



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**  
BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**  
**ÚSTAV INTELIGENTNÍCH SYSTÉMŮ**  
FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF INTELLIGENT SYSTEMS

# **METODY PRO OBARVENÍ UZLŮ MULTIGRAFU**

METHODS FOR COLORING NODES IN MULTIGRAF

**BAKALÁŘSKÁ PRÁCE**  
BACHELOR'S THESIS

**AUTOR PRÁCE**  
AUTHOR

**MARTIN KNOTEK**

**VEDOUcí PRÁCE**  
SUPERVISOR

**Ing. VÁCLAV ŠÁTEK, Ph.D.**

BRNO 2016

**Vysoké učení technické v Brně - Fakulta informačních technologií**

Ústav inteligentních systémů

Akademický rok 2015/2016

**Zadání bakalářské práce**

Řešitel: **Knotek Martin**

Obor: Informační technologie

Téma: **Metody pro obarvení uzlů multigrafu**  
**Methods for Coloring Nodes in Multigraf**

Kategorie: Umělá inteligence

**Pokyny:**

1. Prostudujte teorie grafů se zaměřením na barvení grafů a multigrafy.
2. Vyberte vhodné metody pro obarvení orientovaných multigrafů na základě zadaných podmínek a omezení.
3. Implementujte metody a otestujte na zvolené sadě multigrafů.
4. Porovnejte obarvení grafu na základě definovaného barevného schématu při minimalizaci počtu použitých barev. Pro implementaci použijte softwarové prostředí Octave nebo Matlab.
5. Zhodnoťte dosažené výsledky.

**Literatura:**

- Jiří Demel, Grafy, SNTL Praha, 1988.
- Jiří Demel, Grafy a jejich aplikace, Academia, 2002.
- Kolář, Josef. Teoretická informatika. Praha, 2004.
- James A. McHugh, Algorithmic Graph Theory, 1990.
- Reinhard Diestel, Graph Theory, Third Edition, Springer-Verlag, Heidelberg, 2000.

Pro udělení zápočtu za první semestr je požadováno:

- První dva body zadání.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese

<http://www.fit.vutbr.cz/info/szz/>

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Šátek Václav, Ing., Ph.D.**, UITS FIT VUT

Konzultant: Paar Martin, Ing., FEKT VUT

Datum zadání: 1. listopadu 2015

Datum odevzdání: 18. května 2016

**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**  
Fakulta informačních technologií  
Ústav inteligentních systémů  
612 66 Brno, Božetěchova 2

doc. Dr. Ing. Petr Hanáček  
vedoucí ústavu

## Abstrakt

Tato práce se zabývá algoritmy pro vrcholové barvení a jejich aplikaci na barvení prvků elektrické rozvodné sítě nízkého napětí. Výsledkem práce je program zobrazující průběh barvení pěti implementovaných metod na zvoleném grafu, který představuje distribuční síť elektrické energie ve městě.

## Abstract

This thesis deals with algorithms for vertices coloring and their application to the coloring elements of electrical low voltage grid. Result of this thesis is a program, which displays progress of coloring five implemented algorithms on a selected graph. The graph represents electricity distribution network in a city.

## Klíčová slova

Barvení grafu, vrcholové barvení, graf, multigraf, orientovaný graf, chromatické číslo, elektrická rozvodná síť

## Keywords

Graph coloring, vertex coloring, graph, multigraph, directed graph, chromatic number, power grid

## Citace

KNOTEK, Martin. *Metody pro obarvení uzlů multigrafu*. Brno, 2016. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Šátek Václav.

# Metody pro obarvení uzlů multigrafu

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Václava Šátka, Ph.D. Další informace mi poskytl pan Ing. Martin Paar, Ph.D. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....  
Martin Knotek  
17. května 2016

## Poděkování

Rád bych poděkovat vedoucímu práce Ing. Václavu Šátkovi, Ph.D. za vedení a připomínky při psaní bakalářské práce. Dále bych chtěl poděkovat Ing. Martinu Paarovi, Ph.D. za poskytnuté odborné rady a informace z oblasti energetiky a návrhy na zlepšení programu.

© Martin Knotek, 2016.

*Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.*

# Obsah

<b>1</b>	<b>Úvod</b>	<b>4</b>
<b>2</b>	<b>Teorie grafů</b>	<b>5</b>
2.1	Neorientovaný graf a jeho množiny	5
2.2	Orientovaný graf	5
2.2.1	Množiny hran a vrcholů v orientovaných grafech	6
2.3	Grafové pojmy	6
2.3.1	Vlastnosti hran	6
2.3.2	Typy grafů	7
2.3.3	Další důležité pojmy	7
2.4	Reprezentace grafu	9
2.4.1	Seznamy vrcholů a hran	9
2.4.2	Seznam vrcholů a seznamy okolí vrcholů	9
2.4.3	Matice sousednosti	9
2.4.4	Matice incidence	9
2.4.5	Matice dostupnosti	10
2.4.6	Prüferovo číslo	10
<b>3</b>	<b>Barvení grafu</b>	<b>11</b>
3.1	Chromatické číslo	11
3.2	Praktické využití	12
3.2.1	Barvení mapy	12
3.2.2	Plánování rozvrhu	12
3.2.3	Předepisování léků	12
3.2.4	Skladování nebezpečných látek	12
3.3	Časová a paměťová složitost	13
<b>4</b>	<b>Algoritmy pro vrcholové obarvení grafu</b>	<b>15</b>
4.1	Sekvenční (hladové) barvení grafu	15
4.1.1	Heuristika First Fit (FF)	16
4.1.2	Heuristika Largest Degree Ordering (LDO)	16
4.1.3	Heuristika Saturation Degree Ordering (SDO)	16
4.1.4	Heuristika Incidence Degree Ordering (IDO)	17
4.1.5	Heuristika kombinující LDO a IDO	18
4.1.6	Heuristika kombinující SDO a LDO	18
4.2	Barvení grafu pomocí nezávislých množin	19
4.2.1	Algoritmus pro hledání maximální nezávislé množiny	19
4.2.2	Pravděpodobnostní alg. pro hledání maximální nezávislé množiny	20

4.3	Barvení grafu slepováním uzlů . . . . .	20
4.3.1	Heuristika Dutton & Brigham . . . . .	20
4.3.2	Heuristika Recursive Largest First (RLF) . . . . .	22
4.4	Výměna barev . . . . .	23
<b>5</b>	<b>Algoritmy pro zjišťování barevnosti</b>	<b>24</b>
5.1	Barvení hrubou silou . . . . .	24
5.2	Backtracking . . . . .	24
<b>6</b>	<b>Získání podkladních dat</b>	<b>26</b>
6.1	Rozložení prvků v el. rozvodné síti . . . . .	26
6.2	Vytvoření grafu v programu <i>yEd</i> . . . . .	26
<b>7</b>	<b>Implementace v prostředí <i>MATLAB</i></b>	<b>28</b>
7.1	Reprezentace dat v systému <i>MATLAB</i> . . . . .	28
7.1.1	Vrcholy . . . . .	28
7.1.2	Napájecí místa – trafostanice . . . . .	28
7.1.3	Odběrová místa . . . . .	29
7.1.4	Hrany . . . . .	29
7.1.5	Matice skupin . . . . .	29
7.1.6	Matice sousednosti . . . . .	29
7.1.7	Barvy . . . . .	30
7.2	Získání dat z <i>XML</i> souboru . . . . .	30
7.3	Výpočet odběru . . . . .	30
7.4	Výpočet proudů . . . . .	31
7.4.1	Modifikovaná metoda uzlových napětí . . . . .	31
7.5	Určení sousednosti napájecích vrcholů . . . . .	32
7.6	Implementované algoritmy . . . . .	33
7.7	Problémy při implementaci a jejich řešení . . . . .	33
7.7.1	Matlab neobsahuje typ množina . . . . .	33
7.7.2	Rozměry vybraných matic . . . . .	33
7.7.3	Vytvoření barevného schématu . . . . .	33
7.7.4	Slévání proudů . . . . .	34
7.7.5	Činný a jalový proud . . . . .	34
7.7.6	Výpočetní náročnost backtrackingu . . . . .	34
<b>8</b>	<b>Zhodnocení barvicích algoritmů</b>	<b>35</b>
8.1	Vyhodnocení . . . . .	40
<b>9</b>	<b>Závěr</b>	<b>41</b>
	<b>Literatura</b>	<b>42</b>
	<b>Přílohy</b>	<b>43</b>
	Seznam příloh . . . . .	44

<b>A</b>	<b>Uživatelská příručka</b>	<b>45</b>
A.1	Terminálové rozhraní . . . . .	45
A.2	Grafické uživatelské rozhraní . . . . .	46
A.2.1	Získání/načtení/uložení dat . . . . .	46
A.2.2	Nastavení průběhu výpočtu . . . . .	46
A.2.3	Nastavení grafického zobrazení . . . . .	47
A.2.4	Metoda barvení . . . . .	47
A.2.5	Metoda vylepšení obarvení . . . . .	47
A.2.6	Menu . . . . .	48
<b>B</b>	<b>Jednoduchý příklad grafu</b>	<b>49</b>

# Kapitola 1

## Úvod

Tato bakalářská práce se zabývá grafickým znázorněním proudů tekoucích v elektrických distribučních sítích nízkého napětí. Vznikla na základě požadavku na vyhodnocování toku (barevné odlišení proudů) v zauzlených distribučních sítích – mřížových sítích. Ty jsou v dnešní době používány k distribuci el. energie z transformátorů nízkého napětí pro domácnosti a podniky v městech a městských částech jen vzácně. Vyznačují se vysokou spolehlivostí, ale jejich provoz je drahý.

První kapitola je věnována teorii grafů. Je zde popsáno, co to vlastně grafy jsou, k čemu slouží a jakým způsobem se dají reprezentovat. Také jsou zde definovány pojmy použité v celé této práci.

Další část je zasvěcena obarvování grafů. V ní je mimo jiné vysvětlen pojem *chromatické číslo grafu* a rozdíl mezi hranovým a vrcholovým barvením. Obsahuje též několik příkladů využití barvení grafů k řešení praktických problémů. Dále se zmiňuje o paměťové a časové složitosti, díky níž lze algoritmy snadno porovnávat z hlediska paměťové a časové náročnosti.

Obsáhlá kapitola se zabývá samotnými algoritmy pro vrcholové barvení. Obsahuje popis různých typů a jednotlivých variací algoritmů. Pro názornost je průběh barvení algoritmů znázorněn na obrázcích.

Následující část se zabývá algoritmy, které umí nalézt chromatické číslo grafu.

Data určující rozložení prvků v reálné elektrické rozvodné síti nízkého napětí nejsou snadno dostupná. Zde je vysvětleno jakým způsobem se dají získat či alespoň odhadnout tyto informace.

Kapitola 7 je věnována samotné implementaci programu. Objasňuje, jakým způsobem jsou získána, zpracována a uchována data a které algoritmy byly v projektu implementovány. Také popisuje zajímavé problémy, které nastaly při programování, a jejich řešení.

Další kapitola hodnotí důležité vlastnosti (počet barev a čas) implementovaných metod. Testy obarvení se provádějí na grafech představujících modely měst o různé velikosti v rozsahu od 2 000 až po 80 000 obyvatel.

V závěru práce je zhodnocení celého projektu a několik návrhů na jeho vylepšení.



## Kapitola 2

# Teorie grafů

Graf je složen z tzv. vrcholů (též nazývanými uzly) a hran. Pomocí grafu lze vyjádřit (zobrazit) vztahy mezi daty a tím usnadnit práci s danými daty. Definice grafu je uvedena v kapitole 2.2 a 2.1.

Využití grafů v reálném životě je velmi rozsáhlé. Značnou část úkolů z různých oblastí (např. elektrotechnika, informační technologie, chemie, infrastruktura) lze převést na graf. Spousta lidí ani neví o tom, že podvědomě převádí značnou část praktických problémů na grafy za účelem zjednodušení řešení.

Praktickým využitím obarvování grafů se zabývá samostatná kapitola 3.2.

Teorie v této kapitole vychází z [2] a [5].

### 2.1 Neorientovaný graf a jeho množiny

Neorientovaný graf je trojice  $G = (V, E, \epsilon)$ , kde:

$V$  je neprázdná konečná množina obsahující vrcholy ( $v$ )

$E$  je konečná množina obsahující neorientované hrany ( $e$ )

$\epsilon$  je zobrazením  $\epsilon : E \rightarrow V^2$

Zobrazení přiřazuje každé hraně  $e$  jednoprvkovou nebo dvouprvkovou množinu vrcholů. Tyto vrcholy se nazývají krajními vrcholy hrany  $e$ .

Neorientovaný graf má definovány některé množiny hran a vrcholů uvedených v kapitole 2.2.1, konkrétně jde o  $V_G(x)$ ,  $V_G(A)$ ,  $E_G(x)$ ,  $W_G(A)$ ,  $m_G(x, y)$  a  $d(x)$  a jim odpovídající pojmy.

### 2.2 Orientovaný graf

Orientovaný graf je trojice  $G = (V, E, \epsilon)$ , kde:

$V$  je neprázdná konečná množina obsahující vrcholy ( $v$ ) (z anglického *vertices*)

$E$  je konečná množina obsahující orientované hrany ( $e$ ) (z anglického *edges*)

$\epsilon$  je zobrazením  $\epsilon : E \rightarrow V^2$

Toto zobrazení přiřazuje každé hraně  $e$  **uspořádanou** dvojici vrcholů  $(x, y)$ . Hrana  $e$  vede z počátečního vrcholu  $x$  ( $Pv(e)$ ) do koncového vrcholu  $y$  ( $Kv(e)$ ).

### 2.2.1 Množiny hran a vrcholů v orientovaných grafech

Nechť  $G = (V, E, \epsilon)$  je orientovaný graf,  $x$  a  $y$  jsou jeho libovolné vrcholy a  $A \subseteq V$  je libovolná podmnožina jeho vrcholů. Pak:

$V_G^+(x)$	$= \{z \in V \mid (x, z) \in \epsilon(E)\}$	– množina následníků vrcholu $x$
$V_G^-(x)$	$= \{z \in V \mid (z, x) \in \epsilon(E)\}$	– množina předchůdců vrcholu $x$
$V_G(x)$	$= V_G^+(x) \cup V_G^-(x)$	– množina sousedů vrcholu $x$
$V_G(A)$	$= \bigcup_{x \in A} V_G(x)$	
$E_G^+(x)$	$= \{e \in E \mid P_v(e) = x\}$	– výstupní okolí vrcholu $x$
$E_G^-(x)$	$= \{e \in E \mid K_v(e) = x\}$	– vstupní okolí vrcholu $x$
$E_G(x)$	$= E_G^+(x) \cup E_G^-(x)$	– okolí vrcholu $x$
$W_G^+(A)$	$= \{e \in E \mid P_v(e) \in A \ \& \ K_v(e) \notin A\}$	
$W_G^-(A)$	$= \{e \in E \mid P_v(e) \notin A \ \& \ K_v(e) \in A\}$	
$W_G(A)$	$= W_G^+(A) \cup W_G^-(A)$	– řez určený množinou vrcholů $A$
$m_G^+(x, y)$	$=  E_G^+(x) \cap E_G^-(y) $	– násobnost hrany
$m_G(x, y)$	$=  E_G(x) \cap E_G(y) $	– násobnost hrany
$d_G^+(x)$	$=  E_G^+(x) $	– výstupní stupeň vrcholu $x$
$d_G^-(x)$	$=  E_G^-(x) $	– vstupní stupeň vrcholu $x$
$d_G(x)$	$= d_G^+(x) + d_G^-(x)$	– stupeň vrcholu $x$

## 2.3 Grafové pojmy

V literatuře, zabývající se problematikou grafů (např. [2],[3],[8]), jsou definice a terminologie nejednotné. Proto uvedme pojmy, které budou následně využívány v práci.

### 2.3.1 Vlastnosti hran

**Incidentní vrcholy** Vrcholy  $x, y$  jsou **incidentní** (nebo-li incidují) s hranou  $e$  a zároveň hrana  $e$  inciduje s vrcholy  $x, y$ , když

$$\text{pro } x, y \in V, \exists e \in E, \epsilon : e \rightarrow (x, y) \vee e \rightarrow (y, x). \quad (2.1)$$

**Orientované hrany** mají *počáteční* a *koncový vrchol* a vždy vedou z počátečního do koncového vrcholu.

**Neorientované hrany** jsou chápány jako symetrické spojení dvou vrcholů. U těchto hran nezáleží na jejich směru (vedou jak z počátečního do koncového vrcholu, tak i naopak).

**Smyčka** je hrana začínající i končící ve stejném vrcholu.

**Násobnost hrany** je počet hran vedoucích mezi dvěma stejnými vrcholy. U *orientovaného grafu* musí mít hrany navíc stejný *počáteční* i *koncový vrchol*.

**Ohodnocení hrany** vyjadřuje kvalitu či kvantitu vztahu mezi dvěma vrcholy. Každá hrana může být ohodnocena i z více hledisek nebo nemusí být ohodnocena vůbec.

