejně jako v případě grafů z [21]. Jeden z benchmark grafů (*queen5_5*) pak byl ze stejného důvodu řešen pouze pomocí *Timelim-1s*. Časy výpočtů χ_b^{Δ} v tomto grafu byly pro všechna *b* kratší než 5s. Je evidentní, že výpočetní čas *Δ-chromatické sekvence* závisí na stanoveném časovém limitu pro výpočet jednotlivých $\chi_b^{\Delta}(G)$, a tudíž se mění v závislosti na délce sekvence.

Z výsledků lze pozorovat, že velikost grafu a s ní spojená hustota hran grafu mají významný vliv na čas výpočtu a počet iterací. Výpočetní čas grafu pomocí modelu s časovým omezením závisí také na řádu grafu. Především pro *Timelim-1s* je řád grafu významným faktorem. V případě *Timelim-15s* se pak zvyšuje vliv hustoty hran grafu.

Tabulka 2 Výsledky modelů s časovým omezením (Zdroj: autor)

Graf	Časový limit [s]	Řád grafu	Velikost grafu	Čas výpočtu [s]	Sekvence
queen5_5	1	25	160	8,7	5 5 4 3 3 3 2 2 2 2
queen6_6	1	36	290	13,3	8 6 5 4 4 3 3 2 2 2
	5			27,18	7 6 5 4 3 3 3 2 2 2
	15			56,68	7 6 5 4 3 3 3 2 2 2
queen7_7	1	49	476	18,44	9 8 6 5 4 4 3 3 3 2
	5			45,65	8 7 6 5 4 4 3 3 3 2
	15			97,12	7 7 6 5 4 4 3 3 3 2
queen8_8	1	64	728	28,82	12 9 8 6 5 4 4 4 3
	5			67,09	11 9 7 6 5 4 4 3 3
	15			122,6	11 9 7 6 5 4 4 3 3
myciel6	1	47	236	12,64	6 4 3 3 3 2 2 2 2 2
	5			18,3	6 4 3 3 3 2 2 2 2 2
	15			28,36	6 4 3 3 3 2 2 2 2 2
myciel7	1	95	755	63,23	7 5 4 4 4 4 3 3 3 3
	5			100,42	7 5 4 4 3 3 3 3 3 2
	15			140,46	7 5 4 4 3 3 3 3 3 2
K7x4	1	28	336	14,11	7 7 7 6 4 4 4 3 3 3
	5			23,58	7 7 7 6 4 4 4 3 3 3
	15			43,27	7 7 7 6 4 4 4 3 3 3
K11x4	1	44	880	34,39	11 11 11 10 6 6 6
	5			62,42	11 11 11 9 6 6 6 5
	15			136,74	11 11 11 9 6 6 6 5
K7x11	1	77	2541	89,73	7 7 7 7 7 7 7 7 7 7
	5			220,54	7 7 7 7 7 7 7 7 7 7
	15			354,45	7 7 7 7 7 7 7 7 6 6
david	1	87	406	89,31	11 7 5 5 4 4 3 3 3 3
	5			110,05	11 6 5 4 4 3 3 3 3 3
	15			148,7	11 6 5 4 4 3 3 3 3 3
jean	1	80	254	35,11	10 6 4 3 3 3 2 2 2 2
	5			46,71	10 5 4 3 3 3 2 2 2 2
	15			65,70	10 5 4 3 3 3 2 2 2 2
huck	1	74	301	46,87	11 6 5 4 3 3 3 3 2 2
	5			57,76	11 6 4 4 3 3 3 3 2 2
	15			80,42	11 6 4 4 3 3 3 3 2 2

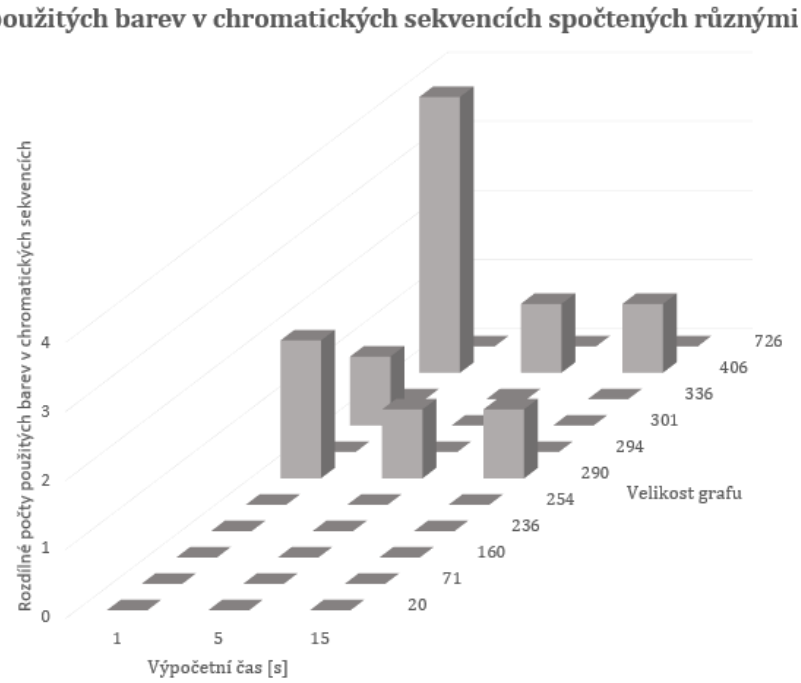
Druhý a třetí člen *Δ-chromatické sekvence* ($\chi_1^{\Delta}(G), \chi_2^{\Delta}(G)$) v *Timelim-1s* a *Timelim-5s* mají obecně nižší počet řešených iterací, než je tomu u ostatních členů sekvence, první člen $\chi_0^{\Delta}(G)$ nevyjímaje. *Δ-chromatické sekvence* testovaných grafů vypočtené modely *Timelim-5s* a *Timelim- 15s* se odlišují pouze ve dvou případech, konkrétně u grafů *K7x11* a *queen7_7*. Zde se dostáváme k jednomu z významných faktorů modelu s časovým omezením a to přesnosti. Z důvodu celkové komplexnosti grafů nebylo pro 5 z celkového počtu 16 grafů získáno exaktní řešení. Z tohoto důvodu není možnost ověřit, zda některý z modelů s časovým omezením dosáhl u těchto grafů optimálního řešení. Grafy, pro které nebylo nalezeno optimální řešení použitím exaktního algoritmu, jsou zobrazeny v Tabulce 3.

Tabulka 3 Přehled grafů bez optimálního řešení (Zdroj: autor)

Graf	Řád graf	Velikost grafu
queen7_7	49	476
queen8_8	64	728
myciel7	95	755
K7x11	44	880
K11x4	77	2541

Pro celkem 11 grafů bylo nalezeno optimální řešení a lze u nich srovnávat výsledky modelu s různým časovým omezením. Pro 9 grafů bylo při použití modelu s různým časovým omezením získáno řešení totožné s řešením. Jediné dva grafy, pro které *Δ- chromatické sekvence* vypočtené modelem s časovým omezením nejsou optimální, jsou grafy *david* a *queen6_6*. V případě *queen6_6* se získaná *Δ-chromatická sekvence* při výpočtu *Timelim-15s* liší pouze v případě členu χ_2^{Δ}. Zbytek sekvence je identický s optimální sekvencí. Graf *david* se od optimální sekvence liší při výpočtu *Timelim-15s* také pouze jedním členem sekvence, a to χ_4^{Δ}. V obou případech byl nejmenší vypočtený počet barev, nutných k zabarvení grafu *david* či *queen6_6*, odlišný od optimálního počtu pouze o 1 barvu. Obrázek 15 zobrazuje odchylky sekvencí vypočtených matematickým modelem s jednotlivými časovými omezeními pomocí 3D sloupcového grafu. Na jedné ose jsou pomocí velikosti grafu zobrazeny všechny grafy, pro které bylo získáno optimální řešení. Horizontální osa představuje jedno z časových omezení matematického modelu. Tento graf názorně ukazuje součet odchylek členů vypočtených sekvencí od hodnot členů

optimální sekvence každého grafu. Ze sloupcového grafu je možné pozorovat, že odchylky sekvencí jsou pozorovány v případě *Δ-chromatické sekvence* grafu *david*, kde $|E_{david}| = 406$. U tohoto grafu se sekvence vypočtená pomocí *Timelim_1s* liší od optimální *Δ-chromatické sekvence o hodnotu 4*. Přesněji řešeno se tato sekvence liší ve čtyřech členech sekvence $\chi_b^{\Delta}(david)$ $b = (1,2,3,5)$ o hodnotu 1. Lze konstatovat, že v žádné *Δ-chromatické sekvenci* benchmark a kompletních multipartitních grafů se žádný člen χ_b^{Δ} neliší od členu z optimální sekvence o hodnotu vyšší než 1.



Obrázek 15 - Přehled odchylek počtů barev v sekvencích, (Zdroj: autor)

Jedním z očekávaných trendů bylo zvyšování odchylky členů χ_b^{Δ} získaných modely s časovým omezením od optimální hodnoty χ_b^{Δ} v případě zvyšování velikosti grafu či snižování časového limitu. Tento trend byl potvrzen pouze v případě snižování časového limitu. Mnoho *Δ-chromatických sekvencí* spočítaných pomocí *Heuristic-1s* a *Heuristic-15s* se mezi sebou liší v jednom až třech členech sekvence. Větší rozdíly mezi sekvencemi při řešení jednotlivými heuristikami jsou pozorovány spíše u grafů s větším počtem hran. Jak však ukazuje sloupcový graf (Obrázek 15), v případě *Timelim_1s* tento trend neplatí. Lze tedy konstatovat, že existuje další faktor, který výpočet sekvence ovlivňuje.

8.2 EXAKTNÍ METODA PRO ŘEŠENÍ MDCP

Exaktní metodou byly testovány všechny grafy popsané v kapitole 3. Prvními skupinami testovaných grafů byly Q_i, C_i a kružnicové grafy. Všechny grafy z těchto skupin se pro řešení matematickým modelem ukázaly jako triviální, a žádný z výpočetních časů celých *Δ-chromatických sekvencí* nepřekračuje hranici 2 sekund. Výpočetní čas většiny sekvencí grafů ze skupiny C_i se pohybuje okolo 1 sekundy, stejně tomu je i v případě kružnicových grafů. Pouze v případě kvartických grafů ze skupiny Q_i jsou výpočetní časy sekvencí v rozmezí 1 až 2 sekund. Všechny sekvence C_i grafů mají 4 členy, sekvence kružnicí pouze 3 členy a všechny sekvence Q_i grafů vždy 5 členů. V případě těchto jednoduchých grafů nebyl zaznamenán vliv řádu grafu na čas výpočtu či počet iterací. Výpočetní časy kružnicových grafů byly naopak nižší než u ostatních skupin, a to i přes významně vyšší řády grafů.

Jak již bylo zmíněno v předchozí podkapitole, pro 5 z celkově 16 grafů ze skupin benchmark a kompletních multipartitních grafů nebylo získáno optimální řešení. V každém z těchto 5 případů byl překročen časový limit 12 hodin. Přehled grafů bez optimální *Δ-chromatické sekvence* je uveden výše v Tabulce 4.

Z Tabulky 4 je zřejmé, že velikost těchto grafů je ve všech případech větší jak 450 hran. Velikost převážné většiny grafů, pro které byla získána optimální řešení, se pohybovala pod 400 hranami. Toto naznačuje omezení použitelnosti exaktní metody v závislosti na velikosti testovaného grafu. Hraniční velikost, pro kterou je u většiny grafů exaktní metodou získána optimální *Δ-chromatická sekvence*, se pohybuje okolo 400 hran. Pro přesnější stanovení této hranice by bylo nezbytné testování většího počtu grafů velikostí v tomto rozsahu.

Výjimkou z tohoto trendu je graf $K4x11$. Graf $K4x11$ má velikost 726 hran, což je velikost srovnatelná s velikostmi grafů *queen8_8* a *myciel7*, pro které optimální sekvence získána nebyla. *Δ-chromatická sekvence* grafu $K4x11$ přitom byla vypočtena v čase nižším jak 1 minuta. Z tohoto důvodu nelze stanovit hraniční velikost grafů, pro které je tímto exaktním modelem možno určit optimální *Δ-chromatická sekvence*.

Tabulka 4 - Přehled výsledků exatního přístupu (Zdroj: autor)

Graf	Řád grafu	Velikost grafu	Výpočetní čas [s]	Chromatická sekvence
queen5_5	25	160	12,95	5 5 4 3 3 3 2 2 2 2 2 2 2 2 1
queen6_6	36	290	18434,66	7 6 4 4 3 3 3 2 2 2 2 2 2 2 2 2 2 2 2 1
queen7_7	49	476	TO	
queen8_8	64	728	TO	
myciel4	11	20	1,80	4 2 2 2 2 1
myciel5	23	71	4,36	5 3 2 2 2 2 2 2 2 2 2 1
myciel6	47	236	18783,82	6 4 3 3 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1
myciel7	95	755	TO	
K4x7	28	294	11,29	4 4 4 4 4 4 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1
K7x4	28	336	700,27	7 7 7 6 4 4 4 3 3 3 3 3 2 2 2 2 2 2 2 2 2 2 2 2 1
K4x11	44	726	23,24	4 4 4 4 4 4 4 4 4 3 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1
K11x4	44	880	TO	
K7x11	77	2541	TO	
david	87	406	16135,74	11 6 5 4 3 3 3 3 3 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1
jean	80	254	1357,69	10 5 4 3 3 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1
huck	74	301	2906,31	11 6 5 4 3 3 3 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1

Důvodem takto velké odchylky ve výpočetním čase co do velikosti grafu může být struktura grafu a také počáteční zabarvení grafu v prvním kroku algoritmu, které je pro následné řešení pomocí metody *Branch and Bound* zásadní (podobné časy však byly zaznamenány i případě několika opakováních testu). Struktura grafu $K4x11$ je velmi pravidelná.

Z výpočtů můžeme předpokládat, že dlouhé výpočetní časy frekvencí jsou povětšinou způsobovány pouze několika chromatickými čísly, konkrétněji $\chi_b^{\Delta}(G)$ pro $b = (2,3)$. Chromatická čísla χ_0^{Δ} *Δ-chromatickýh sekvencí* mnoha benchmark a kompletních

multipartitních grafů jsou spočítána v řádu několika sekund, zatímco chromatická čísla χ_2^Δ a χ_3^Δ stejných grafů bylo vyřešit časově náročné. Tento trend je viditelný napříč všemi skupinami testovaných grafů.

Přehled všech optimálních sekvencí grafů je v Tabulce 4. Tři ze zkoumaných grafů ($queen6\_6$, $K4x7$, $huck$) jsou co do své velikosti srovnatelné. Ta se u všech tří grafů pohybuje okolo 300 hran. Pokud však začneme srovnávat parametry výpočtu jako je výpočetní čas nebo počet iterací, je rozdíl mezi grafy zásadní. χ_0^Δ bylo v porovnání s ostatními chromatickými čísly spočteno relativně rychle. Ale i pro χ_0^Δ se časy výpočtů a počet provedených iterací významně liší. V případě $K4x7$ bylo k obdržení tohoto chromatického čísla zapotřebí pouze 207 iterací a méně než 0,5s výpočetního času. Výpočetní čas $queen6\_6$ pro χ_0^Δ je pak více než desetkrát delší a počet iterací se pohybuje okolo 170 000. Čas výpočtu pro graf $huck$ je pak desetkrát delší než u $queen6\_6$ a počet iterací zde narůstá na téměř pětinásobek.

Zajímavé jsou pak počty iterací a výpočetní časy grafu $queen6\_6$ pro chromatická čísla χ_1^Δ a χ_2^Δ. Ty jsou totiž mnohem vyšší, než pro ostatní dva grafy. Čas výpočtu těchto dvou chromatických čísel u grafu $K4x7$ zůstává pod hranicí 1 sekundy a počet iterací se pohybuje okolo 3000. Čas výpočtu pro $huck$ se naproti tomu blíží 47 minutám spolu s více než $4{,}6 \cdot 10^7$ iterací. Tyto hodnoty jsou pro $queen6\_6$ ještě vyšší. Výpočetní čas překračuje 4 hodiny a počet iterací je se svojí hodnotou $3{,}2 \cdot 10^8$ téměř sedmkrát vyšší. Přestože je, jak již bylo zmíněno, velikost těchto tří grafů velmi podobná, výpočetní čas a potřebný počet iterací byl nejvyšší pro graf $queen6\_6$, který má zároveň méně hran než graf $huck$ a jeho hustota hran je tudíž vyšší, avšak stále má více vrcholů než graf $K4x7$.

Po porovnání všech výsledků a záznamů testování grafů exaktním matematickým modelem můžeme konstatovat, že velikost graf či hustota hran v grafu mají přímý vliv na čas výpočtu a dosažení optimální *Δ-chromatických sekvencí*. Jak však bylo pozorováno v případě kompletního multipartitního grafu $K4x7$, čas výpočtu se oproti grafům s obdobnou velikostí velmi významně lišil. Lze tedy předpokládat existenci dalšího faktoru, který má na čas výpočtu vliv. Mohla by jím být například struktura grafu, která je v případě $K4x7$ pravidelná. Pokud však srovnáme dva podobné grafy $K4x11$ a $K11x4$, vidíme, že v případě $K11x4$ byl při výpočtu chromatické sekvence překročen časový limit 24 hodin, kdežto u druhého grafu $K4x11$ byla sekvence spočtena během 22 sekund. Tyto

grafy se liší ve svém řádu a v počtu podmnožin vrcholů. Graf $K11x4$ má vyšší počet podmnožin nižšího řádu, a celkově oproti obdobnému $K4x11$ větší velikost grafu (cca 100 hran). Poměr výpočetních časů Δ-chromatických sekvencí těchto dvou grafů je však vzhledem k poměru velikosti grafů disproporcionální. Stejný trend je pozorovatelný při srovnání jiných dvou vzájemně podobných kompletních multipartitních grafů, a to $K4x7$ a $K7x4$. Jedním z důvodů by mohlo být nevhodně zvolené počáteční zabarvení grafu, které by způsobilo neefektivní průběh řešení *branch and bound* metodou. Jiné možné zdůvodnění nebylo v této práci odhaleno.