**Rovnoběžné hrany** jsou různé hrany, které spojují stejné (jednoprvkové nebo dvouprvkové) množiny vrcholů.

### 2.3.2 Typy grafů

Grafy se mohou dělit podle typu hran, ze kterých jsou složeny, na:

**Neorientované grafy** jsou složeny pouze z *neorientovaných hran*.

**Orientované grafy** se skládají jen z *orientovaných hran*.

**Smíšené grafy** obsahují oba typy hran (jak *orientované* tak i *neorientované*).

Nebo podle jejich násobnosti na:

**Prosté grafy** mají *násobnost* každé hrany nejvýše jedna.

**Multigrafy** oproti prostým grafům mohou obsahovat i hrany s *násobností* větší jak jedna.

Dále je zde uvedeno několik speciálních grafů:

**Úplný graf** je *neorientovaný graf*, mezi jehož každými dvěma vrcholy existuje právě jedna *neorientovaná hrana*. V *orientovaném grafu* vedou mezi vrcholy namísto jedné *neorientované* právě dvě navzájem opačně *orientované hrany*.

**Bipartitní graf** je takový graf, jehož množinu vrcholů lze rozdělit na dvě množiny tak, že z každého vrcholu první množiny může vést hrana pouze do vrcholů druhé a naopak. Jinými slovy, žádné dva vrcholy ze stejné množiny nesmějí být spojeny hranou.

**Úplný bipartitní graf** je *bipartitní graf*, u kterého vedou hrany z každého vrcholu jedné množiny do každého vrcholu druhé množiny.

**Diskrétní graf** obsahuje alespoň jeden vrchol, ale neobsahuje žádné hrany.

**Prázdný graf** neobsahuje žádné uzly ani hrany.

**Nekonečný graf** má nekonečně mnoho vrcholů.

**Rovinný graf** (nazývaný též **planární**) je takový graf, který lze nakreslit v rovině tak, aby se jeho žádné dvě hrany neprotály – neměli žádné společné body krom krajních vrcholů (počáteční a koncový vrchol).

### 2.3.3 Další důležité pojmy

**Sousední vrcholy** jsou dva vrcholy propojené hranou. O těchto vrcholech se říká, že spolu **sousedí**.

**Orientovaný sled** je posloupnost vrcholů a hran  $v_0, e_1, v_1, e_2, v_2, \dots, e_k, v_k$ , kde pro každou hranu  $e_i$  z této posloupnosti platí  $Pv(e_i) = v_{i-1}$  a  $Kv(e_i) = v_i$ .

**Neorientovaný sled** je posloupnost vrcholů a hran  $v_0, e_1, v_1, e_2, v_2, \dots, e_k, v_k$ , kde každá hrana  $e_i$  z této posloupnosti spojuje vrcholy  $v_{i-1}$  a  $v_i$ .

**Orientovaný/neorientovaný tah** je *orientovaný/neorientovaný sled* v němž se neopakuje žádná hrana.

**Orientovaná/neorientovaná cesta** je *orientovaný/neorientovaný sled* v němž se neopakuje žádný vrchol.

**Kružnice** je *neorientovaný sled* začínající i končící ve stejném vrcholu. Navíc se v něm nesmí opakovat žádný vrchol ani žádná hrana.

**Cyklus** je *orientovaný sled* začínající i končící ve stejném vrcholu. Navíc se v něm nesmí opakovat žádný vrchol ani žádná hrana.

**Dostupnost** Vrchol  $y$  je **orientovaně/neorientovaně dostupný** z vrcholu  $x$ , jestliže existuje *orientovaný/neorientovaný sled* vedoucí z vrcholu  $x$  do vrcholu  $y$ .

**Podgraf** grafu  $G$  je graf  $H$  vzniklý odebráním některých (i žádných) vrcholů a hran z grafu  $G$ .

**Souvislost grafu** Graf je souvislý právě tehdy, když pro každé dva vrcholy  $x, y$  existuje *cesta*, která je spojuje.

U *orientovaných grafů* se navíc rozlišuje mezi tzv. slabou a silnou souvislostí. **Silná souvislost** znamená, že v grafu existuje *cesta* z  $x$  do  $y$  a zároveň z  $y$  do  $x$ . Pokud existuje pouze jedna z těchto dvou *cest*, jedná se o **slabou souvislost**.

**Les** je graf neobsahující *kružnici*.

**Strom** je graf neobsahující *kružnici* (tedy *les*), který je *souvislý*.

**Nezávislá množina grafu** je podmnožina vrcholů grafu, kde žádné dva vrcholy nejsou spojeny hranou (nebo-li nejsou *sousedy*). Graf může mít i více nezávislých množin.

**Maximální nezávislá množina grafu** je *nezávislá množina grafu*, do níž se nedá přidat další vrchol, aniž by zůstala *nezávislou množinou*. Graf může mít i více maximálních nezávislých množin.

**Nejpočetnější nezávislá množina grafu** (též největší nezávislá množina) je *maximální nezávislá množina grafu* s největším počtem vrcholů. Graf může mít i více nejpočetnějších nezávislých množin.

**Nezávislost grafu** je počet vrcholů v *nejpočetnější nezávislé množině grafu*. Značí se  $\alpha(G)$ .

**Klika** grafu  $G$  je maximální podgraf grafu  $G$ , který je *úplným grafem*.

## 2.4 Re prezentace grafu

V této sekci je popsáno několik nejčastěji používaných možností reprezentace grafů.

### 2.4.1 Seznamy vrcholů a hran

Vrcholy jsou popsány výčtem (seznamem) prvků a hrany jsou popsány seznamem trojic. Každá trojice obsahuje jméno hrany a její počáteční a koncový vrchol. Když nezáleží na jménech hran, lze je vypustit a popisovat je pouze dvojicemi vrcholů. V opačném případě, když je potřeba znát více údajů (např. ohodnocení hrany či vrcholu), lze rozšířit trojici na  $n$ -tici přidáním požadovaných informací. Neorientované grafy se popisují stejně jako orientované, přičemž pořadí vrcholů u každé hrany se volí libovolně.

V praxi je tento způsob reprezentace hojně využíván hlavně z důvodů univerzálnosti (možnosti rozšíření o další informace) a relativní úspornosti oproti matici sousednosti popsané v kapitole 2.4.3 a matici incidence popsané v kapitole 2.4.4.

### 2.4.2 Seznam vrcholů a seznamy okolí vrcholů

Jedná se o úspornější variantu předchozího způsobu (kapitola 2.4.1). Množina vrcholů je opět popsána seznamem prvků. Hrany jsou ovšem popisovány po skupinách, kde pro každý vrchol  $x$  je uveden seznam množiny hran  $E^+(x)$  (nebo  $E^-(x)$ ). Jelikož počáteční vrchol  $x$  je společný vždy pro celou skupinu, tak každou hranu stačí popsat pouze svým jménem a koncovým vrcholem.

Pro popis neorientovaného grafu lze využít jeho některé orientace nebo uvádět seznamy množin  $E(x)$  (každá hrana uvedena ve dvou seznamech – méně úsporné).

### 2.4.3 Matice sousednosti

Pro orientovaný graf  $G$  zvolíme (libovolně, ale pevně) pořadí jeho vrcholů, pak můžeme přiřadit grafu  $G$  čtvercovou matici sousednosti  $M_G^+$  řádu  $n$  (kde  $n$  je počet vrcholů) předpisem

$$m_{ij}^+ = m^+(v_i, v_j).$$

Pro neorientované grafy definujeme matici sousednosti  $M_G$  předpisem

$$m_{ij} = m(v_i, v_j).$$

U grafů s malým počtem hran obsahují matice velký počet nul. Proto není pro praxi příliš vhodný.

### 2.4.4 Matice incidence

Pro orientovaný graf  $G$  bez smyček zvolíme (libovolně, ale pevně) jak pořadí jeho vrcholů, tak i pořadí hran, pak můžeme přiřadit grafu  $G$  obdélníkovou matici incidence (též incidenceční matici)  $B_G$  typu  $(n, m)$  (kde  $n$  je počet hran a  $m$  je počet vrcholů) předpisem

$$b_{ij} = \begin{cases} 1, & \text{jestliže } v_i \text{ je počátečním vrcholem hrany } e_j, \\ -1, & \text{jestliže } v_i \text{ je koncovým vrcholem hrany } e_j, \\ 0 & \text{v ostatních případech.} \end{cases}$$

Incidenční matice pro neorientované grafy bez smyček je definována předpisem

$$b_{ij} = \begin{cases} 1, & \text{jestliže } v_i \text{ je incidentní s hranou } e_j, \\ 0 & \text{v ostatních případech.} \end{cases}$$

Tento způsob je sice matematicky elegantní, ale neúsporný, protože v každém sloupci matice jsou pouze dvě nenulové hodnoty.

#### 2.4.5 Matice dostupnosti

Matice dostupnosti  $D_G$  je čtvercová matice řádu  $n$  (kde  $n$  je počet vrcholů) daná předpisem

$$d_{ij} = \begin{cases} 1, & \text{jestliže vrchol } v_j \text{ je } \textit{dostupný} \text{ z vrcholu } v_i, \\ 0 & \text{v ostatních případech.} \end{cases}$$

#### 2.4.6 Prüferovo číslo

Jde o způsob určený pouze pro *stromy* jejichž vrcholy lze jednoznačně uspořádat. V algoritmech 1 a 2 je k uspořádání použito porovnání dle velikostí (tj. pro každé dva vrcholy jde rozhodnout, který z nich je menší).

Prüferovo číslo  $P(T) = \{p_1, p_2, \dots, p_{n-2}\}$ , kde  $n$  je počet vrcholů stromu.

```

k = 1;
repeat
    Necht x je list s nejmenším číslem a y je jeho předchůdce. Pak y je rovno p_k.
    Odstranit x a hranu (x, y) z dalších úvah;
    Jestliže list x byl jediným následníkem vrcholu y, pak se y stane listem;
    k = k + 1;
until zůstaly pouze dva vrcholy
    
```

**Algoritmus 1:** Určení Prüferova čísla [12]

```

Z P(T) určit stupně vrcholů;           % stupeň každého vrcholu je o 1 vyšší než
počet výskytů jeho čísla v P(T)
i = 1;
repeat
    Určit vrchol v_0 s nejnižším číslem a stupněm 1, (v_0, p_i) je hranou stromu. Snížit
    stupně vrcholů v_0 a p_i o 1;
    i=i+1;
    k = k + 1;
until stupně všech vrcholů jsou rovny 0 s výjimkou dvou vrcholů, jejich stupeň je
roven 1.           % Poslední hrana ve stromu spojuje právě tyto dva vrcholy
    
```

**Algoritmus 2:** Rekonstrukce stromu z Prüferova čísla [12]

## Kapitola 3

# Barvení grafu

Barvení grafu se dělí na vrcholové a hranové. Vrcholové obarvení grafu je takové obarvení vrcholů, kdy žádné dva sousední vrcholy nemají stejnou barvu, oproti tomu hranové barvení je takové obarvení hran, kdy žádné dvě hrany incidující se stejným vrcholem nemají stejnou barvu.

### 3.1 Chromatické číslo

**Vrcholové chromatické číslo** (též **vrcholová barevnost**) grafu  $G$  je minimální počet barev potřebných k obarvení vrcholů grafu tak, aby dva sousedící vrcholy neměli stejnou barvu. Značí se  $\chi_v(G)$ .

**Hranové chromatické číslo** (též **hranová barevnost**) grafu  $G$  je minimální počet barev potřebných k obarvení hran grafu tak, aby žádné dvě hrany incidující se stejným vrcholem neměly stejnou barvu. Značí se  $\chi_e(G)$ .

**$k$ -barevný graf** je takový, pro jehož obarvení stačí  $k$  barev. Tedy  $k \geq \chi(G)$ .

V další části práce bude pod pojmem chromatické číslo (barevnost) myšleno vrcholové chromatické číslo a bude značeno  $\chi(G)$ .

Hodnoty chromatického čísla pro specifické grafy (převzato z [1])

$\chi(G) = 1$  pro diskrétní graf

$\chi(G) = 2$  pro bipartitní graf

$\chi(G) = 2$  pro strom

$\chi(G) \geq 3$  obsahuje-li graf kružnici liché délky

$\chi(G) \leq 4$  pro rovinný graf (více v kapitole 3.2.1)

$\chi(G) \leq |V|$  pro libovolný graf

$\chi(G) = |V|$  pro úplný graf

$\chi(G) \leq \max\{d_G(x) \mid x \in V\} + 1$  pro libovolný graf

## 3.2 Praktické využití

Barvení vrcholů v této práci slouží ke grafickému znázornění el. proudu protékajícího elektrickou rozvodnou sítí nízkého napětí. Mimoto lze barvení grafu aplikovat na různé praktické úlohy, jako je např. barvení mapy, plánování rozvrhu hodin či schůzek, řízení světelných křižovatek, kontrola předepisování léků, které by se neměly kombinovat a spoustu dalších. Zde bude popsáno několik reálných problémů řešených pomocí vrcholového barvení grafů.

### 3.2.1 Barvení mapy

Z důvodu dobré přehlednosti je vhodné obarvovat mapy co nejmenším počtem barev.

Převod mapy na graf probíhá tak, že vrcholy představují jednotlivé státy a hrany určují, které státy (vrcholy) spolu sousedí (mají společnou hranici delší jak jediný bod). Takto vytvořený graf je *rovinný* a proto jej lze obarvit čtyřmi barvami (viz Problém čtyř barev).

#### Problém čtyř barev

Jedná se o otázku, zda lze zeměpisnou mapu se státy obarvit čtyřmi barvami tak, aby dva sousední státy neměly stejnou barvu. Přičemž sousedními státy se rozumí dva územní celky mající společnou hraniční čáru (nikoli pouze bod).

Dokázat, že libovolnou mapu v rovině či na kulové ploše lze obarvit čtyřmi barvami, se podařilo až v roce 1976. Více o problému čtyř barev je možné nalézt ve článku [13], z kterého jsem čerpal informace o této problematice.

### 3.2.2 Plánování rozvrhu

Často je potřeba naplánovat rozvrh. Přitom je nezbytné (žádoucí), aby jeden vyučující (student) neměl více přednášek zároveň.

Vrcholy jsou jednotlivé přednášky a hranami se spojí ty, které nemohou probíhat zároveň. Vrcholy (přednášky) obarvené jednou barvou mohou být vyučovány ve stejný čas.

Počet barev udává nejmenší možný počet po sobě jdoucích přednášek potřebný k realizaci všech přednášek.

### 3.2.3 Předepisování léků

Je velice důležité, aby nedocházelo k užití více léků, které se nemají podávat zároveň – ovlivňují se.

Tento problém lze řešit obarvením grafu kde vrcholy představují jednotlivé léky a hrany se přidávají mezi ty léky (vrcholy), které by neměli být užity najednou.

Množství barev poté udává minimální počet dávek nutný k požití všech léků.

### 3.2.4 Skladování nebezpečných látek

Existují látky, které by neměly být uloženy vedle sebe (např. ve stejné místnosti), protože by spolu mohly interagovat<sup>1</sup>.

Každou látku představuje jeden vrchol a pakliže dvě látky (vrcholy) nemohou být uskladněny vedle sebe, přidá se mezi ně hrana.

Barevné skupiny vrcholů poté označují látky, které je možno nechat společně na jednom místě.

---

<sup>1</sup>interakce – vzájemné působení dvou nebo více činitelů



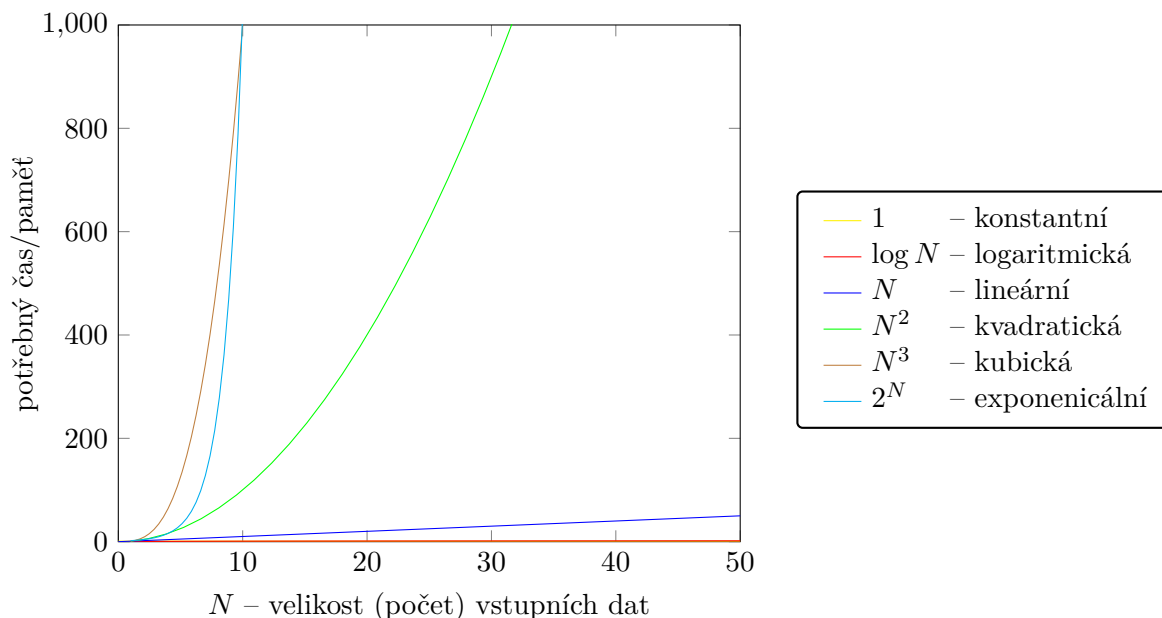
### 3.3 Časová a paměťová složitost

Dříve než budou popsány konkrétní algoritmy se seznámíme s pojmy o časové a paměťové složitosti algoritmů.