8.3 TESTOVÁNÍ NÁHODNĚ GENEROVANÝCH GRAFŮ

Soubor 75 grafů náhodně vygenerovaných programem Mathematica byl testován totožným způsobem, jako všechny předchozí skupiny grafů. Pro každou skupinu náhodně vygenerovaných grafů $G_{n,p}$, byl zaznamenán čas výpočtu a počet. Každá z těchto skupin obsahuje právě pět grafů. Exaktní metoda i model s časovým omezením byly použity pro všechny náhodně vygenerované grafy. Pro mnoho z nich však nebyly zjištěny optimální *Δ-chromatické sekvence*, pravděpodobně z důvodu komplexnosti grafů. Všechny skupiny grafů řádu $i = 25$ byly vyřešeny exaktním přístupem a byly pro ně obdrženy optimální *Δ-chromatické sekvence*. V případě skupiny grafů řádu $i = 40$ byly optimální *Δ-chromatické sekvence* zjištěny pouze pro grafy s hustotou hran *h= 0,3*, tudíž pouze pro skupinu $G_{40,0.3}$. Výpočetní čas všech grafů s vyšší hustotou hran překročil hranici 12 hodin, a tudíž byl výpočet přerušen. Stejný trend byl zaznamenán pro všechny ostatní grafy vyšších řádů $i = \{55, 70, 85\}$. Tento trend se jeví býti stejným, jako tomu bylo v případě benchmark grafů a kompletních multipartitních grafů. Průměrné velikosti grafů $\bar{q}$ v jednotlivých souborech grafů jsou: $\bar{q}_{G_{40,0.3}} = 240$, $\bar{q}_{G_{40,0.5}} = 400$ a $\bar{q}_{G_{55,0.3}} = 450$. Všechny ostatní skupiny grafů řádu vyššího nežli $i = 25$ mají průměrnou velikost graf $\bar{q}$ vyšší. Hraniční velikost grafu, za kterou již zpravidla nejsou získávány optimální řešení, je tak velmi podobná hranici, která byla pozorována v případě testování benchmark grafů a kompletních multipartitních grafů exaktní metodou.

Tabulka 5 - Porovnání časů výpočtů grafů podobných parametrů (Zdroj: autor)

Graf	n	q	χ_0^Δ	Výpočetní čas [s]	χ_1^Δ	Výpočetní čas [s]	χ_2^Δ	Výpočetní čas [s]
queen6_6	36	290	**7**	2,8	**6**	15714,6	**4**	2415,3
myciel6	47	236	**6**	18760	**4**	3,4	**3**	1,8
$G_{40,0.3}$ #1	40	240	**6**	14,7	**4**	2,3	**4**	131,2
$G_{40,0.3}$ #2	40	223	**6**	24,5	**4**	5,8	**4**	155,6
$G_{40,0.3}$ #3	40	235	**6**	4	**4**	167,1	**4**	484,2
$G_{40,0.3}$ #4	40	232	**6**	14,5	**4**	1,4	**4**	155,7
$G_{40,0.3}$ #5	40	237	**6**	10,4	**5**	257,9	**4**	8,9

Tato skutečnost nabízí srovnání grafů z $G_{40,0.5}$ a benchmark grafu $queen7\_7$, jelikož řády grafů i velikosti jsou podobné. Ani u jednoho z těchto grafů nebylo zjištěno optimální řešení. Další možností je srovnat grafy ze skupiny $G_{40,0.3}$ spolu s benchmark grafy $queen6\_6$ a $myciel6$, jak je ukázáno v Tabulce 5. $queen6\_6$ má v porovnání s náhodně generovanými grafy o něco nižší počet vrcholů, a naopak větší velikost grafu. Velikost grafu $myciel6$ je téměř totožná s velikostí generovaných grafů, avšak počet vrcholů je vyšší. Parametry těchto grafů a časy výpočtů jednotlivých χ_b^Δ kde $b = 1,2,3$ jsou vypsány v Tabulce 5.

Výsledky testování náhodně generovaných grafů ukazují, že výpočetní čas je závislý na kombinaci velikosti grafu a jeho řádu. U těchto grafů je pozorován stejný trend jako v předchozí podkapitole. U náhodně generovaných grafů však nebyl zaznamenán žádný graf, který by se tomuto trendu nějak významně vymykal, jako tomu bylo v případě grafu $K4x11$. U skupin náhodně generovaných grafů stejného řádu roste čas výpočtů spolu s rostoucí hustotou hran v grafu. Pro grafy ze skupin $G_{40,0.5}$ a $G_{55,0.3}$ nebylo obdrženo optimální řešení, a to i přes skutečnost, že jejich velikost byla podobná jako grafu $david$. Hustota hran v grafu $david$ je však v porovnání s hustotou hran v náhodných grafech nízká, a to $h_{david} = 0,109$. Všechny popsané výsledky naznačují, že čas výpočtů je závislý na kombinaci řádu grafu a hustoty hran v grafu, nicméně není známo, v jakém vzájemném vztahu tyto dva faktory jsou. Jako jedno z vysvětlení se nabízí struktura grafu. Tato domněnka má reálné základy ve výpočetních časech kompletních multipartitních grafů stejného řádu a podobných velikostí, které se zásadně liší pravděpodobně právě z důvodu odlišné struktury grafu. Celkem pak byly zjištěny optimální *Δ-chromatické sekvence* pouze pro 20 náhodných grafů. Pro všechny ostatní jsou k dispozici pouze výstupy modelu s časovým omezením.

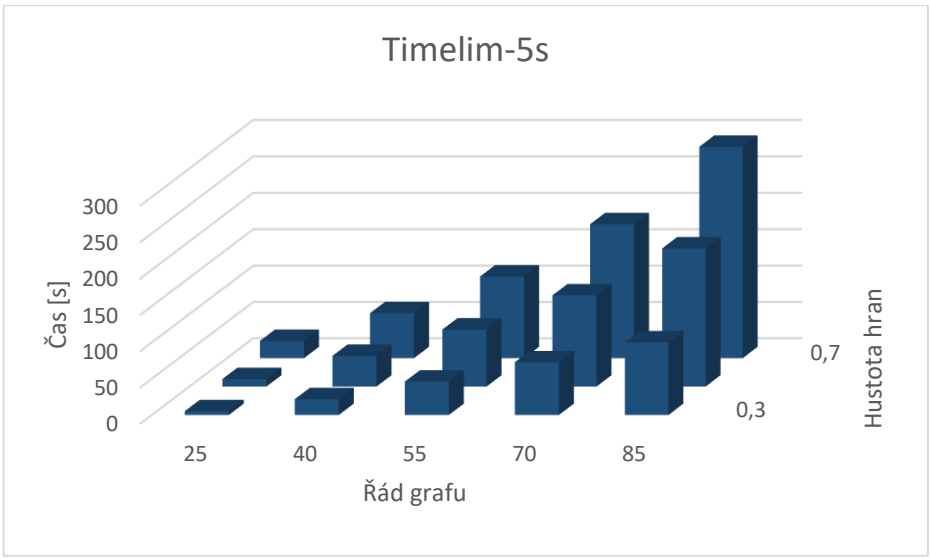
Na Obrázku 16 a 17 lze pozorovat zajímavé jevy vývoje počtu iterací a výpočetních časů v závislosti na parametrech grafů. V případě časů je předvídatelné, že s narůstajícím počtem vrcholů a hran v grafu poroste i výpočetní čas. Tento předpoklad je potvrzen i při použití modelu se všemi 3 časovými omezeními. Také výsledky exaktního přístupu ukazují stejný trend. V případě počtu iterací je však vývoj odlišný. Maximální počet iterací nebyl zaznamenán jen pro grafy $G_{85,h}$, jako tomu bylo v případě časů výpočtů, ale i pro grafy jiných řádů a hustot hran.

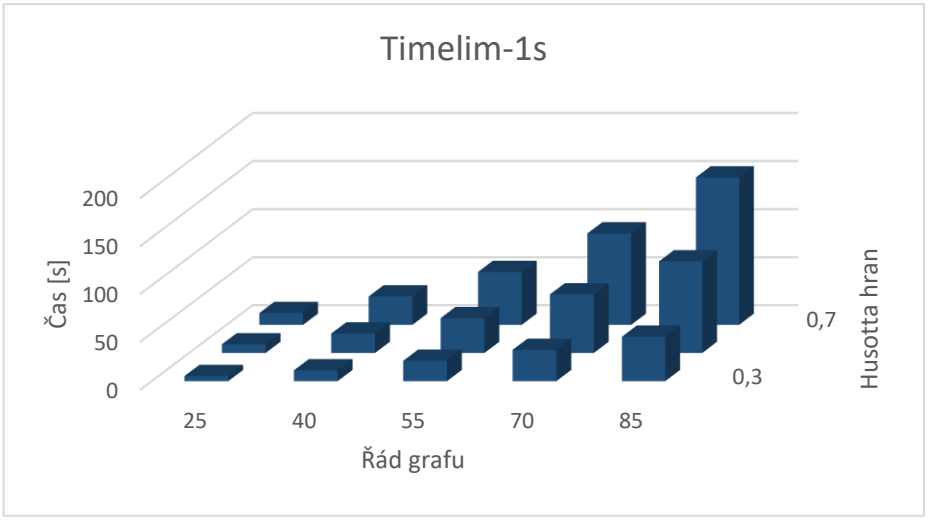
Popis pohybu maximálních počtů iterací při výpočtech *Δ-chromatických sekvencí* náhodně generovaných grafů různých řádů a hustot hran může být popsán následovně:

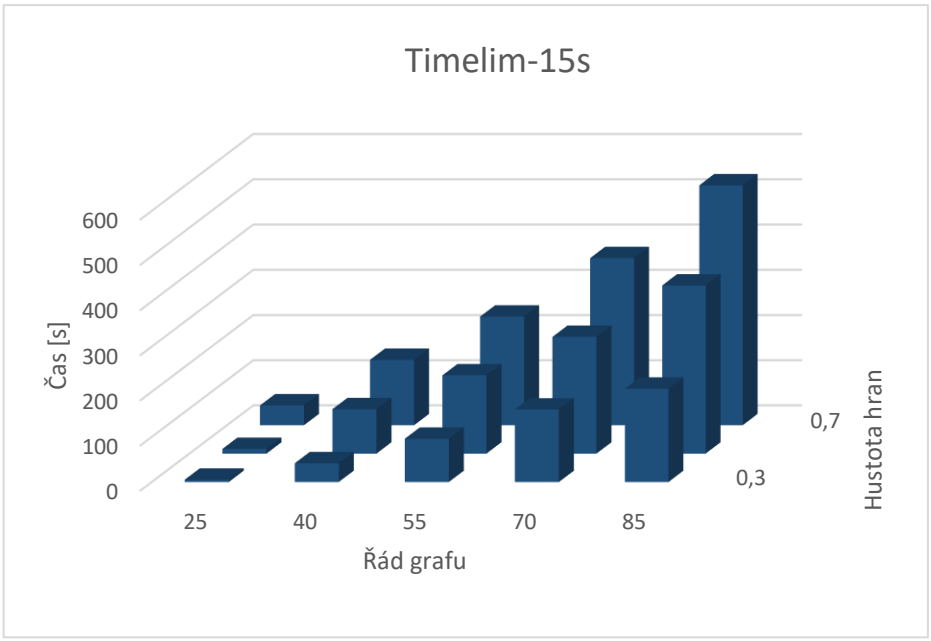
- Se snižujícím se časovým limitem pro výpočet jednotlivých $\chi_b^{\Delta}(G_{i,h})$ se maximální počet iterací pohybuje směrem ke grafům s menším počtem vrcholů.
- S rostoucí hustotou hran v grafu a stejným časovým limitem pro výpočet jednotlivých $\chi_b^{\Delta}(G_{i,h})$ se maximální počet iterací pohybuje směrem ke grafům s nižším počtem vrcholů.
- S rostoucím řádem grafu a stejným časovým limitem pro výpočet jednotlivých $\chi_b^{\Delta}(G_{i,h})$ se maximální počet iterací pohybuje směrem ke grafům s nižší hustotou hran.

Z výsledků lze pozorovat, že nárůst počtu iterací je nižší v případě změny hustoty hran grafu, než v případě nárůstu počtu vrcholů grafu. Pohyby maximálních počtů iterací, popsané v předchozích 3 bodech, ukazují efektivitu jednotlivých délek intervalů časových omezení pro grafy specifických parametrů. Snižování maximálního počtu iterací během zvyšování hustoty hran grafu či jeho řádu naznačuje, že graf je pro specifický časový limit příliš náročný. Průběh výpočtu je tedy zastavován již během prvních časově náročných iterací. Proto by mělo být zváženo upravení časového limitu.

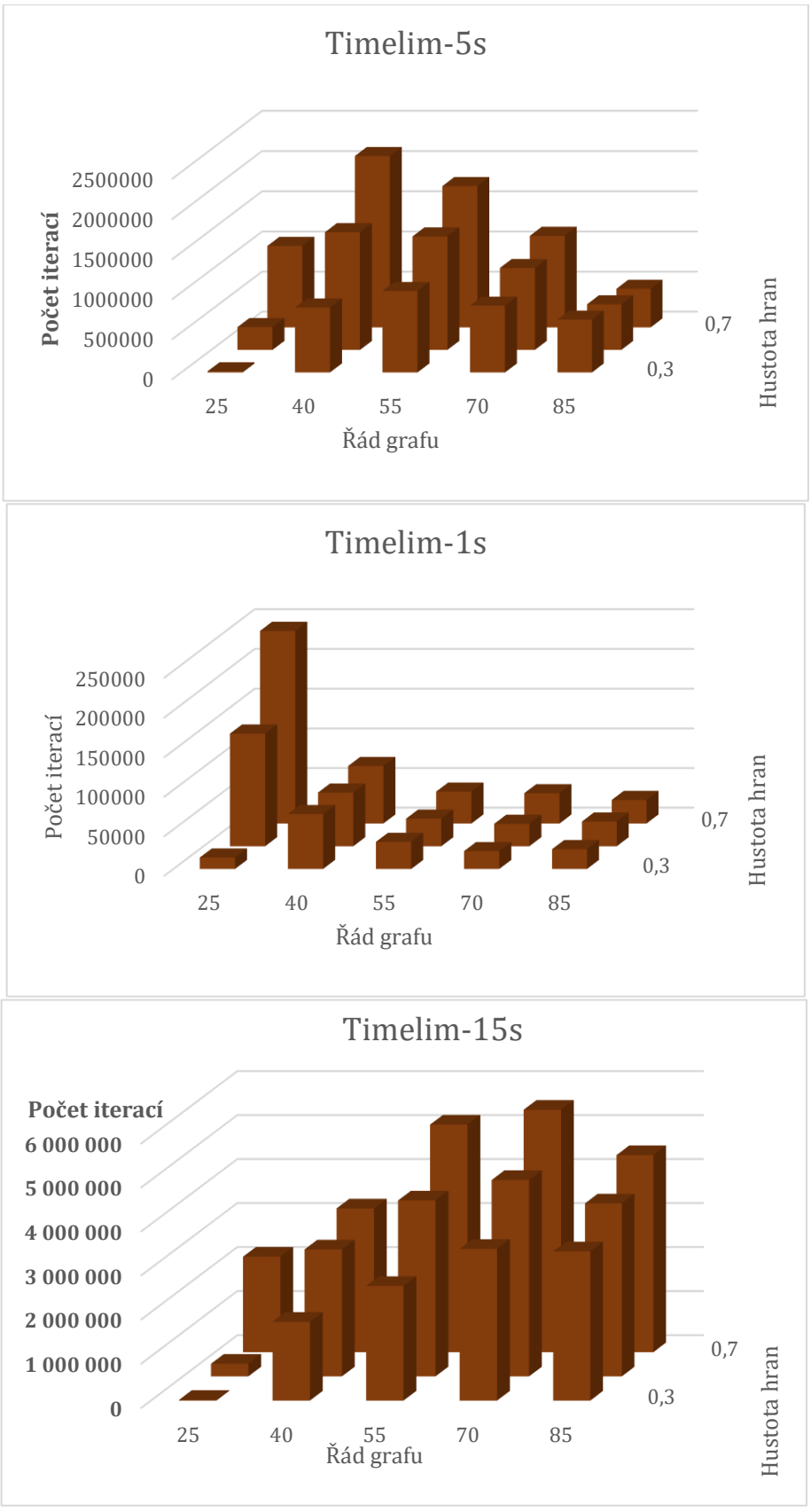
Přesnost modelu může srovnávat pouze u těch grafů, pro které bylo získáno optimální řešení. Z 20 grafů se známými optimálními *Δ-chromatickými sekvencemi* jich 18 při použití *Timelim-15s* obdrželo *Δ-chromatickou sekvenci* totožnou se sekvencí optimální. Pouze sekvence grafů $G_{25,0.7}$#4 a $G_{40,0.3}$#3 se od těch optimálních lišily. V obou případech to byl druhý člen sekvence χ_1^{Δ}. 18 grafů tedy obdrželo řešení totožné s optimálním při použití *Timelim-15s*, 16 grafů při použití *Timelim-5s* a 12 grafů při použití *Timelim-1s*. Pouze 2 grafy tedy obdržely lepší řešení při použití *Timelim-15s* než při použití *Timelim-5s*.







Obrázek 16 – Grafy znázorňující vývoj časů výpočtů vzhledem k řádu grafu a hustotě hran (Zdroj: autor)



Obrázek 17 – Grafy znázorňující vývoj iterací výpočtů vzhledem k řádu grafu a hustotě hran (Zdroj: autor)

8.4 SROVNÁNÍ S OSTATNÍMI ALGORITMY

Pro porovnání výstupů modelu této práce a dalších algoritmů řešících MDCP byly použity výstupy algoritmů *Colouring degree algorithm* (dále jen CDA), *Tabu search algorithm* (dále jen TSA) a χ_b^{Δ} *irredundant–colouring algorithm* (dále jen ICA)z dizertační práce Isabelle Niewoudt [9]. Z této práce pochází taktéž exaktní *Critical* χ_b^{Δ} *– colouring algorithm* (dále jen CCA), avšak pro ten nejsou v [9] uvedeny žádné výstupy pro benchmark grafy a kompletní multipartitní grafy. Dva exaktní algoritmy [9] jsou porovnávány s matematickým modelem této práce bez stanoveného časového limitu pro výpočet jednotlivých χ_b^{Δ}. Dva heuristické algoritmy [9], CDA a TSA jsou srovnávány s *Timelim-1s*, *Timelim -5s* a *Timelim -15s*.

CDA byl testován v 5 opakováních, tudíž byly také v práci uvedeno 5 různých výsledků. Pro potřeby srovnávání byly vybrány výsledky z pátého opakování, protože se nejvíce blížily optimálnímu řešení a byly tak ze všech opakování nejlepší. U TCA byly výsledky vzaty z prvního opakování, jelikož v ostatních případech v *Δ-chromatických sekvencích* chyběl vyšší počet chromatických čísel. Pro CDA nebyly v [9] zaznamenány žádné časy výpočtů, proto můžeme srovnat pouze přesnost algoritmů porovnáním výstupních sekvencí. Výsledky a informace o výpočtech exaktních barvících algoritmů z [9] byly zaznamenány pouze pro první 3 členy sekvence, tudíž nemůžeme porovnávat celé sekvence. Podrobné informace o všech algoritmech, se kterými budou výsledky v této kapitole srovnávány, jsou podrobně popsány v [9]. Veškeré srovnání algoritmů a modelů je založeno na informacích o výpočtech sekvencí benchmark a kompletních multipartitních grafů. Časové hranice pro ICA a CCA byly podle předlohy práce Isabelle Niewoudt [9] stanoveny na 3 hodiny. Časová hranice pro exaktní přístup v případě matematického modelu z této práce pak na 12 hodin, jak již bylo zmíněno dříve. Následující text bude podrobněji analyzovat informace z Tabulky 6 a Tabulky 7.

Pro potřeby srovnání heuristických přístupů a modelu s časovým omezením byly sledovány počty iterací a časy výpočtů. Nejrychlejším algoritmem byl CDA. Tento algoritmus získal pro jednotlivé grafy řešení v řádu sekund. V některých případech byl až několikrát rychlejší než TSA nebo *Timelim-1s*. Pro CDA nejsou u kompletních multipartitních grafů zaznamenány v [9] žádné výsledky. Lze jen předpokládat, zda by i v tomto případě byly časy výpočtů kratší než při použití ostatních algoritmů.

CDA tedy sice je nejrychlejším z algoritmů, nicméně jeho výstupy jsou velmi nepřesné. Pouze v jednom případě byla *Δ-chromatická sekvence* totožná s optimální *Δ-chromatickou sekvencí* daného grafu, a to pro graf $myciel4$ řádu $i = 11$. Vypočtené sekvence kompletních multipartitních grafů dokonce nesplnily premisu nerostoucích $\chi(G)$ v sekvenci. Všechny *Δ-chromatické sekvence* testovaných grafu se vyjma sekvence grafu $myciel4$ liší od optimálních *Δ-chromatický sekvencí* alespoň ve dvou členech. Sekvence grafů $queen5\_5$ a $myciel6$ pak stejně jako v případě kompletních multipartitních grafů nesplňují premisu nerostoucích $\chi(G)$. TSA je ve srovnání s CDA co do výsledků přesnější. Dva grafy použitím tohoto algoritmu obdržely *Δ-chromatické sekvence* totožné se sekvencemi optimálními a všechny z vypočtených sekvencí byly nerostoucího charakteru. Časy výpočtů jsou však neporovnatelně vyšší. Pouze šest grafů obdrželo *Δ-chromatické sekvence* v čase nižším než 1 minuta. Výpočetní časy sedmi dalších grafů pak dokonce přesáhly 200 sekund.

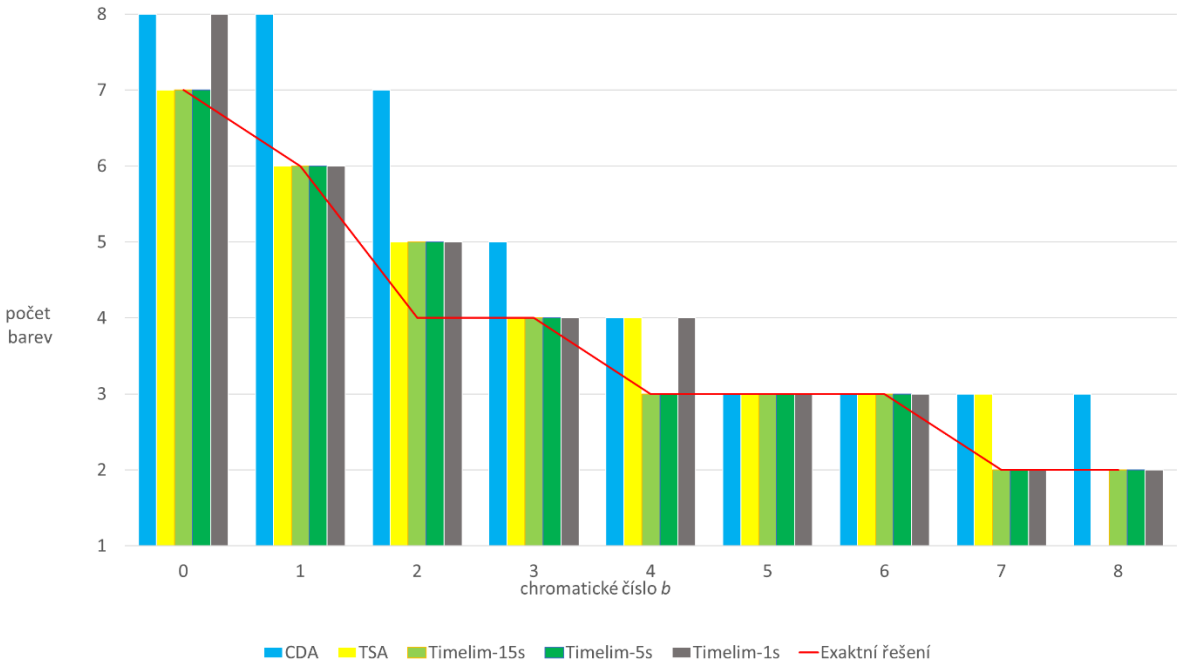
Model s různým časovým omezením z této práce se umístily co do časů výpočtů mezi algoritmy z [9]. Převážná většina sekvencí byla modelem s časovým omezením vyřešena rychleji, než při použití TSA, nicméně CDA je stále neporovnatelně rychlejší. Výpočetní časy *Timelim-1s* se ve většině případů pohybují do jedné minuty. Výpočetní časy *Timelim-5s* pak jsou samozřejmě vyšší, nicméně většina z nich se stále drží pod hranicí jedné minuty. Výsledky *Timelim-15s* byly známy z valné části do 2 minut. Nejdelší čas výpočtu pro algoritmy z [9] byl pro graf $myciel7$. Pro model s časovým omezením to pak byl graf $david$. Co do délky časů výpočtů je pak pořadí algoritmů následující:

1. CDA
2. *Timelim-1s*
3. *Timelim-5s*
4. *Timelim -15s*
5. TSA

Budeme-li srovnávat přesnost jednotlivých algoritmů, pořadí se změní. Jak již bylo řečeno, co do přesnosti má CDA jen chabé výsledky. Většina z vypočtených sekvencí se oproti těm optimálním velmi významně liší. Tento algoritmus tedy může sloužit pro rychlý odhad sekvence, kde více jak na přesnosti výsledku záleží na času výpočtu.

Timelim-5s obdržel sekvenci totožnou se sekvencí optimální v 11 případech ze 13 (benchmark a kompletních multipartitních grafů). Ve 2 zbývajících případech se sekvence lišila od té optimální pouze v jednom členu $\chi(G)$ a pouze o hodnotu 1. Sekvence *Timelim-5s* a *Timelim-15s* se odlišují pouze v případě grafu *queen*7_7. I zde jsou ale vypočtené členy sekvencí nižší než v případě CDA či TSA.

Lze konstatovat, že žádná z *Δ-chromatických sekvencí* vypočtených algoritmy CDA a TSA nebyla optimální sekvenci blíže než *Δ-chromatická sekvence* získaná matematickým modelem s časovým omezením, představeným v této práci. Přesnost jednotlivých algoritmů pro graf *queen*6_6 je ukázána na Obrázku 18. Tento graf byl vybrán, protože ani *Tiemlim-15s* pro něj nezískal sekvenci, která by byla totožná se sekvencí optimální.



Obrázek 18 – Porovnání chromatických sekvencí grafu queen6_6 vypočtených heuristickými algoritmy, modelem s různým časovým omezením a exaktním přístupem (Zdroj: autor)

Pro srovnání exaktních přístupů byl srovnáván pouze čas výpočtů. Iterace v případě algoritmů z [9] mají úplně jiný charakter, než iterace zaznamenané pro celočíselný matematický model. Výpočetní časy ICA se od těch uvedených v [9] liší z důvodu opětovného spuštění algoritmu na stejném zařízení, na kterém byl testován celočíselný model.

Kompletní multipartitní grafy $K4x7$, $K4x11$ a $K7x4$ jsou pro $b = 0$ řešeny o něco rychleji pomocí ICA s časy nepřekračujícími jednu sekundu. Ani celočíselný model však tuto hranici u jmenovaných grafů nepřekračuje. Rozdíl je zde oproti časům výpočtů u jiných grafů téměř zanedbatelný. V případě grafů $david$, $jean$ a $huck$ je ICA významně rychlejší a časy výpočtů zde nepřekračují 5 sekund. Každý z prvních 3 členů sekvence grafu $jean$ je spočítán rychleji pomocí ICA. Graf $myciel6$ však obdržel členy *Δ-chromatické sekvence* pouze při použití celočíselného modelu. Algoritmus z [9] nevypočítal hodnotu prvního členu sekvence do limitu 3 hodin, a tudíž nebyl spočten ani žádný z následujících členů. Stejná situace je pozorovatelná v případech $\chi_2^{\Delta}(queen6\_6)$, $\chi_2^{\Delta}(huck)$, $\chi_2^{\Delta}(K7x4)$ nebo $\chi_1^{\Delta}(david)$. Ve všech těchto případech byly členy $\chi_b^{\Delta}(G)$ získány pouze prostřednictvím celočíselného modelu.