Tyto složitosti se používají pro měření kvality algoritmu z hlediska rychlosti výpočtu a paměťové náročnosti. Zjištění časové a paměťové složitosti je důležité především u algoritmů pracujících s velkým množstvím vstupních dat. Čím větší množství dat přijde na vstup, tím více se projeví rozdíly v efektivitě jednotlivých algoritmů. Pro klasifikaci (která je nezávislá na konkrétním technickém vybavení či programovacím jazyce) časové/paměťové náročnosti algoritmu zpracovávajícího velký objem dat se používá tzv. asymptotická složitost.

**Asymptotická složitost** (značí se  $O$ ) určuje jakým způsobem se bude měnit chování (rychlost/velikost potřebné paměti) algoritmu v závislosti na velikosti vstupních dat. V obrázku 3.1 jsou uvedeny typické příklady asymptotické složitosti.

Vždy, když se v textu píše o časové/paměťové složitosti, je tím myšlena asymptotická časová/paměťová složitost.



Obrázek 3.1: Příklady asymptotické složitosti

**Časová složitost** [7] je nejhorší možná doba běhu algoritmu v závislosti na velikosti vstupních dat. Pokud dobu nelze přesně spočítat, tak se stanoví jako nejlepší horní odhad.

**Paměťová (též prostorová) složitost** [7] je největší možná velikost paměti, která je potřebná pro vykonání algoritmu, v závislosti na velikosti vstupních dat. Pokud velikost paměti nelze přesně spočítat, tak se stanoví jako nejlepší horní odhad.

Výsledkem algoritmu 3 je funkce  $g(N)$ . Algoritmus tedy bude mít časovou složitost  $O(g(N))$ .

1. Určíme počet  $f(N)$  provedených elementárních operací algoritmu v závislosti na vstupu o velikosti  $N$ , v nejhorším (nejdelším) možném průběhu algoritmu. Pokud neumíme určit počet operací přesně, najdeme alespoň co nejlepší horní odhad na  $f(N)$ .
2. Ve výsledné formuli  $f(N)$ , která je součtem několika členů, ponecháme pouze nejrychleji rostoucí funkci, ostatní zanedbáme.
3. Vypustíme multiplikační konstanty.

**Algoritmus 3:** Získání časové složitosti [7]

Po malé úpravě algoritmu 3 jej lze použít i pro výpočet paměťové složitosti algoritmu. Jen je nutno za počet provedených operací dosadit velikost paměti potřebné pro vykonání daného algoritmu.

## Kapitola 4

# Algoritmy pro vrcholové obarvení grafu

Tato kapitola se zabývá nejrozšířenějšími obecnými algoritmy pro vrcholové barvení grafu a jejich konkrétními modifikacemi.

Algoritmy uvedené níže se snaží obarvit graf  $G$  co možná nejmenším počtem barev  $k$ . V ideálním případě by se počet barev použitých k obarvení grafu rovnal jeho chromatickému číslu, tedy  $k = \chi(G)$ . Čím menší je rozdíl mezi  $k$  a  $\chi(G)$ , tím je graf  $G$  kvalitněji obarven.

Pro účely barvicích algoritmů je nutné převést multigrafy na prosté grafy (odstraněním násobných hran) a orientované grafy na neorientované (zanedbáním orientace hran).

### 4.1 Sekvenční (hladové) barvení grafu

Jedná se o nejjednodušší způsob obarvování grafu. Je velice rychlý, ale nepodává příliš dobré výsledky z hlediska počtu použitých barev.

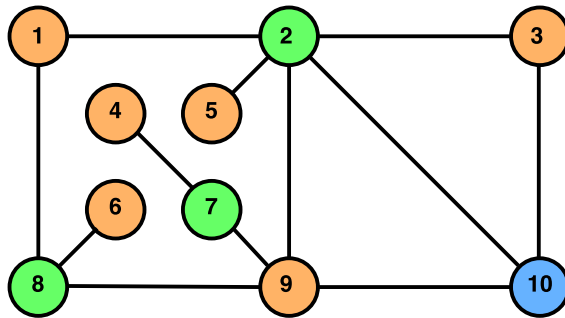
Princip je takový, že se vybere dosud neobarvený vrchol, který se obarví nejmenší možnou barvou. Kvalita algoritmu 4 tedy závisí na pořadí vybírání vrcholů, které ještě nemají přiřazenou barvu.

```
[inicializace]
K = 0; % počet použitých barev
while [Test ukončení] existuje neobarvený vrchol grafu do
  [Volba vrcholu]
  zvolíme dosud neobarvený vrchol  $v_0$ ;
  [Určení barvy]
  najdeme minimální přirozené číslo  $b$  tak, že neexistuje vrchol spojený s  $v_0$  a již
  obarvený barvou  $b$ ;
  [Obarvení  $v_0$ ]
   $v_0$  obarvíme barvou  $b$ ;
  if  $b > K$  then
    |  $K = b$ 
  end
end
```

**Algoritmus 4:** Sekvenční barvení [12]

Dále jsou uvedeny některé základní heuristiky pro výběr vrcholu. Vždy je nutné zvážit, kterou z nich při implementaci použít, protože na rozdílných grafech budou tyto heuristiky podávat různě kvalitní výsledky. Tj. na jednom grafu může poskytnout daná heuristika lepší obarvení než jiná, avšak na druhém grafu tomu může být zcela naopak.

Na obrázku 4.1 je znázorněn příklad grafu, který byl barven postupně shora dolů a zleva doprava. Vrcholy zde znázorňují kolečka, ty jsou propojeny čarami (hranami). Číslo v každém kolečku značí krok, ve kterém byl vrchol vybrán k obarvení. Pokud u dále popsaných heuristik nastane situace, kdy bude pro výběr následujícího vrcholu určeno více vrcholů, pak se z nich, podobně jako v případě obr. 4.1, vybere vrchol, který je na daném obrázku nejvýše vlevo.



Obrázek 4.1: Příklad pro popis

Popis základních heuristik i jejich kombinací vychází z článku [6] zabývající se algoritmy pro barvení grafu.

#### 4.1.1 Heuristika First Fit (FF)

Vrchol  $v_0$  se vybere náhodně z množiny dosud neobarvených uzlů. Toto lze implementovat s lineární časovou složitostí  $O(N)$ .

Oproti této heuristice se následující tři snaží obarvovat nejprve nejobtížněji obarvitelné vrcholy a tím snížit počet použitých barev potřebných k obarvení grafu.

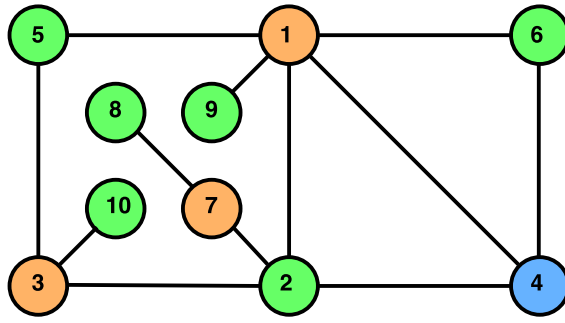
#### 4.1.2 Heuristika Largest Degree Ordering (LDO)

Za  $v_0$  se dosadí neobarvený vrchol s největším stupněm. Z toho vyplývá, že vrcholy s velkým stupněm budou obarveny dříve než vrcholy s malým stupněm. Viz obrázek 4.2. Tuto heuristiku lze implementovat s kvadratickou časovou složitostí  $O(N^2)$ .

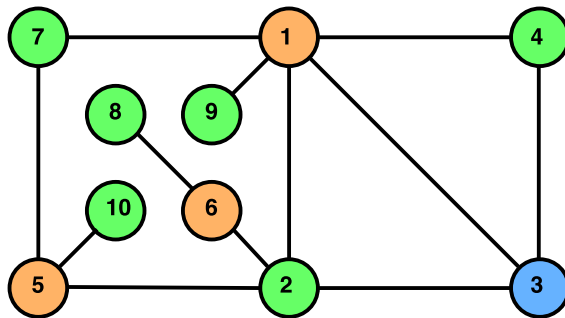
Před samotnou činností algoritmu je vhodné si vrcholy seřadit sestupně podle velikosti stupně.

#### 4.1.3 Heuristika Saturation Degree Ordering (SDO)

Za  $v_0$  se zvolí ten vrchol, který má největší stupeň nasycení. Stupeň nasycení  $D(v)$  je počet různých barev jeho již obarvených sousedů. Pokud má největší hodnotu  $D(v)$  více vrcholů, vybere se ten, který má v doposud neobarveném podgrafu největší stupeň jak je patrné z obrázku 4.3. Tato heuristika dosahuje zpravidla lepších výsledků než LDO a lze ji zprovoznit s časovou složitostí  $O(N^2)$ .



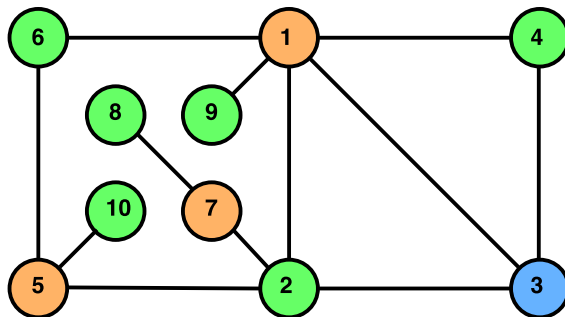
Obrázek 4.2: LDO



Obrázek 4.3: SDO

#### 4.1.4 Heuristika Incidence Degree Ordering (IDO)

Je velice podobná heuristice SDO, ale namísto velikosti stupně nasycení zaměřuje svůj výběr na počet incidujících (sousedních) vrcholů, které již byly obarveny. Za  $v_0$  se tedy zvolí ten vrchol, který má největší počet obarvených sousedů. Pokud je takovýchto vrcholů více, pak se stejně jako v případě SDO vybere ten, který má v doposud neobarveném podgrafu největší stupeň. Postup této heuristiky je znázorněn na obrázku 4.4. U IDO je časová složitost lineární funkcí počtu hran. Tedy  $O(N)$ .

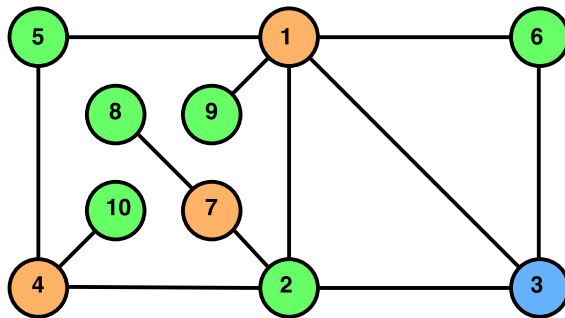


Obrázek 4.4: IDO

Pro zlepšení obarvení grafu, nebo-li pro menší počet použitých barev, se využívá kombinací základních heuristik. Zde budou představeny dva způsoby kombinace a to heuristika kombinující LDO s IDO a SDO s LDO.

#### 4.1.5 Heuristika kombinující LDO a IDO

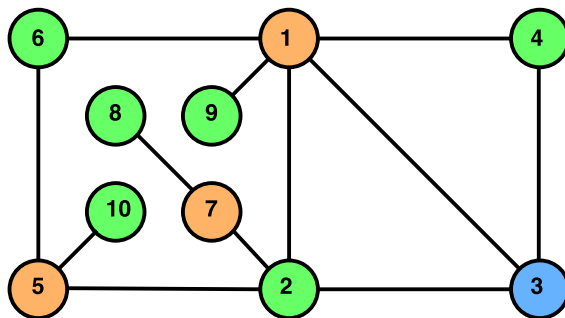
Heuristika pracuje stejně jako LDO do doby, než nalezne dva (a více) vrcholy se stejným stupněm vrcholu. Průběh je ukázán na obrázku 4.5. Pro rozhodnutí, který z nich se má vybrat, použije stejně jako IDO počet sousedních dosud neobarvených uzlů. Heuristiku je možné implementovat s časovou složitostí  $O(N^2)$ .



Obrázek 4.5: LDO a IDO

#### 4.1.6 Heuristika kombinující SDO a LDO

Heuristika funguje stejně jako SDO do doby, než nalezne dva (či více) vrcholy se stejným stupněm nasycení. Z nich vybere ten, který má největší stupeň vrcholu. Postup je znázorněn na obrázku 4.6. Tuto modifikovanou SDO heuristiku je možné implementovat s kubickou časovou složitostí  $O(N^3)$ .



Obrázek 4.6: SDO a LDO

## 4.2 Barvení grafu pomocí nezávislých množin

Oproti sekvenčnímu barvení grafu (kapitola 4.1) podává barevní pomocí nezávislých množin lepších výsledků obarvení (tj. obecně potřebuje méně barev k obarvení grafu), ale za cenu náročnější implementace.

Algoritmus pro barvení grafu  $G$  pomocí nezávislých množin funguje tak, že nalezne maximální nezávislou množinu v neobarveném podgrafu  $G$  a poté obarví všechny vrcholy této množiny jednou barvou. Výběr další množiny a její barvení probíhá tak dlouho, dokud není obarven celý graf  $G$ . Toto je ukázáno na algoritmu 5.

```
[inicializace]
K = 0;                                     % počet použitých barev
repeat
  [Volba nové barvy]
  K = K + 1;
  [Určení nezávislé množiny]               % X je nezávislá množina vrcholů
  v grafu G určíme maximální nezávislou množinu X;
  [Obarvení vrcholů z nezávislé množiny]
  všechny vrcholy množiny X obarvíme barvou K a potom je spolu s hranami,
  které jsou s nimi incidentní, odstraníme z grafu G;
until v G není neobarvený vrchol
```

**Algoritmus 5:** Barvení grafu pomocí nezávislých množin [12]

Podobně, jako u sekvenčního barvení záleží kvalita obarvení na pořadí výběru vrcholů, u barvení pomocí nezávislých množin se kvalita odvíjí od výběru nezávislé množiny.

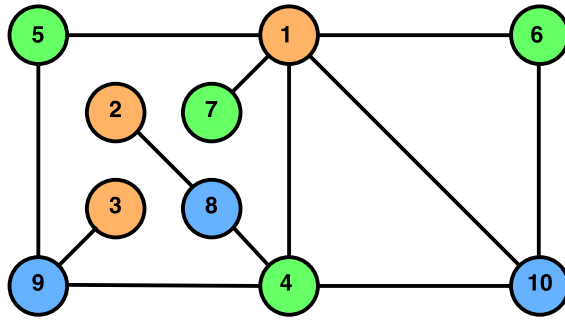
Dále jsou uvedeny dva algoritmy pro vyhledávání maximálních nezávislých množin.

### 4.2.1 Algoritmus pro hledání maximální nezávislé množiny

U prvního z nich, algoritmu 6, je kvalita závislá na volbě vrcholu. Tuto volbu je možné provádět např. pomocí náhodného výběru či zvolení vrcholu s největším stupněm. Další možnosti (včetně dvou uvedených) jsou podobné heuristikám v kapitole 4.1. Na obrázku 4.7 je ukázán průběh barvení pomocí tohoto algoritmu.

```
[inicializace]
X = ∅;
Y = V;                                     % Y ⊆ V(G) - pomocná proměnná
repeat
  [Volba vrcholu]
  zvolíme v0 ∈ Y;
  [Aktualizace X, Y]
  X = X ∪ {v0};
  Y = Y \ {v0} \ VG(v0);           % z Y se odstraní v0 a všechny vrcholy
  % sousedící s v0
until Y == ∅                               % v X je maximální nezávislá množina vrcholů
```

**Algoritmus 6:** Hledání maximální nezávislé množiny vrcholů  $X$  [12]



Obrázek 4.7: Barvení pomocí max. nezávislých množin

### 4.2.2 Pravděpodobnostní alg. pro hledání maximální nezávislé množiny

Algoritmus 7 využívá pro hledání maximální nezávislé množiny pravděpodobnost.

Algoritmus začíná zkopírováním vrcholů. Poté do potenciálně nezávislé množiny  $X$  přidává vrcholy s pravděpodobností  $\frac{1}{2 \cdot d(v)}$  (kde  $d(v)$  je stupeň vrcholu), vyjma případu, kdy uzel nemá žádné sousedy, tehdy je přidán okamžitě. Dále v této množině algoritmus nalezne všechny sousedy a z každého páru odstraní vrchol s větším stupněm, čímž z  $X$  udělá nezávislou množinu. Nyní  $X$  sjednotí s výslednou nezávislou množinou. V další iteraci již nebude používat vrcholy z  $X$  ani jejich sousedy. Pokud existují vrcholy, se kterými lze pracovat, pak začne novou iteraci vybíráním vrcholů do prázdné množiny  $X$ .

Způsob, jakým je algoritmus zapsán, umožňuje paralelizaci a tím i snížení časové náročnosti.

## 4.3 Barvení grafu slepováním uzlů

Myšlenka barvení grafu pomocí slepování uzlů (vrcholů) je založena na faktu, že dva nesousední vrcholy mohou mít stejnou barvu.

Algoritmus 8 se snaží zredukovat původní graf na úplný graf. Toho dosáhne tím, že v každé iteraci určí dva různé nesousední vrcholy, které spojí do jednoho vrcholu. Tento nový vrchol je spojen se stejnými vrcholy v grafu, se kterými byl spojen alespoň jeden ze sloučených vrcholů.

Kvalita obarvení grafu záleží na zvoleném výběru vrcholů  $v_1$  a  $v_2$ . Pro vhodný výběr existuje řada heuristik, z nichž v této práci jsou popsány heuristiky RLF (4.3.2) a Dutton & Brigham (4.3.1).

### 4.3.1 Heuristika Dutton & Brigham

Tato kapitola vychází z článku [4].

Pro každé dva nesousední vrcholy  $v_i$  a  $v_j$  se zjistí počet společných sousedů  $c_{ij}$ . Poté se vybere dvojice uzlů  $v_i, v_j$ , pro kterou platí  $c_{ij} \geq c_{rs}$ , kde  $v_r, v_s$  je jiná libovolná dvojice nesousedních vrcholů. Jinými slovy, vybere se ta dvojice nesousedních vrcholů, které spolu mají nejvíce společných sousedů. Následně se sloučí  $v_i$  s  $v_j$  a upraví se hodnota  $c_{rs}$  všech ovlivněných nesousedních párů. Dokud se pomocí slučování vrcholů nestane graf úplným, pokračuje se výběrem další dvojice  $v_i, v_j$ . Po ukončení algoritmu je počet vrcholů úplného grafu roven počtu barev použitých k jeho obarvení (viz kapitola 3.1).

Časová složitost algoritmu je  $O(N^3)$  a paměťová složitost je  $O(N^2)$ .



```

[inicializace]
M = ∅;                                     % nezávislá množina
V' = V;                                    % kopie vrcholů grafu
while V' ≠ ∅ do
  X = ∅;                                    % pomocná množina
  [Výběr vrcholů - možnost paralelizace]
  forall the v ∈ V' do
    if d(v) = 0 then
      | přidání v do X;
    else
      | přidání v do X s pravděpodobností  $\frac{1}{2 \cdot d(x)}$ ;
    end
  end
  [Odstranění sousedních vrcholů - možnost paralelizace]
  forall the v, u ∈ X do
    if v a u sousedí                         % u ∈ V(v)
    then
      | Vrchol s vyšším stupněm je smazán z X;
    end
  end
  [Aktualizace M, V']
  M = M ∪ X;
  z V' jsou odebrány vrcholy v X a jejich sousedé;
end
                                     % v M je maximální nezávislá množina

```

**Algoritmus 7:** Hledání maximální nezávislé množiny pomocí pravděpodobnosti

```

while [Test ukončení] G není zredukován na úplný graf do
  [Určení slepovaných vrcholů]
  určíme 2 navzájem různé vrcholy  $v_1, v_2$  nespojené hranou;
  [Slepení vrcholů]
  vrcholy  $v_1, v_2$  nahradíme jediným vrcholem  $v_0$ , který spojíme se všemi vrcholy
  grafu G, se kterými byl spojen alespoň jeden z vrcholů  $v_1, v_2$ ;
end
[Obarvení]
každý vrchol vzniklého úplného grafu obarvíme jinou barvou, vrcholy původního
grafu dostanou barvy podle barev vrcholů, do kterých postupným spojováním přešly;

```

**Algoritmus 8:** Barvení grafu slepováním vrcholů [12]

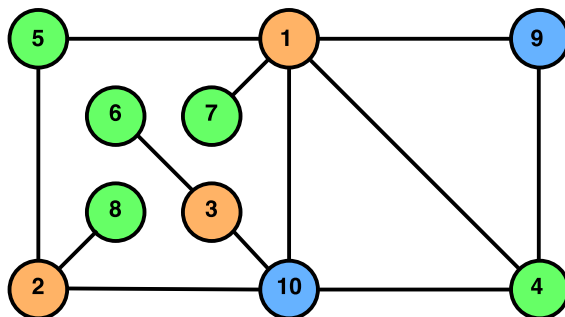
### 4.3.2 Heuristika Recursive Largest First (RLF)

Heuristika Recursive Largest First pracuje se třemi množinami. První je množina vrcholů, do které postupně přidává obarvené vrcholy. Dále si udržuje pomocnou množinu obsahující v danou chvíli neobarvitelné vrcholy. A z poslední množiny potenciálně obarvitelných vrcholů vybírá vrchol k obarvení a to na základně počtu sousedů v množině s neobarvitelnými uzly.

Celý postup je popsán v algoritmu 9 a ukázán na obrázku 4.8.

```
C = ∅; % množina obarvených vrcholů
V' = V; % množina potenciálně obarvitelných vrcholů
U = ∅; % množina neobarvitelných vrcholů
K = 0; % barva
[nová barva]
K = K + 1;
vybereme vrchol v ∈ V' s největším stupněm;
[obarvení vrcholu]
obarvíme vrchol v barvou K;
přesuneme vrchol v ∈ V' do C a všechny vrcholy z množiny V' sousedící s vrcholem
v přesuneme do množiny U;
if V' ≠ ∅ then
    [výběr dalšího vrcholu]
    vybereme vrchol v ∈ V' s největším počtem sousedů v U;
    poté se znovu vrátíme na krok [obarvení vrcholu];
else if C = V then
    [výsledek]
    ukončení algoritmu; % graf je obarven K barvami
else
    [aktualizace hodnot V' a U]
    V' = U; % do V' jsou přiřazeny všechny doposud neobarvené vrcholy
    U = ∅;
    a skočíme zpět na krok [nová barva];
end
```

Algoritmus 9: Recursive Largest First [9]



Obrázek 4.8: Barvení pomocí slepování uzlů (RLF)

## 4.4 Výměna barev

Algoritmus výměny barev sám o sobě vrcholy neobarvuje, proto se vždy používá v kombinaci s nějakým jiným heuristickým algoritmem. Tento algoritmus pracuje nad již korektně obarveným grafem nebo nad doposud obarvenou částí grafu. Principem výměny barev je přebarvení grafu tak, aby mohl být vybraný vrchol obarven již použitou barvou. K tomu je zapotřebí, aby barvený graf obsahoval alespoň dvě barvy.

Zde popsaný algoritmus 10 je variantou z [11], pracující pouze pro korektně obarvené grafy.

Pro největší použitou barvu  $B$  musí platit  $B \geq 2$ .

Vybereme vrchol  $v$  jehož barvu  $b$  chceme nahradit jinou již použitou barvou. Zvolíme dvě různé barvy  $c, d$ , které byly již použity ve stávajícím obarvení. Necht' barva  $c$  je v této dvojici tou barvou která má nižší číslo. Navíc volíme barvy  $c$  a  $d$  tak, aby byly odlišné od barvy  $b$ , kterou chceme odstranit. Formálně zapsáno

$$1 \leq c < d \leq B \wedge c \neq b \wedge d \neq b \quad (4.1)$$

V barveném grafu najdeme jeho podgraf, který obsahuje jen vrcholy s barvou  $c$  nebo  $d$ . Typicky je tento podgraf složen z více komponent. Ty rozdělíme do čtyř tříd.

Třída  $A$  – obsahuje komponenty, jejichž žádný vrchol ve výchozím grafu  $G$  nesousedí s vrcholem  $v$ , u kterého chceme změnit barvu.

Třída  $B$  – obsahuje komponenty, které mají vrcholy, jež sousedí ve výchozím grafu  $G$  s vrcholem  $v$ , přičemž tyto vrcholy jsou všechny obarveny jen barvou  $c$

Třída  $C$  – obsahuje komponenty, které mají vrcholy, jež sousedí ve výchozím grafu  $G$  s vrcholem  $v$ , přičemž tyto vrcholy jsou všechny obarveny jen barvou  $d$

Třída  $D$  – obsahuje komponenty, které mají vrcholy, jež sousedí ve výchozím grafu  $G$  s vrcholem  $v$ , přičemž některé vrcholy jsou obarveny barvou  $c$  a jiné barvou  $d$ .

**if** *Je-li třída  $D$  neprázdná* **then**

U zvolené dvojice barev nelze provést výměnu vedoucí k požadovanému obarvení vrcholu  $v$ . V tomto případě je možnost vyzkoušet algoritmus znovu s jinou dvojicí barev  $c$  a  $d$ , ale pouze za předpokladu, že daná dvojice vyhovuje vztahu 4.1. Jinak algoritmus končí s tím, že se nepodařilo obarvit vrchol  $v$  požadovaným způsobem.

**else**

Existují dvě možnosti přebarvení. Jelikož jsou téměř totožné, popíši zde pouze jednu a druhou možnost dopíši do závorky.

Ve všech komponentách třídy  $B$  (třídy  $C$ ) zaměníme barvy  $c$  a  $d$ , tj. vrcholy obarvené barvou  $c$  budou nyní obarveny barvou  $d$  a naopak. Po této výměně už žádný vrchol sousedící s vrcholem  $v$  nebude obarven barvou  $c$  (barvou  $d$ ), díky čemuž vrchol  $v$  můžeme obarvit touto barvou.

**end**

**Algoritmus 10:** Algoritmus výměny barev

Opakovaným algoritmu 10 s vhodným výběrem vrcholu  $v$  a barev  $c$  a  $d$  lze dosáhnout i vícenásobného vylepšení obarvení.

## Kapitola 5

# Algoritmy pro zjišťování barevnosti

V této kapitole jsou popsány exaktní algoritmy, které jsou schopny obarvit graf počtem barev, který je roven chromatickému číslu grafu.

### 5.1 Barvení hrubou silou

Barvení hrubou silou je exaktní algoritmus sloužící k určení všech možných způsobů  $k$ -obarvení grafu  $G$ .

Algoritmus vychází z jednoduché myšlenky. Prochází všechny možnosti rozložení  $k$  barev na  $n$  vrcholů a u každé z nich kontroluje, zda se jedná o přípustné obarvení grafu, tj. zda dva libovolné sousedící vrcholy nemají přiřazenou stejnou barvu. Pokud nalezne přípustné obarvení, pak lze graf obarvit  $k$  barvami.

Počet možností, kterými lze  $n$  vrcholů obarvit  $k$  barvami, je  $k^n$ . Proto je časová složitost barvení hrubou silou exponenciální. Kvůli své velké složitosti jej prakticky nelze využít na větších grafech.

Pro nalezení chromatického čísla grafu je zapotřebí spouštět algoritmus opakovaně a to vždy s počtem barev  $k$  o jedno vyšším než v předchozí iteraci. Algoritmus pracuje postupně s  $k = 1, 2, \dots, n$ . První nalezené řešení udává nejmenší možný počet barev potřebný k obarvení grafu, nebo-li chromatické číslo grafu.

### 5.2 Backtracking

Oproti barvení hrubou silou, algoritmy využívající zpětného navracení neprocházejí všechny možnosti obarvení – jsou proto rychlejší. Rychlost závisí především na očíslování vrcholů.

Následující algoritmus 11 ukazuje využití backtrackingu, u kterého velmi záleží na očíslování vrcholů. Je vhodné, aby se nejmenšími čísly očíslovala co největší klika a přitom číslovat nižšími čísly vrcholy s velkým stupněm. Dále se doporučuje číslovat sousední vrcholy sousedními čísly (pokud je to možné).

Pokud stačí zjistit, zda je graf obarvitelný  $k$  barvami, pak lze algoritmus 11 upravit a zrychlit. A to tak, že při inicializaci se nastaví  $OMEZ = k + 1$ . V 3. kroku, když je nalezeno obarvení  $k$  barvami, výpočet úspěšně končí. Pokud je výpočet ukončen v kroku 5, pak nelze graf obarvit  $k$  barvami.

1. [inicializace]
 

$x = 1;$  % výběr prvního vrcholu  
 $B(x) = 1;$  % barva  
 $OMEZ = n + 1;$  %  $n$  je počet vrcholů grafu
2. [postup vpřed – obarvení dalšího vrcholu]
 

Položíme  $x = x + 1$ . Jestliže  $x > n$ , pokračujeme podle kroku 3. V opačném případě položíme  $B(x) = \text{minimum ze všech barev nevyskytujících se v množině } P(x)$ . Jestliže  $B(x) \geq OMEZ$ , položíme  $y = x$  a pokračujeme podle kroku 4, jinak zopakujeme krok 2.
3. [všechny vrcholy obarveny, počet barev je menší než OMEZ]
 

Pro všechny vrcholy  $x$  grafu  $G$  provedeme  $BARVA(x) = B(x)$  a dále položíme  $OMEZ = \text{maximum z hodnot } BARVA$ . Označme  $y$  první vrchol, který má barvu  $OMEZ$ , a pokračujeme krokem 4.
4. [návrat – pokus o snížení barvy ve vrcholu  $y$ ]
 

Označme  $x$  poslední vrchol z množiny  $P(y)$  a pokračujeme krokem 5.
5. [pokračování návratu – pokus o zvýšení barvy ve vrcholu  $x$ ]
 

Jestliže  $x == 1$ , výpočet končí. V opačném případě položíme  $b = \text{minimum ze všech barev, které jsou větší než } B(x) \text{ a nevyskytují se v množině } P(x)$ . Pokud  $b < OMEZ$  a zároveň  $b \leq x$ , položíme  $B(x) = b$  a pokračujeme podle kroku 2, v opačném případě položíme  $x = x - 1$  a zopakujeme krok 5.

**Algoritmus 11:** Zjištění barevnosti pomocí backtrackingu [2]

## Kapitola 6

# Získání podkladních dat

Jelikož podstatou práce je grafické znázornění proudů tekoucích v zauzlené rozvodné síti nízkého napětí (NN), bylo zapotřebí získat data, na kterých by bylo možno zhodnotit implementované algoritmy a ukázat funkčnost programu.

Distributoři el. energie veřejně neposkytují informace o rozložení el. prvků v rozvodných sítích NN a jejich vlastnostech. Například rozmístění trafostanic, el. vlastnosti vodičů, velikost odběrů v jednotlivých rodinných domech apod. Více informací o získání či generování dat je popsáno v následujících částech.

### 6.1 Rozložení prvků v el. rozvodné síti

Pro vytvoření co nejvíce realistických dat bylo vybráno malé město Slavkov u Brna. Z webové aplikace<sup>1</sup> je možné získat přibližné fyzické umístění vedení NN. Umístění trafostanic lze zjistit v územním plánu města<sup>2</sup>. Počty a rozmístění domů, panelových domů, firem a dalších budov bylo získáno znalostí města a náhledem do mapy<sup>1</sup>.

### 6.2 Vytvoření grafu v programu *yEd*

Pomocí takto získaných informací se vytvoří v programu *yEd Graph Editor*<sup>3</sup> graf znázorňující elektrickou rozvodnou síť (trafostanice, vedení a umístění jednotlivých míst odebírajících elektřinu).

K uložení potřebných informací je nutno v programu přidat k hranám i uzlům další informace. To lze provést změnou v nastavení (*Edit* → *Manage Custom Properties...*). U uzlů (*Node Properties*) jsou přidány vlastnosti *napeti*, *pocet\_odberu*, *typ\_odberu*, *region* a u hran (*Edge properties*) *pocet\_odberu*, *typ\_odberu* a *typ\_vodice*.

Nyní následuje popis těchto vlastností.

**napeti** Uzly s nenulovou hodnotou jsou brány jako trafostanice dodávající zadanou hodnotu napětí.

**pocet\_odberu** je přirozené číslo udávající počet odběrných míst nacházejících se na dané hraně či v daném uzlu.

<sup>1</sup><http://geoportal.eon.cz/itc/default.aspx?serverconf=vsite&wmcid=1143>

<sup>2</sup><http://www.slavkov.cz/images/files/uap/slavkov/3A%20TI%20-%20Energetika,plyn,spoje.pdf>

<sup>3</sup><https://www.yworks.com/products/yed>

**typ\_odberu** je celočíselná hodnota udávající druh odběru. Určuje, zda se jedná o rodinný dům, panelový dům, firmu, hospodu či jiný specifický typ odběrného místa.

**typ\_vodice** je celočíselná hodnota udávající vlastnosti vodiče (vodivost, maximální zatížení).

Vytvořený graf je nutné uložit ve formátu *GraphML* (přípona .graphml). Jedná se o formát založený na *XML*.