U chromatického čísla $\chi_1^{\Delta}(huck)$ byl co do časů výpočtů velký rozdíl. Významné rozdíly v časech výpočtů jednotlivých členů sekvencí jsou i v mnoha dalších případech, avšak u $huck$ byl tento čas v případě algoritmu ICA velmi nízký (30,01s), kdežto u celočíselného modelu se čas výpočtu pohyboval okolo 3000 sekund. Čas výpočtu ICA je ve srovnání s ostatními časy sekvence neobvyklý. Jak ICA, tak celočíselný model jsou co do účinnosti srovnatelné a poskytují řešení v podobných časových intervalech. Žádný z těchto dvou řešících přístupů se z pohledu rychlosti nezdá významně lepší než druhý. Výhodou celočíselného modelu však je, že v několika případech obdržel výsledek, avšak ICA dosáhl svého časového limitu a výpočet tak byl ukončen. Tento případ je pozorovatelný u čtyř členů $\chi_b^{\Delta}(G)$. Ve dvou z nich pak celočíselný model sice získal řešení, nicméně čas výpočtu byl delší než časový limit ICA.

Tabulka 6 Srovnání výsledků a časů výpočtů exaktních algoritmů Zdroj: autor)

Srovnání exaktních algoritmů

Jméno grafu	Velikost	Řád	b=0				b=1				b=2			
			ICA		Exaktní model		ICA		Exaktní model		ICA		Exaktní model	
			Chromatické číslo	Čas výpočtu [s]	Chromatické číslo	Čas výpočtu [s]	Chromatické číslo	Čas výpočtu [s]	Chromatické číslo	Čas výpočtu [s]	Chromatické číslo	Čas výpočtu [s]	Chromatické číslo	Čas výpočtu [s]
myciel6	*236*	*47*	**6**	TO	**6**	18760,03		---	**4**	3,41		---	**3**	1,88
queen6	*290*	*36*	**7**	27,73	**7**	2,77	**6**	798,41	**6**	15714,60			**4**	2415,34
queen7	*476*	*49*	**7**	75,40	**7**	7,30				TO				---
Huck	*301*	*74*	**11**	5,87	**11**	32,84	**6**	0,00	**6**	2827,35		TO	**4**	2,67
Jean	*254*	*80*	**10**	6,36	**10**	29,16	**6**	340,43	**5**	1291,94	**4**	**3,27**	**4**	4,20
David	*406*	*87*	**11**	11,67	**11**	56,94	**7**	TO	**6**	4799,53		---	**5**	10318,68
K4x7	*294*	*28*	**4**	0,64	**4**	0,27	**4**	0,51	**4**	0,81	**4**	58,22	**4**	0,83
K4x11	*726*	*44*	**4**	3,22	**4**	0,26	**4**	6,62	**4**	1,49	**4**	211,79	**4**	1,21
K7x4	*336*	*28*	**7**	1,09	**7**	0,23	**7**	805,36	**7**	1,89		TO	**7**	472,76

			ICA		Exaktní model	
			chromatická sekvence	Čas výpočtu [s]	chromatická sekvence	Čas výpočtu [s]
myciel4	*20*	*11*	4 2 2 2 2 1	0,30	4 2 2 2 2 1	1,80
myciel5	*71*	*23*	5 3 2 2 2 2 2 2 2 2 2 1	74,16	5 3 2 2 2 2 2 2 2 2 2 1	4,36
queen5	*160*	*25*	5 5 4 3 3 3 2 2 2 2 2 2 2 1	30,08	5 5 4 3 3 3 2 2 2 2 2 2 2 1	12,95

Tabulka 7 - Srovnání výsledků heuristických algoritmů a modelu s časovým omezením (Zdroj: autor)

Srovnání s heuristickými algoritmy												
				CDA		TSA		Timelim-1s		Timelim-5s		Timelim-15s	
Graf	Velikost	Řád	Optimální sekvence	Vypočtená sekvence	Čas výpočtu [s]	Vypočtená sekvence	Čas výpočtu [s]	Vypočtená sekvence	Čas výpočtu [s]	Vypočtená sekvence	Čas výpočtu [s]	Vypočtená sekvence	Čas výpočtu [s]
queen5	160	25	**5 5 4 3 3 3 2 2**	5 7 5 4 3 3 3 2	0,45	**5 5 4 3 3 3 2 2**	28,26	**5 5 4 3 3 3 2 2**	8,70	**5 5 4 3 3 3 2 2**	8,70	**5 5 4 3 3 3 2 2**	8,70
queen6	290	36	**7 6 4 4 3 3 3 2**	8 8 7 5 4 3 3 3	1,31	8 6 5 4 4 3 3 2	92,67	8 6 5 4 4 3 3 2	13,33	7 6 5 4 3 3 3 2	27,18	7 6 5 4 3 3 3 2	56,68
queen7	476	49		10 8 7 5 4 4 3 3	3,48	9 7 6 5 4 4 3 3	207,71	9 8 6 5 4 4 3 3	18,44	8 7 6 5 4 4 3 3	45,65	7 7 6 5 4 4 3 3	97,12
myciel4	20	11	**4 2 2 2 2 1**	**4 2 2 2 2 1**	0,01	**4 2 2 2 2 1**	0,54	**4 2 2 2 2 1**	1,80	**4 2 2 2 2 1**	1,80	**4 2 2 2 2 1**	1,80
myciel5	71	23	**5 3 2 2 2 2 2 2**	5 3 3 3 2 2 2 2	0,17	5 3 3 3 2 2 2 2	4,88	**5 3 2 2 2 2 2**	4,36	**5 3 2 2 2 2 2**	4,36	**5 3 2 2 2 2 2 2**	4,36
myciel6	236	47	**6 4 3 3 3 2 2 2**	6 4 3 4 3 3 2 2	1,47	6 4 3 3 3 3 3 2	54,47	**6 4 3 3 3 2 2 2**	13,67	**6 4 3 3 3 2 2 2**	18,81	**6 4 3 3 3 2 2 2**	28,03
queen8	728	64		11 9 8 6 5 5 4	7,61	10 8 7 5 5 4 4	444,21	12 9 8 6 5 4 4 4	28,82	11 9 7 6 5 4 4 3	67,09	11 9 7 6 5 4 4 3	122,60
myciel7	755	95		7 6 6 5 4 4 3	15,33	7 6 5 4 4 4 4	573,55	7 5 4 4 4 4 4 3	63,23	7 5 4 4 3 3 3 3	100,42	7 5 4 4 3 3 3 3	140,46
Huck	301	74	**1 1 6 4 4 3 3 3 3**	11 6 5 4 4 3 3	6,03	11 6 4 4 4 3 3	239,18	11 6 5 4 3 3 3 3	46,87	**1 1 6 4 4 3 3 3 3**	57,76	**1 1 6 4 4 3 3 3 3**	80,42
Jean	254	80	**1 0 5 4 3 3 3 2 2**	10 7 5 4 3 3 3	6,46	11 6 4 4 3 3 3	208,50	10 6 4 3 3 3 2 2	35,11	**1 0 5 4 3 3 3 2 2**	46,71	**1 0 5 4 3 3 3 2 2**	65,70
David	406	87	**1 1 6 5 4 3 3 3 3**	11 7 6 5 4 3 3	12,00	12 7 5 5 4 4 4	445,51	11 7 5 5 4 4 3 3	89,31	11 6 5 4 4 3 3 3	110,05	11 6 5 4 4 3 3 3	148,70
K4x7	28	294	**4 4 4 4 4 4 3 2**	4 14 8 6 4 4 4 2	0,65	8 11 8 5 5 4 4 3	69,24	**4 4 4 4 4 4 3 2**	11,29	**4 4 4 4 4 4 3 2**	11,29	**4 4 4 4 4 4 3 2**	11,29
K7x4	28	336	**7 7 7 6 4 4 4 3 3 3 3 2**	7 13 7 7 4 4 4 4	1,14	8 11 7 7 4 5 4 4	106,31	**7 7 7 6 4 4 4 3**	14,11	**7 7 7 6 4 4 4 3**	23,58	**7 7 7 6 4 4 4 3**	43,27
K4x11	44	726	**4 4 4 4 4 4 4 4 4 4 3 2**	4 22 12 8 6 6 4 4	3,27	10 9 11 7 7 6 5 5	326,33	**4 4 4 4 4 4 4 4**	23,24	**4 4 4 4 4 4 4 4**	23,24	**4 4 4 4 4 4 4 4**	23,24
K11x4	44	880		11 21 11 11 6 6 6 6	6,30	16 20 - 9 7 7 - 6	484,87	11 11 11 10 6 6 6 5	34,39	11 11 11 9 6 6 6 5	62,42	11 11 11 9 6 6 6 5	136,74
K7x11	77	2541		7 34 19 13 10 10 7 7	46,79		3195,64	7 7 7 7 7 7 7	89,73	7 7 7 7 7 7 7	220,54	7 7 7 7 7 7 7	354,45

9 ZÁVĚR

Cílem této práce bylo popsat současný stav v oblasti teorie barvení grafů, a vytvořit matematický model řešící problém defektivního barvení grafů. Model byl testován a srovnán s jinými algoritmy řešícími stejný typ úloh.