Pro ukázkou jsou zde uvedeny vybrané části XML souboru ukázkového grafu, který je zobrazen v příloze na obrázku [B.1](#).

Důležité vlastnosti:

```
<key attr.name="napeti" for="node" id="d4"> <default>0.0</default> </key>
<key attr.name="pocet\_odberu" for="node" id="d6"> <default>0</default> </key>
<key attr.name="typ\_odberu" for="node" id="d7"> <default>0</default> </key>
<key attr.name="region" for="node" id="d8"> <default>0</default> </key>
<key attr.name="pocet\_odberu" for="edge" id="d14"> <default>0</default> </key>
<key attr.name="typ\_odberu" for="edge" id="d15"> <default>1</default> </key>
<key attr.name="typ\_vodice" for="edge" id="d16"> <default>1</default> </key>
```

Příklad vrcholu:

```
<node id="n0" >
  <data key="d4">230.0</data>
  <data key="d11" >
    {<y:GenericNode configuration="com.yworks.flowchart.start2">
      {<y:Geometry height="10.0" width="10.0" x="100.0" y="100.0"/>
        {</y:GenericNode>
      }</data>
    }</data>
  </node>
```

Příklad hrany:

```
<edge id="e2" source="n3" target="n2">
  <data key="d19">
    <y:PolyLineEdge>
      <y:Path sx="0.0" sy="0.0" tx="0.0" ty="0.0">
        <y:Point x="200.0" y="0.0"/>
      </y:Path>
    </y:PolyLineEdge>
  </data>
</edge>
```

## Kapitola 7

# Implementace v prostředí *MATLAB*

*MATLAB* (*matrix laboratory*) je programové prostředí určené pro práci s maticemi. Je optimalizováno pro maticové operace. Také umožňuje práci s řídkými maticemi, díky čemuž je možné snížit paměťovou náročnost programu. Pro tyto výhody jsou jednotlivé proměnné popsané v kapitole 7.1 reprezentovány jako matice (nebo vektory).

Níže jsou popsána řešení využitá při programování výsledného produktu.

### 7.1 Reprezentace dat v systému *MATLAB*

Pro správný chod programu je zapotřebí uchovat důležité informace o vrcholech, hranách a barvách jim přidělených. K tomu slouží níže popsané proměnné.

Ukázky hodnot jednotlivých proměnných pro jednoduchý příklad grafu jsou v příloze B.

#### 7.1.1 Vrcholy

Vrcholy jsou reprezentovány maticí, kde každý řádek obsahuje informace o jednom vrcholu. Číslo řádku udává název vrcholu (též ID vrcholu). V řádku jsou uloženy informace o poloze (souřadnice), velikosti napětí a regionu, do kterého daný vrchol spadá. Získání obsahu proměnné  $V$  je popsáno v kapitolách 7.2 a 7.4.

$$V = \begin{pmatrix} X, Y, NAPETI, REGION \\ \vdots \\ X, Y, NAPETI, REGION \end{pmatrix}$$

Vrchol může představovat trafostanici, odběrové místo nebo pouhé spojení vodičů. Proto jsou zapotřebí další dvě proměnné ( $N$  a  $O$ ) popsané níže. Získání dat je popsáno v kapitole

#### 7.1.2 Napájecí místa – trafostanice

V tomto sloupcovém vektoru se uchovávají ID (čísla řádků) všech napájecích vrcholů (trafostanic). Jeho délka je tedy rovna počtu napájecích vrcholů. V kapitole 7.2 je vysvětleno, jak se získá proměnná  $N$ .

$$N^T = [\text{ID první trafostanice}, \dots, \text{ID poslední trafostanice}]$$



### 7.1.3 Odběrová místa

Informace o odběrových místech jsou uloženy v matici. Na každém řádku je uloženo ID vrcholu, ke kterému daná data patří, dále vypočtená velikost odběru a právě kvůli výpočtu odběru nese i informace o počtu a typu odběru – zda se jedná o rodinný dům, firmu apod. a kolik se takových objektů v daném vrcholu nachází. Naplnění proměnné  $\mathbf{O}$  je popsáno v kapitolách 7.2 a 7.3.

$$\mathbf{O} = \begin{pmatrix} \text{ID\_VRCHOLU}, \text{POCET\_ODBERU}, \text{TYP\_ODBERU}, \text{VELIKOST} \\ \vdots \\ \text{ID\_VRCHOLU}, \text{POCET\_ODBERU}, \text{TYP\_ODBERU}, \text{VELIKOST} \end{pmatrix}$$

### 7.1.4 Hrany

Hrany jsou reprezentovány maticí, kde každý řádek obsahuje informace o jedné hraně. Číslo řádku udává název hrany (též ID hrany). V řádku jsou uloženy informace o počátečním a koncovém vrcholu, délce, typu vodiče a velikosti protékajícího proudu. Získání obsahu proměnné  $\mathbf{H}$  je popsáno v kapitolách 7.2 a 7.4.

$$\mathbf{H} = \begin{pmatrix} \text{ODKUD}, \text{KAM}, \text{DELKA}, \text{TYP\_VODICE}, \text{PROUD} \\ \vdots \\ \text{ODKUD}, \text{KAM}, \text{DELKA}, \text{TYP\_VODICE}, \text{PROUD} \end{pmatrix}$$

### 7.1.5 Matice skupin

Jedná se o řídkou<sup>1</sup> čtvercovou matici řádu  $n$  (kde  $n$  je počtu vrcholů grafu), kde jsou vyplněny jen řádky odpovídající ID napájecích vrcholů. Jeden vyplněný řádek nese informace o tom, kterými vrcholy protéká proud z dané trafostanice. Též lze snadno zjistit, z kolika trafostanic je napájen určený vrchol. Vytvoření matice  $\mathbf{S}$  je vysvětleno v kapitole 7.5.

$$\mathbf{S} = \begin{pmatrix} (1, 1) & \dots & (1, n) \\ \vdots & \ddots & \vdots \\ (n, 1) & \dots & (n, n) \end{pmatrix}$$

kde  $n$  je počet řádků v matici  $\mathbf{V}$ .

### 7.1.6 Matice sousednosti

Matice sousednosti má stejný rozměr jako matice skupin  $\mathbf{S}$  a určuje, které trafostanice (barevné skupiny) spolu sousedí. Z důvodu šetření paměti je matice implementována jako řídká<sup>1</sup>. Naplnění matice  $\mathbf{I}$  je vysvětleno v kapitole 7.5.

$$\mathbf{I} = \begin{pmatrix} (1, 1) & \dots & (1, n) \\ \vdots & \ddots & \vdots \\ (n, 1) & \dots & (n, n) \end{pmatrix}$$

kde  $n$  je počet řádků v matici  $\mathbf{V}$ .

<sup>1</sup>Více o řídké matici (`sparse`) v nápovědě `MATLABu`.

### 7.1.7 Barvy

Jde o řádkový<sup>1</sup> sloupcový vektor s informacemi o barvách přidělených jednotlivým napájecím vrcholům (trafostanicím). Barvy jsou reprezentovány celými čísly. Hodnoty vektoru  $B$  jsou barvy získané některým z implementovaných barvicích algoritmů vyjmenovaných v kapitole 7.6.

$$B = \begin{pmatrix} \text{barva první trafostanice} \\ \vdots \\ \text{barva poslední trafostanice} \end{pmatrix}$$

## 7.2 Získání dat z XML souboru

Soubor, jehož vytvoření je popsáno v kapitole 6, se načte pomocí funkce `xmlread`. Následně se vyhledají, zpracují a uloží potřebné údaje do proměnných  $V$ ,  $N$ ,  $O$  a  $H$  (jejich struktura je popsána v kapitole 7.1).

Vytvořený XML parser nejprve projde všechny vrcholy (elementy se jménem `node`). Veškeré hodnoty matice  $V$  (souřadnice  $X$  a  $Y$ , `NAPETI` a `REGION`) stačí najít v načteném souboru a uložit. Pokud vrchol obsahuje nenulový počet odběrů, tak je vytvořeno nové odběrné místo.

Následně projde i hrany (elementy se jménem `edge`). U hran je určen počáteční a koncový vrchol (`ODKUD` a `KAM`), typ vodiče (`TYP_VODICE`), počet a typ odběrů. Velikost protékajícího proudu (`PROUD`) se dopočítá později, proto se prozatím nastaví na nulu. Ze souřadnic počátečního a koncového vrcholu hrany je pomocí Pythagorovy věty vypočtena délka hrany (`DELKA`). Hrana se v závislosti na počtu odběrů rozdělí na více menších, stejně dlouhých hran. Aby měla každá hrana počáteční a koncový bod, je vytvořeno tolik nových vrcholů ve  $V$ , kolik odběrů se na hraně nachází. Každý takto vytvořený vrchol odpovídá nově vytvořenému odběrnému místu.

Pro každé nově vytvořené odběrové místo je přidán záznam (řádek) v matici  $O$  obsahující ID vrcholu (`ID_VRCHOLU`), ke kterému se odběr váže, počet odběrů (`POCET_ODBERU`), typ odběru (`TYP_ODBERU`) a prozatím nulovou velikost odběru (`VELIKOST`), která bude dopočítána později.

Vrcholy s hodnotou napětí (`NAPETI`) větší jak nula jsou považovány za trafostanice. ID těchto vrcholů obsahuje vektor  $N$ .

## 7.3 Výpočet odběru

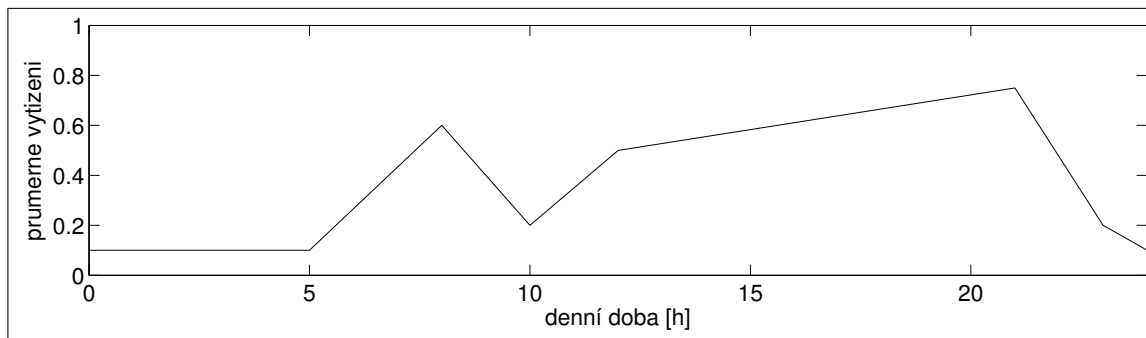
Pro každý odběrný vrchol je v závislosti na typu odběru a denní době vypočtena spotřeba.

Typ odběru udává, jaká je maximální velikost odběru (velikost jističe). Hodnota jističe je vybírána náhodně pro rodinné domy na 20, 25, 32, 40 nebo 50 ampér, pro bytové jednotky na 10, 16 nebo 20 ampér, pro firmy na 63 ampér a všechny ostatní typy na 40 ampér.

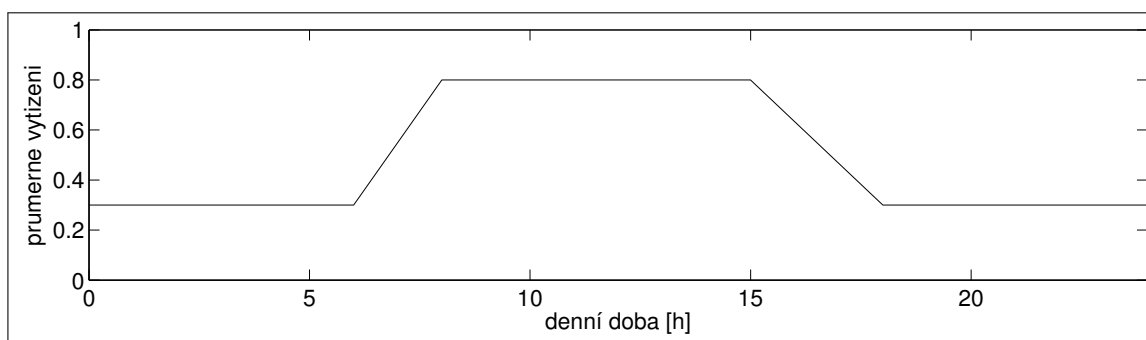
Dále jednotlivé typy určují časovou charakteristiku. V práci jsou implementovány časové charakteristiky<sup>2</sup> pouze pro rodinné a bytové domy (viz obrázek 7.1) a firmy (viz obrázek 7.2). Se všemi ostatními typy se zachází, jako by se jednalo o firmu.

Pokud se k nějakému vrcholu vztahuje více odběrů, pak jsou všechny tyto odběry sečteny.

<sup>2</sup>Hodnoty časové charakteristiky jsou odhadnuté, protože nejsou k dispozici reálná data.



Obrázek 7.1: Průběh průměrného vytížení rodinného/bytového domu během dne



Obrázek 7.2: Průběh průměrného vytížení firmy během dne

## 7.4 Výpočet proudů

Pomocí modifikované metody uzlových napětí (popsané níže v 7.4.1) se vypočítá hodnota napětí pro každý vrchol grafu. Poté se projdou všechny hrany a zjišťuje se, zda je napětí v počátečním vrcholu hrany větší než v koncovém. Pokud ano, pak je hrana orientována správně – orientace udává směr proudu. Jinak jsou prohozeny počáteční a koncový vrchol, čímž hrana získá orientaci korespondující se směrem protékajícího proudu.

Pro každou hranu  $e$  se vypočítá velikost proudu  $I_e$  podle vzorce (7.1).

$$I_e = (U_p - U_k) \cdot G_e \quad [A] \quad (7.1)$$

kde

$U$  je napětí ve voltech  $[V]$ ,

$G$  je elektrická vodivost v siemensích  $[S]$ ,

$p$  je počáteční vrchol hrany  $e$ ,

$k$  je koncový vrchol hrany  $e$ .

Vypočtené hodnoty proudu a napětí se uloží do proměnných  $H$  a  $V$ .

### 7.4.1 Modifikovaná metoda uzlových napětí

Modifikovaná metoda uzlových napětí (dále jen MMUN) je popsána v materiálech [10]. MMUN je v tomto případě matice uzlových napětí (MUN) rozšířená o zdroje napětí. Vytvářenou matici budeme značit  $M$ .

Ke zhotovení MUN je potřeba znát vodivosti  $G$  jednotlivých hran. Ty se vypočítají podle vzorce (7.2).

$$G = \frac{1}{R} \quad [S] \quad (7.2)$$

kde

$R$  je elektrický odpor v ohmech  $[\Omega]$ . Velikost odporu vodiče<sup>3</sup> je dána délkou (DELKA) a typem vedení (TYP\_VEDENI) u každé hrany v  $H$ .

Zhotovení MUN:

Prvky  $M_{vv}$  na hlavní diagonále obsahují součet všech vodivostí  $G$ , které jsou připojeny k vrcholu  $v$ . Neboli součet vodivostí všech hran incidujících s vrcholem  $v$ .

Prvky  $M_{uv}$  ( $u \neq v$ ) obsahují záporně vzatý součet všech vodivostí  $G$ , které jsou připojeny mezi vrcholy  $u$  a  $v$ . Tedy součet záporných hodnot vodivostí hran, které propojují dva sousedící vrcholy  $u$  a  $v$ .

Na pravé straně rovnice se na řádky odpovídající odběrným místům přiřadí záporná velikost odběru.

Rozšíření MUN na MMUN:

Rozešíření MUN se zvětší o počet napájecích vrcholů. Do každého nově vzniklého řádku/sloupce se na index odpovídající ID napájecího vrcholu přiřadí jednička a na pravou stranu rovnice je uložena hodnota napětí dodávaného danou trafostanicí.

Nyní každý řádek matice  $M$  odpovídá jedné rovnici. Stačí tedy vyřešit soustavu rovnic. Po výpočtu představují řádky nerozšířené matice  $M$  hodnoty napětí v těchto vrcholech.

## 7.5 Určení sousednosti napájecích vrcholů

Nejprve je nutné vytvořit matici skupin  $S$ . Pro každou trafostanicí se naleznou všechny orientovaně dostupné vrcholy z tohoto napájecího místa. Následně se do logického vektoru (o délce rovnající se počtu všech vrcholů) přiřadí na indexy odpovídajícím orientovaně dostupným vrcholům hodnota `true`.

$$S_{tv} = \begin{cases} \text{true}, & \text{pokud je vrchol } v \text{ napájen trafostanicí } t, \\ \text{false} & \text{v ostatních případech.} \end{cases} \quad (7.3)$$

S pomocí matice  $S$  lze vytvořit matici sousednosti  $I$ .

Dvě trafostanice  $t_1$  a  $t_2$  spolu sousedí právě tehdy, když

$$t_1, t_2 \in N, \exists v \in V, v : \text{ je napájen z } t_1 \wedge \text{ je napájen z } t_2. \quad (7.4)$$

Pro každý vrchol z matice  $S$  se zjistí, ze kterých trafostanic je napájen. Pokud je napájen z více než jedné, pak jsou nalezené trafostanice považovány za sousedící. Tudíž nesmějí nabývat stejných barev.

$$I_{t_1 t_2} = \begin{cases} \text{true}, & \text{pokud spolu trafostanice } t_1 \text{ a } t_2 \text{ sousedí,} \\ \text{false} & \text{v ostatních případech.} \end{cases} \quad (7.5)$$

<sup>3</sup>Hodnoty odporu jsou převzaty z [http://www.nktcables.com/~media/Files/NktCables/Products/CZ/LV%20cables/LV\\_new%20datasheets2014/1-AYKY.pdf](http://www.nktcables.com/~media/Files/NktCables/Products/CZ/LV%20cables/LV_new%20datasheets2014/1-AYKY.pdf).

## 7.6 Implementované algoritmy

Byly implementovány různé typy algoritmů popsaných v kapitolách 4 a 5. Konkrétně jde o algoritmy: sekvenční barvení heuristikou LDO popsané v kapitole 4.1.2, barvení pomocí maximálních nezávislých množin popsané v kapitole 4.2.1, barvení slepováním uzlů heuristikou RLF popsané v kapitole 4.3.2, zjištění barevnosti za pomoci backtrackingu popsané v kapitole 5.2 a vylepšení stávajícího obarvení metodou výměny barev popsané v kapitole 4.4.

## 7.7 Problémy při implementaci a jejich řešení

Implementace algoritmů v Matlabu s sebou přináší řadu komplikací a problémů, které je potřeba vyřešit.

### 7.7.1 Matlab neobsahuje typ množina

U některých implementovaných algoritmů bylo potřeba udržovat informace o tom, zda se některý vrchol nachází v dané množině.

Jedním ze způsobů, jak implementovat množinu v Matlabu, je logický vektor, kde index vektoru značí číslo vrcholu a hodnota *true/false* říká, zda vrchol patří/nepatří do množiny. Z důvodu ušetření paměti lze použít řídké matice.

### 7.7.2 Rozměry vybraných matic

Napájecí vrcholy mohou představovat kterýkoli vrchol, tudíž mohou odkazovat na libovolný záznam (řádek) v matici *Vrcholy*.

Důsledkem je, že jsou *Matice skupin*, *Matice sousednosti* a vektor *Barvy* vytvořeny jako řídké matice s počtem řádků (popř. sloupců) odpovídajícím počtu všech vrcholů. Reálně je však vyplněno jen tolik řádků (popř. sloupců), kolik je trafostanic. Kdyby nebyly použity řídké matice, velmi by se zvýšila paměťová náročnost programu.

### 7.7.3 Vytvoření barevného schématu

Vznikla potřeba programově vytvořit barevné schéma v RGB modelu s určitým počtem barev tak, aby byly rozdíly mezi jednotlivými barvami co největší a tudíž bylo možno jednu od druhé dobře rozlišit. Navíc se nesmějí použít barvy vyhrazené pro jiné účely (konkrétně černá, bílá a červená).

Barevný model RGB nenabízí snadné řešení tohoto problému, ovšem model HSV je pro rozdělení na několik částí vhodný. Výběr černé a bílé barvy se eliminuje ponecháním složek sytosti (Saturation) a jasu (Value) na maximum. Jediné, s čím se bude pracovat, je tedy barevný tón (Hue). Barevný tón si lze představit jako interval, ve kterém se určí body tak, aby se jejich počet rovnal požadovanému počtu barev a jejich vzdálenost mezi sebou (a krajními body intervalu) byla stejná. Každý takto vytvořený bod poté udává hodnotu barevného tónu jedné z barev, přičemž krajní body intervalu představují vyhrazenou červenou barvu. Nyní jen stačí převést barevný model HSV na RGB.

### 7.7.4 Slévání proudů

Existují místa v rozvodné síti, kde dochází ke slévání a následnému průtoku více proudů. Tuto skutečnost je pro větší přehlednost nutno graficky odlišit. U vrcholů napájených z více trafostanic je proto nutné upravit barvu.

Barva pro takovýto vrchol je určena jako součet barev všech trafostanic napájejících daný vrchol podělený jejich počtem. Formálně zapsáno

$$\frac{\sum \text{RGB barev trafostanic, které napájejí daný vrchol}}{\text{počet takovýchto trafostanic}}. \quad (7.6)$$

Nové barvy tedy vznikají mícháním původních barev. Navíc jsou hrany vycházející z takovýchto vrcholů nejen barveny vypočtenou barvou, ale i zobrazovány čárkovaně (místo běžné plné čáry).

Pro vysvětlení bude nejlepší příklad. Vrchol je napájený ze dvou trafostanic, které mají barvu modrou a zelenou. Výsledná barva bude

$$\frac{[0, 1, 0](\text{zelená}) + [0, 0, 1](\text{modrá})}{2} = [0, 0.5, 0.5],$$

což v RGB modelu odpovídá žluté barvě.

### 7.7.5 Činný a jalový proud

V praxi je dobré vidět jak jsou jednotlivé složky proudu vůči sobě posunuty. Proto jsou místo jednoho vykreslovaného grafu, který představoval činný proud, vykreslovány do jednoho okna grafy dva (pro zobrazení každé složky zvlášť, přehledně vedle sebe).

Kvůli tomu jsou téměř všechny proměnné zdvojeny. Jedna vždy představuje hodnoty pro činný a druhá pro jalový proud.

Než byl vznesen požadavek na zobrazení jalového proudu, tak se počítalo pouze s činným. Nyní bylo nutné navíc vypočítat odběry, ze kterých se následně počítá rozložení jalového proudu v síti.

Řešením tohoto problému je určit hodnoty jalového odběru z již vypočteného odběru činného. Tedy pro každý odběrový vrchol se náhodně zvolí hodnota  $\varphi$  tak, aby platilo  $0, 9 < \cos \varphi < 1$ . K výpočtu jalového odběru  $Q_J$  se použije vzorec (7.7).

$$Q_J = Q_C \cdot \tan \varphi \quad (7.7)$$

kde  $Q_C$  je velikost činného odběru.

Všechny další výpočty se pak provádějí zvlášť pro činný a zvlášť pro jalový proud.

### 7.7.6 Výpočetní náročnost backtrackingu

Algoritmus backtracking umí získat optimální obarvení, ale jeho nalezení u větších grafů mnohdy trvá neúnosně dlouho.

Proto se vždy od nalezení lepšího než stávajícího obarvení se začne počítat počet návratů. Když tento počet přesáhne určenou mez, je algoritmus ukončen a vrací doposud nejlepší nalezené obarvení grafu.

## Kapitola 8

# Zhodnocení barvicích algoritmů

Jako testovací data poslouží graf el. rozvodné sítě NN Slavkova u Brna, jehož vytvoření je popsáno v kapitole 6. Jelikož je výroba takového grafu časově náročná, bylo při vytváření větších grafů, které simulují lidnatější města, využito multiplikace Slavkovů a jejich propojení vedením. Naopak při výrobě malých grafů se použila jen část města.

Pro každou testovanou velikost grafu (města) se vytvoří více variant. Jednotlivé varianty se mohou lišit v tom, které trafostanice spolu sousedí.

U každého testovaného grafu se vypočte podle vzorce (8.1) tolik hodnot času, kolik se vytváří variant daného grafu.

$$t_i = (i - 1) \cdot \frac{24}{a - 1} \quad \text{pro } i = \{1, 2, \dots, a - 1, a\} \quad (8.1)$$

kde

$a$  je počet vytvářených variant,

$i$  je číslo varianty.

Pro každou variantu  $i$  se vypočítají hodnoty odběrů v čase  $t_i$  (popsáno v kapitole 7.3), dále se určí velikost a směr protékajících proudů (viz kapitola 7.4) a sousednost jednotlivých trafostanic (popsáno v kapitole 7.5). Následně se provede obarvení všemi porovnávanými metodami a u jednotlivých metod se uloží délka jejich výpočtu a počet barev použitých k obarvení. Takto získaná data jsou dále využita při vytváření obrázků.

Ke grafickému porovnání implementovaných metod z hlediska počtu použitých barev a délky výpočtu jsou použity krabicové grafy<sup>1</sup>, kde červená čára značí medián, modrý obdélník vyznačuje první a třetí kvartil (25. a 75. percentil), černé čárkované čáry určují minimum a maximum hodnot (nepovažovaných za extrémy) a červené křížky udávají extrémní hodnoty.

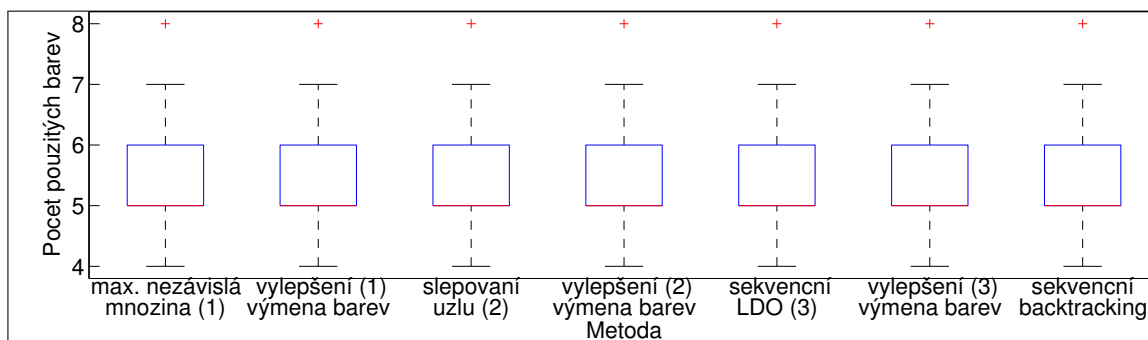
Je nemožné vylepšit optimální obarvení nalezené pomocí backtrackingu. Proto je nesignifikantní se jej pokoušet vylepšit. Z toho důvodu nejsou v obrázcích zobrazeny výsledky pro vylepšení obarvení backtrackingu metodou výměny barev.

### Graf č.1

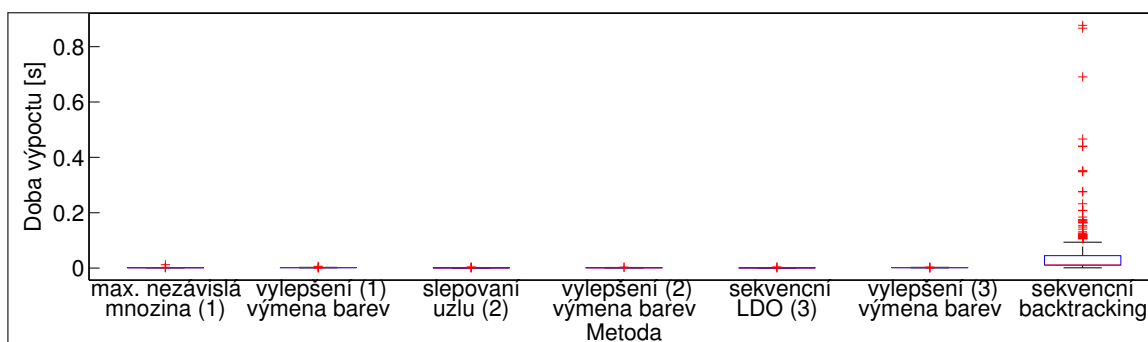
Odpovídá obci s přibližně 2 000 obyvateli. Graf obsahuje 774 vrcholů, z nichž je 443 odběrových, 12 představuje trafostanice a 319 spojovací místa. Jednotlivé vrcholy jsou spojeny 822 hranami. Při testování bylo vytvořeno 5 000 variant tohoto grafu.

<sup>1</sup>Více o použitých krabicových grafech (boxplot) v nápovědě *MATLABu*.

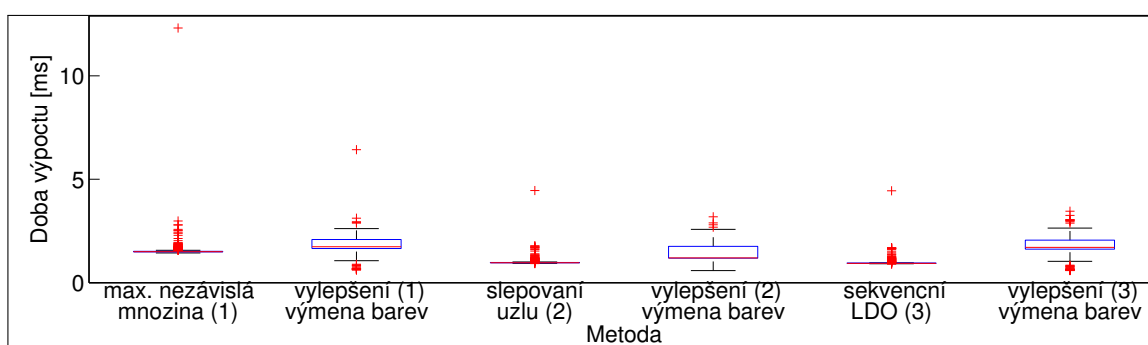
Na obrázku 8.1 je vidět porovnání testovaných metod z hlediska počtu barev použitých k obarvení grafu a na obrázcích 8.2 a 8.3 je zobrazeno porovnání délky výpočtu obarvení jednotlivých metod. Jelikož backtracking trvá oproti ostatním metodám řádově déle, nelze na obrázku 8.2 porovnat všechny metody. Proto je zde i obrázek 8.3, na kterém jde vidět porovnání délky výpočtu všech testovaných metod kromě backtrackingu.



Obrázek 8.1: Porovnání metod z hlediska počtu barev (s backtrackingem) – graf č.1



Obrázek 8.2: Porovnání metod z hlediska doby výpočtu (s backtrackingem) – graf č.1



Obrázek 8.3: Porovnání metod z hlediska doby výpočtu – graf č.1

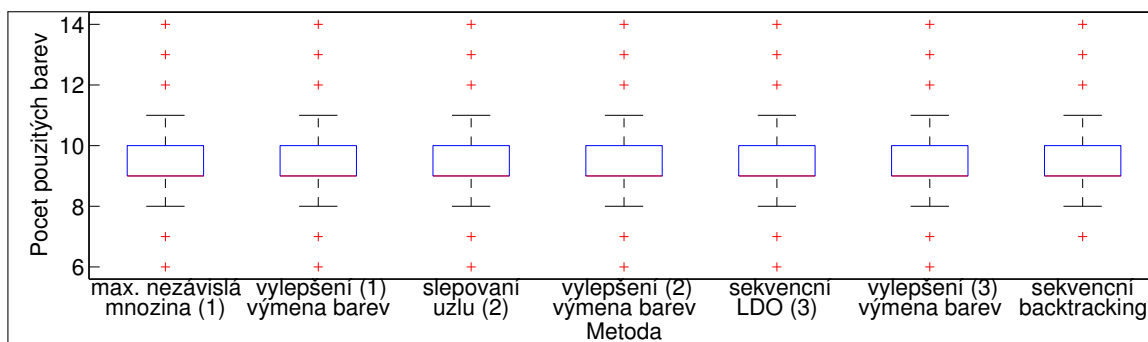
## Graf č.2

Odpovídá malému městu s přibližně 5 000 obyvateli. Graf obsahuje 2 377 vrcholů, z nichž je 1 386 odběrových, 34 představuje trafostanice a 957 spojovací místa. Jednotlivé vrcholy

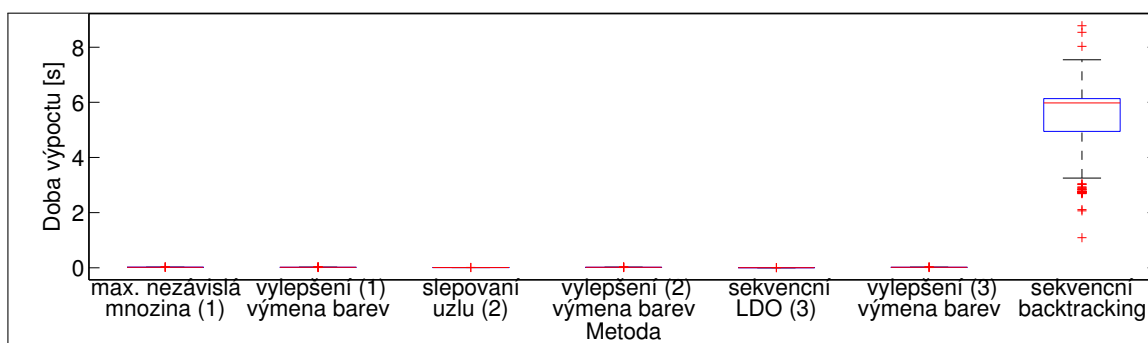


jsou spojeny 2524 hranami. Při testování bylo vytvořeno 1 000 variant tohoto grafu.