V první části práce je popsán vznik problematiky barvení grafů pomocí shrnutí historie „problému 4 barev“ a vývojem stejnojmenného teorému, který byl matematicky dokázán až více než 100 let od svého vzniku. Na něm byl zevrubně popsán klasický problém barvení grafu.

Rešeršní část měla za úkol shrnout teoretické poznatky z různých oblastí teorie barvení grafů. Publikované práce a vědecké články, zaměřené na problematiku barvení grafů, se v teoretických částech zaměřují zejména na stanovení podmínek zabarvitelnosti. Dalším zkoumaným faktorem je stanovení horní hranice minimálního počtu barev, kterými lze zabarvit libovolný graf. V této části jsou také nastíněny některé zákonitosti platící pro Δ-*chromatické sekvence*, se kterými pracuje i matematický model představený v této práci. Obecným pravidlem podoby chromatických sekvencí je skutečnost, že jejich členy po sobě následují v nerostoucím pořadí a že určitý člen sekvence a všechny členy po něm následující budou rovny hodnotě 1. Další části rešerše byly věnovány algoritmům určeným k barvení grafů. Bylo představeno několik exaktních algoritmů a popsán jejich vývoj spolu s postupným vylepšováním různými autory. V části heuristických algoritmů bylo zmíněno několik typů přístupů k řešení a provedeno porovnání účinnosti jednotlivých metod. Bylo nalezeno velké množství algoritmů řešících různé variace úlohy barvení grafů.

Matematický model určený k řešení úloh defektivního barvení grafů byl představen kapitole 4. Model je sestaven tak, aby byl řešitelný metodou celočíselného lineárního programování. Samotná podoba modelu je velmi obecná, a tudíž ho lze řešit velkým počtem nástrojů a programů. Model je tvořen účelovou funkcí, která má za úkol minimalizovat počet barev použitých k zabarvení vstupního grafu, a několika omezujícími podmínkami. Omezující podmínky pak v sobě zahrnují pravidla zabarvování. Jedna z podmínek povoluje modelu zabarvit každý vrchol právě jednou barvou. Druhá z podmínek pak představuje pravidlo zabarvování, které říká, že vrchol n zabarvený

barvou *c* může sousedit s nanejvýše *b* vrcholy zabarvenými toutéž barvou. V modelu byly rovněž použity dvě binární proměnné. Jedna z nich má za úkol monitorovat použití dané barvy, druhá pak sleduje, jaký vrchol byl zabarven jakou barvou. Model byl převeden do programovacího kódu a implementován do softwaru IBM ILOG CPLEX Optimization Studio, který platí za jednoho z nejlepších řešitelů úloh lineárního programování. Vytvořený algoritmus pak byl schopen počítat celé Δ-*chromatické sekvence* pomocí iterativního procesu, který po každém nalezeném řešení stanovenou hodnotu *b* zvýšil tento parametru o 1 a celý proces výpočtu opakoval, dokud nebylo dosaženo zabarvení celého grafu pouze jednou barvou. Vstupem do algoritmu byla matice sousednosti grafu, zadaný počet vrcholů grafu, který se rovnal vstupní množině použitelných barev, a počáteční hodnota parametru b. Výstupem algoritmu byla Δ-*chromatická sekvence*, čas výpočtů, počet iterací a počet řešených uzlů pro každý člen sekvence a výsledné množiny barevných tříd.

Matematický model byl řešen exaktní metodou a metodou použití modelu s 3 časovými omezeními. Pokud v tomto časovém limitu nebylo nalezeno optimální řešení, zaznamenalo se nejlepší přípustné řešení a algoritmus se přesunul k výpočtu následujícího členu sekvence. Jednotlivé heuristické algoritmy se pak lišily pouze délkou časového limitu.

Všechny algoritmy byly následně testovány na několika skupinách grafů různých velikostí, struktur a hustot hran. Skupiny testovaných grafů byly zvoleny velmi podobně jako v práci Isabelle Niewoudt [9], a to z důvodu možnosti následného porovnání algoritmů. Exaktní algoritmus nalezl optimální řešení pro celkem 11 grafů ze skupiny 16 kompletních multipartitních grafů a benchmark grafů. Tyto skupiny obsahovaly co do řešení nejtěžší a největší grafy ze všech testovaných skupin grafů. Grafy z ostatních skupin byly vyřešeny v řádu sekund, mnohdy i méně. Získání optimálního Δ-*chromatické sekvence* se ukázalo jako nesnadné a výpočetně velmi náročné. To však bylo vzhledem ke stanovení náročnosti úlohy pro řešení algoritmem očekáváno. Hlavním důvodem je obrovský počet přípustných řešení, která tento algoritmus musí projít. U exaktního algoritmu byl zjištěn vliv velikosti grafu a hustoty hran na obtížnost zabarvení grafu. Jako nejobtížnější se ukázal výpočet defektivního zabarvení pro hodnoty parametru $b = \{1,2\}$. Právě s rostoucím parametrem *b* roste i počet přípustných řešení a úloha expanduje. Z časových důvodů byl čas výpočtu Δ-*chromatické sekvence* omezen na 12 hodin. Pokud

výpočet tento časový limit překročil, byly zaznamenány pouze vypočtené optimální členy sekvence. Po bližší analýze výsledků algoritmů bylo zjištěno, že grafy bez zjištěné optimální Δ-*chromatické sekvence* se vyznačují vyšším počtem vrcholů, či vyšší hustotou hran než grafy, pro které optimální Δ-*chromatická sekvence* zjištěna byla. Takovýto výsledek byl očekáván. Avšak v případě kompletních multipartitních grafů byl rozdíl ve výpočetních časech velmi výrazný i mezi grafy se stejným počtem vrcholů a podobnou hustotou hran. Lze předpokládat, že struktura grafu má na výpočet také nezanedbatelný vliv a může významně ovlivnit rychlost získání řešení.

Heuristické algoritmy vytvořené na základě matematického modelu se mezi sebou lišily pouze v délce časového intervalu pro výpočet každého členu Δ-*chromatické sekvence*. Algoritmy byly testovány na stejných skupinách grafů jako exaktní algoritmus, takže mohla být určena jejich přesnost a případy, kdy se nejlepší přípustné řešení nalezené heuristickým algoritmem shodovalo s řešením optimálním. Jelikož byly optimální Δ-*chromatické sekvence* známy pouze pro 11 benchmark a kompletních multipartitních grafů, mohla být určována přesnost heuristických algoritmů pouze u těchto grafů. Z těchto 11 grafů bylo pro 9 z nich získáno řešení shodné s optimálním alespoň jedním z heuristických algoritmů. Zbylé dva členy se pak odlišovaly pouze v případě jednoho členu Δ-*chromatické sekvence*. Takovéto výsledky lze hodnotit jako velmi zdařilé, obzvlášť vzhledem k jejich výpočetním časům. Pokud heuristický algoritmus nedosáhl optimálního řešení, pak jeho řešení nebylo v žádném z případů výrazně odlišné od řešení optimálního. V žádném z případů se žádný člen Δ-*chromatické sekvence* nelišil od členu optimálního o více než 1.

Jednou ze skupin grafů, na kterých byly testovány všechny algoritmy představené v této práci, byly grafy náhodně generované. Na těchto skupinách grafů byl v případě heuristický algoritmů pozorován zajímavý jev vývoje počtu iterací v závislosti na velikosti grafu a hustotě jeho hran. Z výsledků lze vyvodit závěr, že každý z heuristických algoritmů je vhodnější k řešení grafů s jinými parametry. Z pozorování vývoje počtu iterací v rámci řešení různě velkých grafů lze pozorovat, že od určité náročnosti úlohy barvení grafu začínají jednotlivé heuristiky selhávat, a počet řešených iterací se snižuje. Lze předpokládat, že grafy, u kterých je počet iterací maximální, jsou grafy hraniční velikosti co do efektivnosti heuristiky. Grafy větší a s vyšší hustotou hran jsou pro dané heuristiky

příliš komplexní a algoritmus není schopen dostatečně rychlých výpočtů iterací, což vede ke snižování přesnosti algoritmu.

V poslední části práce byly algoritmy založené na matematickém modelu srovnávány s algoritmy z dizertační práce Isabelle Nieuwoudt [9]. Porovnávány byly výpočetní časy jednotlivých členů Δ-*chromatických sekvencí* a v případě heuristických algoritmů i optimalita výsledků. Ze srovnání vyplývá, že jak heuristické algoritmy, tak algoritmus exaktní jsou co do efektivnosti a časové náročnosti více než konkurenceschopné. Srovnání výsledků rychlosti algoritmů dopadlo velmi podobně, každý algoritmus vykazoval lepší výsledky u jiných grafů. Jeden z algoritmů z [9] se ukázal jako velmi rychlý, avšak také velmi nepřesný. V případě exaktních algoritmů dopadl co do rychlosti výpočtu lépe algoritmus založený na matematickém modelu. Co do přesnosti heuristik se ukázaly přesnější algoritmy z této práce. U žádného z grafů nebylo dosaženo lepšího zabarvení použitím některého z algoritmů z [9].