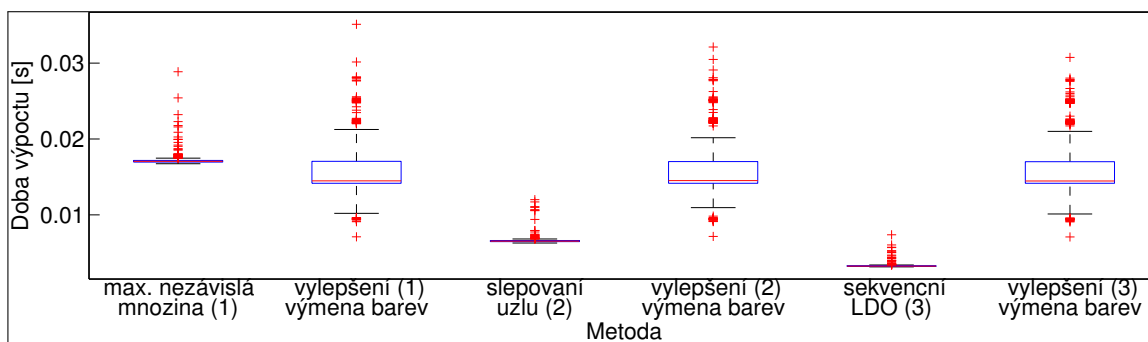
Na obrázku 8.4 je zobrazeno porovnání testovaných metod z hlediska počtu barev použitých k obarvení grafu. K porovnání doby výpočtu slouží obrázky 8.2 a 8.3.



Obrázek 8.4: Porovnání metod z hlediska počtu barev (s backtrackingem) – graf č.2



Obrázek 8.5: Porovnání metod z hlediska doby výpočtu (s backtrackingem) – graf č.2



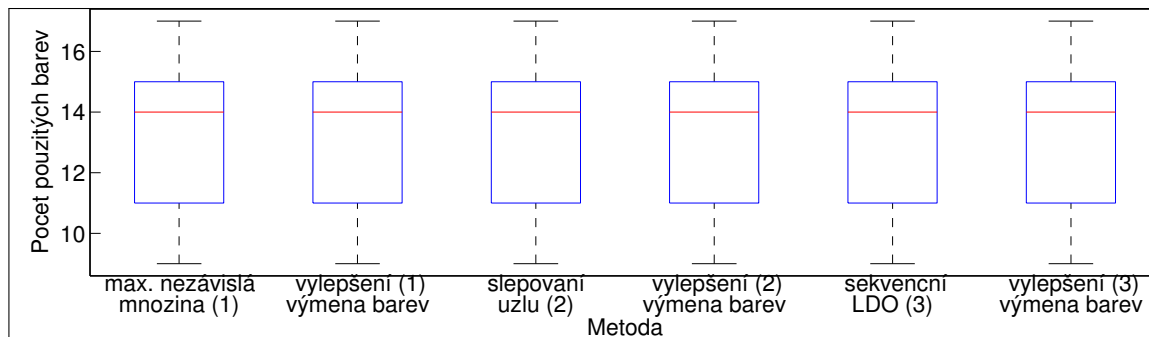
Obrázek 8.6: Porovnání metod z hlediska doby výpočtu – graf č.2

Výpočet obarvení pomocí backtrackingu u variant tohoto grafu trval příliš dlouho a proto byl algoritmus vždy po určeném počtu návratů předčasně ukončen. Z toho důvodu backtracking vždy nevrátil optimální obarvení – tedy takové, při kterém se počet barev použitých k obarvení rovná chromatickému číslu. Proto je možné na obrázku 8.4 vidět, že minimální hodnota extrému při použití backtrackingu je vyšší než u ostatních metod.

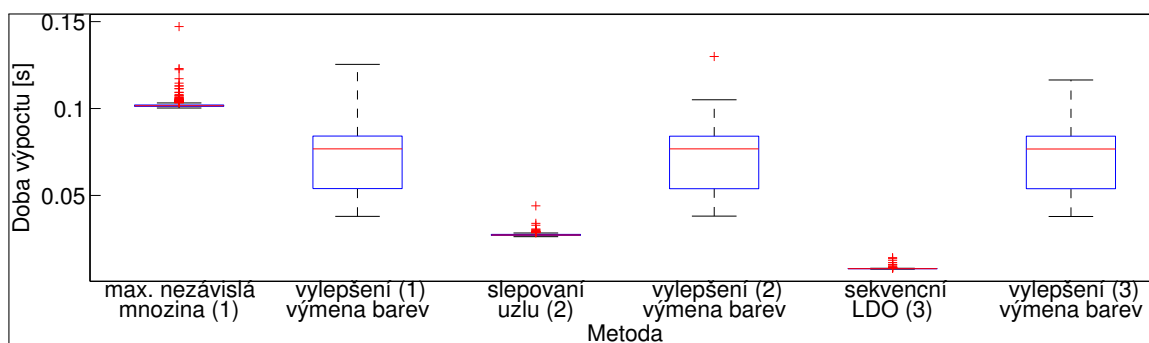
Protože by výpočty backtrackingu pro větší grafy byly rovněž předčasně ukončeny a tudíž by nebyly relevantní, nebudou již tyto hodnoty do dalších obrázků zanašeny.

### Graf č.3

Odpovídá městu s přibližně 10 000 obyvateli. Graf obsahuje 4 734 vrcholů, z nichž je 2 759 odběrových, 62 představuje trafostanice a 1 913 spojovací místa. Jednotlivé vrcholy jsou spojeny 5 038 hranami. Při testování bylo vytvořeno 1 000 variant tohoto grafu.



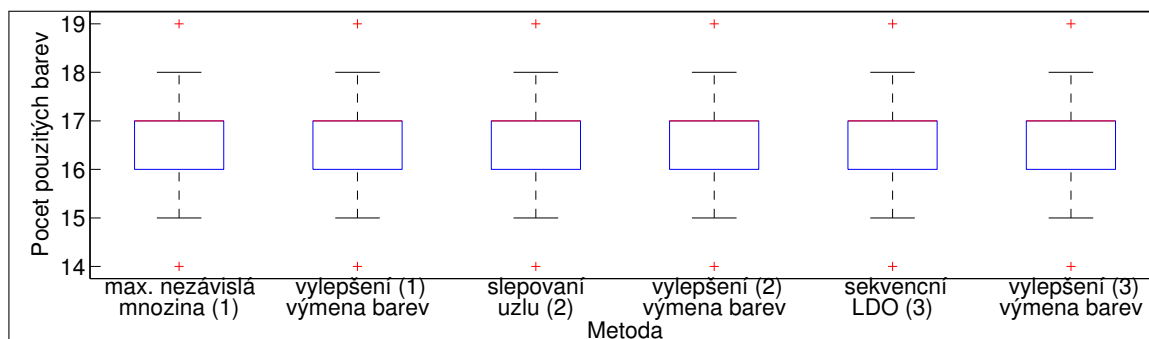
Obrázek 8.7: Porovnání metod z hlediska počtu barev – graf č.3



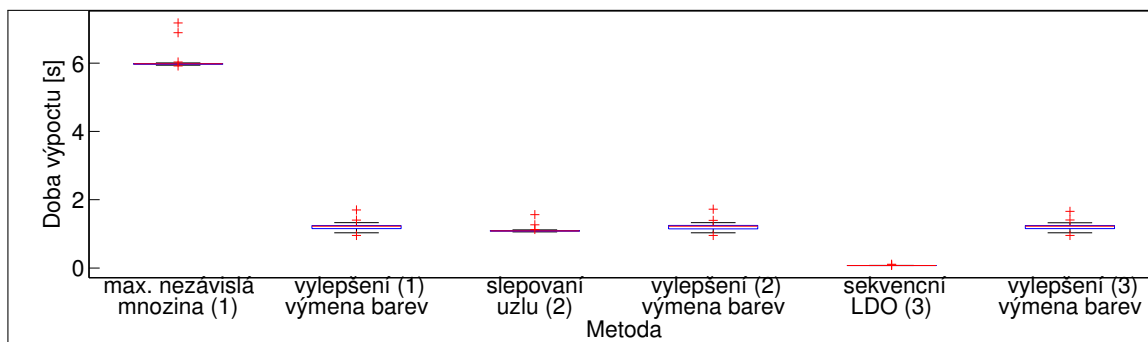
Obrázek 8.8: Porovnání metod z hlediska doby výpočtu – graf č.3

### Graf č.4

Odpovídá městu s přibližně 40 000 obyvateli. Graf obsahuje 19 016 vrcholů, z nichž je 11 093 odběrových, 251 představuje trafostanice a 7 672 spojovací místa. Jednotlivé vrcholy jsou spojeny 20 256 hranami. Při testování bylo vytvořeno 50 variant tohoto grafu.



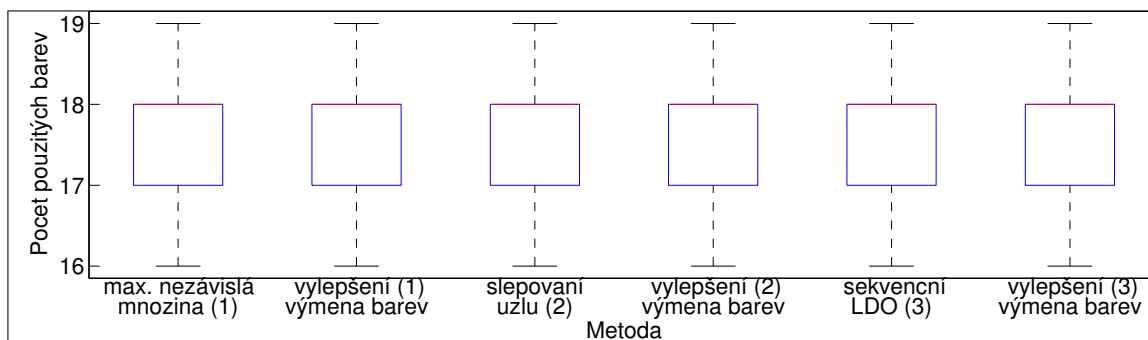
Obrázek 8.9: Porovnání metod z hlediska počtu barev – graf č.4



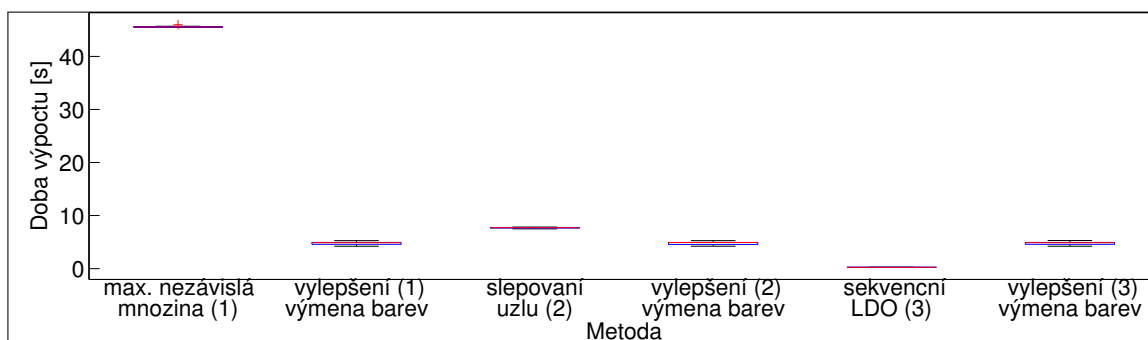
Obrázek 8.10: Porovnání metod z hlediska doby výpočtu – graf č.4

### Graf č.5

Odpovídá velkému městu s přibližně 80 000 obyvateli. Graf obsahuje 37 946 vrcholů, z nichž je 22 136 odběrových, 496 představuje trafostanice a 15 314 spojovací místa. Jednotlivé vrcholy jsou spojeny 40 431 hranami. Při testování bylo vytvořeno 10 variant tohoto grafu.



Obrázek 8.11: Porovnání metod z hlediska počtu barev – graf č.5



Obrázek 8.12: Porovnání metod z hlediska doby výpočtu – graf č.5

## 8.1 Vyhodnocení

Prvním hodnoceným kritériem je počet barev, kterými graf obarví daná metoda. Z obrázků 8.1, 8.4, 8.7, 8.9 a 8.11 je zřejmé, že nenastaly žádné velké rozdíly mezi jednotlivými metodami v počtu barev použitých k obarvení grafu. Ani algoritmus výměny barev nedokázal vylepšit obarvení u dostatečného množství variant tak, aby to bylo možno vyčíst z obrázků.

Dalším kritériem je rychlost jednotlivých algoritmů. Na obrázcích 8.2, 8.3, 8.5, 8.6, 8.8, 8.10 a 8.12 je jasně vidět, že nejrychlejší je sekvenční barvení (heuristika LDO), zvolna následované algoritmem slepování uzlů (heuristika RLF), o něco pomalejší pak je algoritmus barvení pomocí maximální nezávislé množiny a jako nejpomalejší vyšel z tohoto srovnání algoritmus zpětného navracení (backtracking).

Z výsledků testování vyplývá, že pro obarvování grafů (reprezentujících elektrické rozvodné sítě nízkého napětí) je nejlepší algoritmus sekvenčního barvení heuristikou LDO, protože oproti ostatním metodám obarví grafy několikanásobně rychleji a to za použití stejného počtu barev.

# Kapitola 9

## Závěr

Práce popisuje algoritmy pro vrcholové barvení grafu a pojmy s tím související. Bylo implementováno 5 metod, kde každá představuje určitou skupinu barvicích algoritmů. Barvení využívající backtrackingu je příklad algoritmu, který obarví graf a nalezne jeho chromatické číslo. Metoda výměny barev zase pracuje nad již obarvenými grafy a snaží se vylepšit jejich obarvení. Zbývající tři algoritmy (sekvenční barvení heuristikou LDO, barvení pomocí maximálních nezávislých množin, barvení slepováním uzlů) představují ty nejčastěji používané.

Výsledkem bakalářské práce je program s grafickým uživatelským rozhraním a jeho zjednodušená terminálová verze. V práci je popsán návrh a implementace programu vytvořeného v systému *MATLAB*. Implementovaný program obarví všechny trafostanice (a proudy z nich vytékající) tak, aby dvě spolu sousedící trafostanice neměly stejnou barvu. K barvení se využívá vždy jeden z výše popsaných algoritmů.

Program má implementované grafické uživatelské rozhraní, v kterém lze pomocí množství tlačítek a přepínačů nastavovat různé chování. Výsledný obarvený graf je zobrazován v novém okně. Je vytvořena i verze bez grafického rozhraní, která pouze vypočítá obarvení grafu danou metodou a výsledek uloží. Program je popsán v příloze A (uživatelská příručka).

Při porovnávání implementovaných algoritmů na grafech představujících elektrickou distribuční síť města Slavkova u Brna vyšel jako nejvhodnější algoritmus sekvenčního barvení heuristikou Largest Degree Ordering, který k obarvení grafu využil stejný počet barev jako ostatní testované algoritmy, ale oproti nim byla jeho časová náročnost výrazně nižší.

Jelikož tento projekt na žádný dřívější nenavazuje, je spousta prostoru pro jeho zdokonalení. Níže je pár návrhů na vylepšení.

Do budoucna by se dalo implementovat více algoritmů pro vrcholové barvení a případně zrychlit ty stávající. Nyní lze graf načítat a ukládat pouze jako proměnné *MATLABu* (načítat lze i z formátu *GraphML*), proto by bylo vhodné přidat další formáty (formát CSV, aj.), ze/do kterých bude možno načítat/ukládat grafy.

Také by se dala rozšířit funkčnost programu přidáním funkce, která by místo všech prvků rozvodné sítě zobrazila pouze napájecí vrcholy (trafostanice) a hrany, které představují jejich sousednost. To by mohlo vést k jinému úhlu pohledu a pochopení průběhu barvení dané metody.

Je tu i možnost přidat do grafického uživatelského rozhraní další ovládací prvky umožňující např. zapínat/vypínat různé popisky jednotlivých prvků rozvodné sítě apod.

# Literatura

- [1] Dannhoferová, J.; Foltýnek, T.: Teorie Grafů. [online], [cit. 2016-4-28].  
URL <https://is.mendelu.cz/eknihovna/opory/index.pl?cast=22289>
- [2] Demel, J.: *Grafy a jejich aplikace*. Academia, 2002, 257 s., ISBN 80-200-0990-6.
- [3] Diestel, R.: Graph theory. [online], 2010 [cit. 2016-1-6], ISBN 978-3-642-14278-9.  
URL <http://diestel-graph-theory.com/>
- [4] Dutton, R. D.; Brigham, R. C.: A new graph colouring algorithm. In *The Computer Journal*, ročník 24, číslo 1, 1981, s. 85–86.  
URL <http://comjnl.oxfordjournals.org/content/24/1/85.full.pdf>
- [5] Kolář, J.: Teoretická informatika. [online], 2004 [cit. 2016-1-3], ISBN 80–900853–8–5.  
URL <http://www.exfort.org/20051/4/x36tin/pdf/ti.pdf>
- [6] Mansuri, A.; Gupta, V.; Chandel, R. S.: Coloring Programs in Graph Theory. In *Int. Journal of Math. Analysis*, ročník 4, číslo 50, 2010, s. 2473–2479.  
URL <http://m-hikari.com/ijma/ijma-2010/ijma-49-52-2010/mansuriIJMA49-52-2010.pdf>
- [7] Mareš, M.; Valla, T.: Časová a pomětová složitost. [online], 2014-01-24 [cit. 2016-1-13], URL <http://mj.ucw.cz/vyuka/ads/02-complex.pdf>.
- [8] McHugh, J. A.: *Algorithmic Graph Theory*. New Jersey: Prentice-Hall, 1990, 327 s., ISBN 0-13-023615-2.
- [9] Palubeckis, G.: On the recursive largest first algorithm for graph colouring. In *International Journal of Computer Mathematics*, ročník 85, číslo 2, 2008-2, s. 191–200, URL <http://www.tandf.co.uk/journals>.
- [10] Sedláček, J.; Steinbauer, M.: Analýza obvodů – univezální metody. [online], [cit. 2016-4-26], URL [http://www.utee.feec.vutbr.cz/files/predmety/BEL1/Multimed\\_uc/BEL1\\_blok3.pdf](http://www.utee.feec.vutbr.cz/files/predmety/BEL1/Multimed_uc/BEL1_blok3.pdf).
- [11] Večerka, A.: Grafové algoritmy. [online], 2007 [cit. 2016-4-20], URL [https://phoenix.inf.upol.cz/esf/ucebni/Grafy\\_a\\_grafove\\_algoritmy.pdf](https://phoenix.inf.upol.cz/esf/ucebni/Grafy_a_grafove_algoritmy.pdf).
- [12] Šeda, M.: Teorie grafů. [online], 2003-11 [cit. 2016-1-14].  
URL [http://www.uai.fme.vutbr.cz/~mseda/TG03\\_MS.pdf](http://www.uai.fme.vutbr.cz/~mseda/TG03_MS.pdf)
- [13] Šišma, P.: Problém čtyř barev. In *Historie matematiky*, Praha: Prometheus, 1995, s. 169–180, URL <http://dml.cz/dmlcz/401043>.

# Přílohy

## Seznam příloh

<b>A</b>	<b>Uživatelská příručka</b>	<b>45</b>
A.1	Terminálové rozhraní . . . . .	45
A.2	Grafické uživatelské rozhraní . . . . .	46
A.2.1	Získání/načtení/uložení dat . . . . .	46
A.2.2	Nastavení průběhu výpočtu . . . . .	46
A.2.3	Nastavení grafického zobrazení . . . . .	47
A.2.4	Metoda barvení . . . . .	47
A.2.5	Metoda vylepšení obarvení . . . . .	47
A.2.6	Menu . . . . .	48
<b>B</b>	<b>Jednoduchý příklad grafu</b>	<b>49</b>



# Příloha A

## Uživatelská příručka

### A.1 Terminálové rozhraní

Umožňuje vypočítat obarvení grafu vybranou metodou bez grafické prezentace výsledků. V prostředí *Matlab* se program spouští příkazem `terminal(parametry)`, kde `parametry` jsou dvojice (s výjimkou parametru `GUI`): `název parametru`, `hodnota parametru` následované dalšími `parametry`. Význam jednotlivých parametrů je popsán v tabulce [A.1](#).

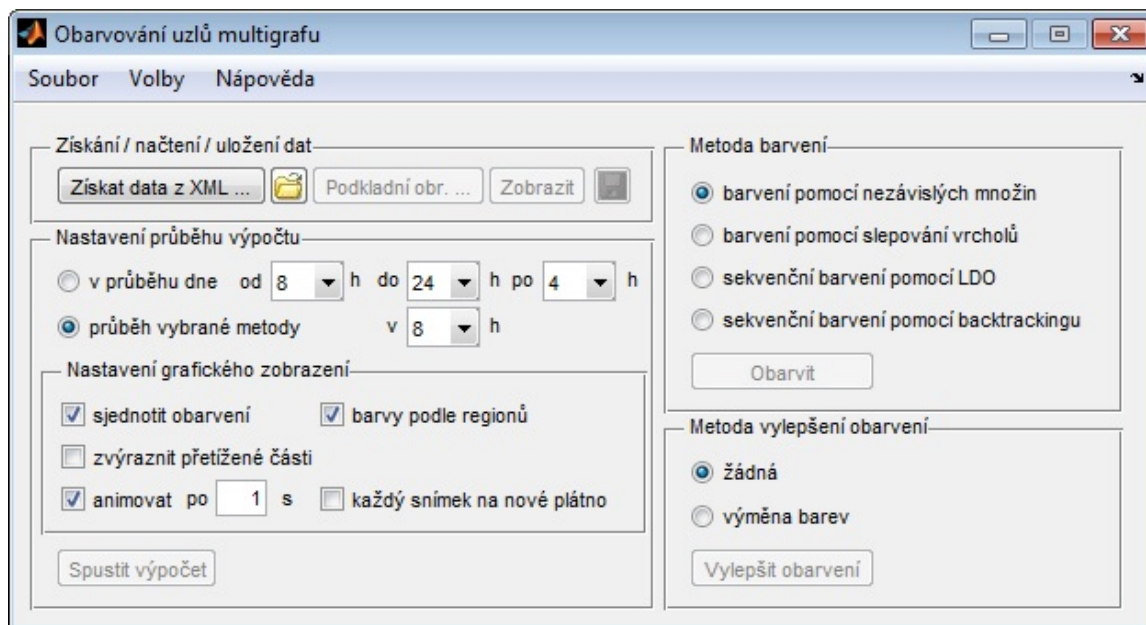
Název parametru	Hodnota parametru	Popis
<code>gui</code>	bez hodnoty	spustí grafické uživatelské rozhraní (viz <a href="#">A.2</a> )
<code>vstup_xml</code>	jméno souboru	jméno XML souboru ( <code>graphml</code> ), ze kterého se získá graf default: " (žádný soubor)
<code>vstup</code>	jméno souboru	jméno souboru ( <code>mat</code> ) s uloženými proměnnými default: " (žádný soubor)
<code>vystup</code>	jméno souboru	jméno souboru ( <code>mat</code> ) kam se uloží proměnné default: " (žádný soubor – výsledek se neuloží)
<code>meritko</code>	$0 < \text{meritko}$	kolik metrů představuje jedna jednotka ve vstupním souboru default: 1 (1 jednotka představuje 1 metr ve skutečnosti)
<code>cas</code>	$0 \leq \text{cas} \leq 24$	pro jaký čas (v kolik hodin) se má počítat (simulovat) odběr default: 8 (výpočet pro osmou hodinu ranní)
<code>metoda</code>	<code>sekvenčni_LDO</code> <code>slepovani_uzlu</code> <code>max_mnozina</code> <code>backtracking</code>	určuje metodu použitou k obarvení grafu default: <code>sekvenčni_LDO</code> sekvenční barvení heuristikou Largest Degree Ordering barvení slepováním uzlů barvení pomocí maximální nezávislé množiny sekvenční barvení s využitím <code>backtrackingu</code>
<code>vylepseni</code>	<code>zadne</code> <code>vymena_barev</code>	určuje metodu použitou k vylepšení obarvení grafu default: <code>zadne</code> nesnaží se nijak vylepší obarvení grafu vylepšuje obarvení grafu metodou výměny barev

Tabulka A.1: Popis parametrů

Po vypočítání obarvení je vypsána zpráva obsahující informace o počtu barev, době výpočtu, počtu vrcholů (všech, odběrových a napájecích) a počtu hran (všech a hran určujících sousednost trafostanic).

## A.2 Grafické uživatelské rozhraní

Spouští se příkazem `GUI` nebo `terminal('GUI')` v příkazovém okně programu *Matlab*.



Obrázek A.1: Grafické uživatelské rozhraní

V následujících sekcích jsou popsány jednotlivé části grafického uživatelského rozhraní.

### A.2.1 Získání/načtení/uložení dat

`Získat data z XML ...` – slouží k získání dat z *XML* souboru s koncovkou *.graphml*. Pro výběr souboru se otevře dialogové okno.

`Podkladní obr. ...` – slouží k načtení dříve uložených dat ze souboru s koncovkou *.mat*. Pro výběr souboru se otevře dialogové okno.

`Zobrazit` – slouží k načtení podkladního obrázku (koncovka *.jpg*), který se bude zobrazovat jako pozadí pod grafem. Pro výběr obrázku se otevře dialogové okno.

`Zobrazit` – na plátně zobrazí aktuální graf.

`Uložení` – slouží k uložení dat do souboru s koncovkou *.mat*. Pro výběr souboru se otevře dialogové okno.

### A.2.2 Nastavení průběhu výpočtu

`v průběhu dne` určuje, že se bude animovat průběh dne (v určených hodinách).

`průběh vybrané metody` určuje, že se bude animovat postup barvení vybrané metody.

`Spustit výpočet.` Po stisku se vypočítá rozložení el. proudů v grafu a obarví graf metodou určenou v panelu *Metoda barvení* (A.2.4) a *Metoda lokálního vylepšení* (A.2.5).

### A.2.3 Nastavení grafického zobrazení

**sjednotit obarvení** – pokud je zaškrtnuto, pak je jedena část plátna (přebarvená) přebarvena podle druhé (původní). Jako původní se vybere ta, na jejíž obarvení byl použit větší počet barev. Pokud mají obě části (činná i jalová) stejný počet barev, tak se přebarvuje jalová podle činné. Přebarvená část plátna má osy a popisky šedě, zatímco původní část má osy a popisky černě. Při této volbě může nastat u přebarvené části situace, kdy spolu sousedí dvě trafostanice obarvené stejnou barvou.

**barvy podle regionů** – pokud je zaškrtnuto, pak je graf barven podle regionů tak, že barvy trafostanic v jednom regionu jsou barveny blízkými barvami (jednou částí barevného spektra). Jsou nadefinována 4 barevná schémata (oranžová, zelená, modrá, fialová). Trafostanice, které nemají přidělen region jsou barveny černě.

Pokud zaškrtnuto není, je používán celý rozsah barevného spektra na všechny trafostanice. Je předdefinováno 10 barev. Pokud tento počet nestačí, jsou další trafostanice barveny černě.

**zvýraznit přetížené části** – pokud je zaškrtnuto, tak obarvuje hrany grafu (vedení), kterými protéká více proudu než je povolená maximální hodnota, červenou barvou.

**animovat** – pokud je zaškrtnuto, pak se animuje průběh dne/metody a mezi jednotlivými snímky je ponechána nastavená časová mezera.

Pokud zaškrtnuto není, pak se obarvení vypočítá, ale není automaticky zobrazeno (graf lze zobrazit pomocí ).

**každý snímek na nové plátno** – pokud je zaškrtnuto, pak se každá aktualizace plátna neprojeví změnou obarvení na aktuálním plátně, ale zobrazí se na novém plátně.

### A.2.4 Metoda barvení

**barvení pomocí nezávislých množin** – barvení grafu pomocí maximálních nezávislých množin.

**barvení pomocí slepování vrcholů** – barvení slepováním uzlů, heuristika RLF (Recursive Largest First).

**sekvenční barvení pomocí LDO** – sekvenční barvení, heuristika LDO (Largest Degree Ordering).

**sekvenční barvení pomocí backtrackingu** – sekvenční barvení s využitím backtrackingu.

– spustí barvení grafu vybranou metodou. U barvení se projeví pouze nastavení

**animovat** a  **každý snímek na nové plátno**.

### A.2.5 Metoda vylepšení obarvení

**žádná** – nepoužije žádný algoritmus.

**výměna barev** – algoritmus výměny barev.


– začne vylepšovat obarvení grafu vybranou metodou. U barvení se projeví pouze nastavení  **animovat** a  **každý snímek na nové plátno**.

## A.2.6 Menu

### Soubor

Získat data (.graphml) ... – stejné jako Získat data z XML ... (popsáno v kapitole A.2.1).

Načíst data (.mat) ... – stejné jako  (popsáno v kapitole A.2.1).

Uložit data (.mat) ... – stejné jako  (popsáno v kapitole A.2.1).

Zavřít – zavře poslední vykreslované plátno a grafické uživatelské rozhraní

### Volby

Přidat podkladní obrázek ... – stejné jako Podkladní obr. ... (popsáno v kapitole A.2.1).

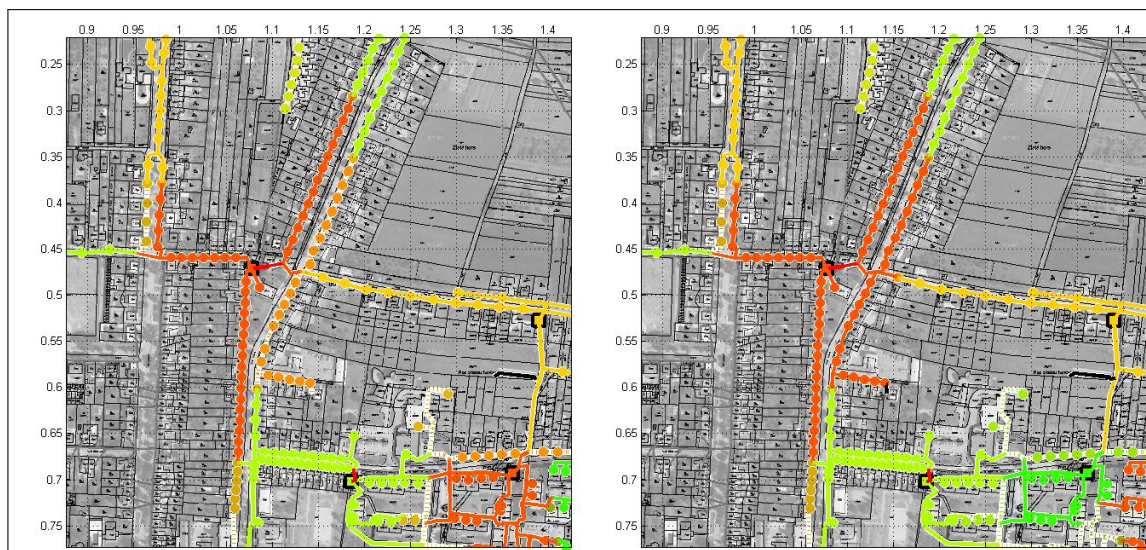
Smazat podkladní obrázek – odstraní podkladní obrázek. Graf se bude zobrazovat na bílém pozadí. Změna se projeví až na novém plátně.

Nastavení měřítka ... – umožňuje změnit měřítko grafu v dialogovém okně.

### Nápověda

Nápověda – vypíše nápovědu k programu.

O programu – vypíše stručné informace o původu programu.

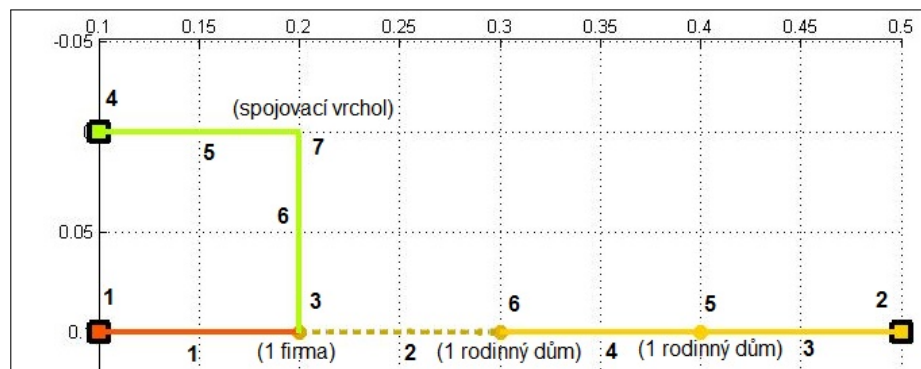


Obrázek A.2: Zobrazení části grafu (vlevo – činný proud, vpravo – jalový proud)

## Příloha B

# Jednoduchý příklad grafu

V této příloze jsou zobrazeny hodnoty jednotlivých proměnných pro graf (část grafu s činnou složkou) zobrazený na obrázku B.1. Čísla u vrcholů a hran udávají jejich ID.



Obrázek B.1: Jednoduchý graf – pro ukázkou proměnných

$$V = \begin{pmatrix} 100 & 100 & 230,000 & 1 \\ 500 & 100 & 230,000 & 2 \\ 200 & 100 & 229,489 & 0 \\ 100 & 0 & 230,000 & 3 \\ 400 & 100 & 229,724 & 0 \\ 300 & 100 & 229,457 & 0 \\ 200 & 0 & 229,744 & 0 \end{pmatrix} \quad B = \begin{pmatrix} 1 \\ 2 \\ 0 \\ 3 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad H = \begin{pmatrix} 1 & 3 & 100 & 1 & 40,890 \\ 3 & 6 & 100 & 1 & 2,580 \\ 2 & 5 & 100 & 1 & 22,081 \\ 5 & 6 & 100 & 1 & 21,390 \\ 4 & 7 & 100 & 1 & 20,445 \\ 7 & 3 & 100 & 1 & 20,445 \end{pmatrix}$$

$$S = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad I = \begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$N = \begin{pmatrix} 1 \\ 2 \\ 4 \end{pmatrix} \quad o = \begin{pmatrix} 3 & 1 & 9 & 58,756 \\ 5 & 1 & 1 & 0,692 \\ 6 & 1 & 1 & 23,969 \end{pmatrix}$$