Možností dalšího postupu je mnoho. Jednou z nich je zaměření se na efektivitu heuristik a bližší prozkoumání této problematiky. Při detailní analýze by mohla být stanovena obecná hranice efektivity heuristického algoritmu v závislosti na časovém limitu a poté, v závislosti na velikosti grafu, stanoven optimální časový limit pro výpočet jednotlivých členů Δ-*chromatické sekvence*. Další cestou je samozřejmě implementace modelu na reálný problém. Nespornou výhodou tohoto matematického modelu je jeho obecnost. Při konstrukci dalších omezujících podmínek, vhodných k řešení konkrétní úlohy, je možné tímto algoritmem řešit velké množství úloh. Upravený matematický model byl například použit k řešení náročného hlavolamu, ve kterém bylo třeba uspořádat 19 šestiúhelníků do určitého obrazce tak, aby součet každé diagonály či řádku byl roven předem dané hodnotě. Jednoduchou úpravou modelu z této práce a přidáním doplňujících omezujících podmínek byl tento hlavolam během krátké doby vyřešen. Dalším přístupem by pak mohlo být rozšíření modelu o další omezující podmínky, které by efektivně snížili množinu přípustných řešení, a tím celý výpočet urychlily.

LITERATURA

1. **VOLEK, Josef a Bohdan LINDA.** *Teorie grafů - aplikace v dopravě a veřejné správě.* Pardubice : Univerzita Pardubice, 2012. 978-80-7395-225-9.

2. **Weisstein, Eric W.** Four-Color Theorem. *MathWorld.* [Online] http://mathworld.wolfram.com/Four-ColorTheorem.html.

3. **Krantz, Steven G.** The Four-Color Problem: Concept and Solution. *Washington University St. Louis - Department of Mathematics.* [Online] Washington University St. Louis.

4. *On a generalization of chromatic number.* **JA Andrews, MS Jacobson.** 48, 1985, Sv. Congressum Numeraticum .

5. *Defective colourings of graphs in surfaces: Partitions into subgraphs of bounded.* **L Cowen, R Cowen, D Woodall.** 10, místo neznámé : Journal of Graph Theory, 1986.

6. *Conditional colourability in graphs.* **Harary F, Jones K.** místo neznámé : Graphs and applications, 1985.

7. **Frick M, MA Henning.** Extremal results on defective colourings of graphs. *Discrete Mathematics.* 1994, 126.

8. **M, Frick.** *Generalised colourings of graphs.* Johannesburg : Rand Afrikaans University, 1986.

9. **Niewoudt, Isabelle.** *On The Maximum Degree Chromatic Number of a Graph.* Stellenbosch : autor neznámý, 2007.

10. **Malaguti E, Toth P.** A survey on vertex coloring problems. *INTERNATIONAL TRANSACTIONS IN OPERATIONAL RESEARCH.* 2010, 17.

11. **R, Brown.** Chromatic scheduling and the chromatic number problem. *Management Science.* 4, 1982, 19.

12. **Hansen, P., Labbe´ , M., Schindl, D.** *Set covering and packing formulations of graph coloring: algorithms and first polyhedral results.* Montréal : Universite´ de Montre´ al, 2005.

13. **Méndez-Díaz, I., Zabala, P.** A cutting plane algorithm for graph coloring. *Discrete Applied Mathematics.* 2008.

14. **D, Brélaz.** New methods to color the vertices of a graph. *Communications of the ACM.* 4, 1979, 22.

15. **Hertz, A., de Werra, D.** Using tabu search techniques for graph coloring. *Computing.* 1987, 39.

16. **Johnson, D.S., Aragon, C.R., McGeoch, L.A., Schevon, C.** Optimization by simulated annealing: An experimental evaluation; Part II, graph coloring and number partitioning. *Operational Research.* 1991.

17. **Davis, L.** *Handbook of Genetic Algorithms.* New York : Van Nostrand Reinhold, 1998.

18. **Galinier, P., Hao, J.K.** Hybrid evolutionary algorithms for graph coloring. *Journal of Combinatorial.* 1999, Sv. 3, 4.

19. **Bentz C, Costa MC, de Werra D, Picouleau C & Ries B.** On a graph coloring problem arising from discrete. *Networks.* 2008, strana 256-267.

20. *Quasi-optimal bandwidth allocation for multi-spot.* **Alouf S, Altman E, Galtier J, Lalande J-F & Touati C.** Miami (Florida) : autor neznámý, 2005. 1.

21. **READ, Ronald C. a Robin J. WILSON.** *An Atlas of Graphs.* Oxford : Clarendon Pres, 1998. 0-19-853289-X.

22. **F., Harary.** *Graph Theory.* USA : Addison-Wesley Publishing Company Inc., 1972. 69-19989.

23. **Weisstein, Eric W.** Quartic Graph. *MathWorld--A Wolfram Web Resource.* [Online] http://mathworld.wolfram.com/QuarticGraph.html.

24. **Johnson, D. S., & Trick, M. A.** *DIMACS series in discrete mathematics and theoretical computer science.* [Online] 2016. http://dimacs.rutgers.edu/Volumes/Vol26.html.

25. **Weisstein, Eric W.** Queen Graph. *MathWorld--A Wolfram Web Resource.* [Online] http://mathworld.wolfram.com/QueenGraph.html.

26. **Demlová, Marie.** Teorie algoritmů. *Třída P a třída NP.*

27. **třídy, Složitostní.** is.mendelu.cz. *https://is.mendelu.cz/eknihovna/opory/zobraz_cast.pl?cast=7709.* [Online]

28. **Karp, RM.** *Reducibility among combinatorial problems.* New York : Plenum Press, 1972.

29. **JI, Brown.** The complexity of generalized graphs. *Discrete Applied Mathematics.* 1966.

30. **L Cowen, W Godddard, CE Jesurum.** Defective colouring revisited. *Journal of Graph Theory.* 1997.

31. **K, Poh.** On the linear vertex-arboricity of a planar. *Graph Theory.* 1990.

32. **W, Goddard.** *Acyclic colorings of planar graphs.* 1991.

33. **E.C., Sewell.** An improved algorithm for exact graph coloring. *American Mathematical Society.* 1996.

34. **Blöchliger I, Zefferey N.** A reactive tabu search using partial solutions for the graph coloring problem. *Computers and Operations Research.* 2008, Sv. 3, 35.

35. **Fleurent, C., Ferland, J.A.** Object-oriented implementation of heuristic search methods for graph coloring. IN Johnson, D.S., Trick, M.A. . *DIMACS Series in Discrete Mathematics and Theoretical Computer Science, American Mathematical Society.* 1996.

36. **Morgenstern, C.A.** Distributed coloration neighborhood search. In Johnson, D.S., Trick, M.A. . *DIMACS Series in Discrete.* American Mathematical Society, 1996.

37. **Weisstein, Eric W.** Cycle Graph. *MathWorld--A Wolfram Web Resource.* [Online] http://mathworld.wolfram.com/CycleGraph.html.

38. —. Mycielski Graph. *MathWorld--A Wolfram Web Resource.* [Online] http://mathworld.wolfram.com/MycielskiGraph.html.

39. *istudentzone.* [Online] 2017. http://www.istudentzone.cz/en/news/pristi-zastavka-praha-joys-prague-public-transport/.

SEZNAM OBRÁZKŮ

Obrázek 1 - Graf J – příklad klik v grafu. (Zdroj: autor)
Obrázek 2 - Použití Brélazova algoritmu (Zdroj: autor)
Obrázek 3 – Graf zabarvený Brélazovým algoritmem (Zdroj: autor)
Obrázek 4 Graf reprezentujícíc počítačovou síť (Zdroj: autor)
Obrázek 5 – Schéma počítačové sítě (Zdroj: autor)
Obrázek 6 - Schéma linek metra v Praze (Zdroj: [35])
Obrázek 7 Graf stanic metra (Zdroj: autor)
Obrázek 8 Výsledné zabarvení grafu metodou MDCP (Zdroj: autor)
Obrázek 9 - Kubické grafy řádu 8,10 a12 (Zdroj: autor)
Obrázek 10 - Kvartické grafy řádu 8,10 a 12 (Zdroj: autor)
Obrázek 11 - Příklady kružnicových grafů Zdroj: [37]
Obrázek 12 - Kompletní multipartitní graf (zdroj: [40])
Obrázek 13 - Queen grafy (Zdroj: [25])
Obrázek 14 - Mycielski grafy (Zdroj: [38])
Obrázek 15 - Přehled odchylek počtů barev v sekvencích, (Zdroj: autor)
Obrázek 16 – Grafy znázorňující vývoj časů výpočtů vzhledem k řádu grafu a hustotě hran (Zdroj: autor)
Obrázek 17 – Grafy znázorňující vývoj iterací výpočtů vzhledem k řádu grafu a hustotě hran (Zdroj: autor)
Obrázek 18 – Porovnání chromatických sekvencí grafu *queen*6_6 vypočtených heuristickými algoritmy, modelem s různým časovým omezením a exaktním přístupem (Zdroj: autor)

SEZNAM TABULEK

Tabulka 1 Přiřazení stanic k vrcholům grafu

Tabulka 2 Výsledky modelů s časovým omezením

Tabulka 3 Přehled grafů bez optimálního řešení

Tabulka 4 - Přehled výsledků exatního přístupu

Tabulka 5 - Porovnání časů výpočtů grafů podobných parametrů

Tabulka 6 Srovnání výsledků a časů výpočtů exaktních algoritmů

Tabulka 7 - Srovnání výsledků heuristických algoritmů a modelu s časovým omezením