

O JEDNÉ MAPĚ NA ZAKÁZKU ČÁST 2: POČET OBARVENÍ n BARVAMI

PAVEL STRÍŽ (CZ)

Abstrakt. Článek počítá počet n -obarvení v daném grafu, chceme-li mapě krajů či okresů. Vedle běžných 4 barev užitých v teorému čtyř barev jsem dopočítal n -obarvení pro přípustné barvy. Dvacetidvouuzlový graf byl stále ještě výpočetně přijatelný.

Klíčová slova. Optimalizace, R, Sage, Python, Picat, OpenStreetMap.

ABOUT PREPARATION OF ONE MAP PART 2: THE NUMBER OF n -COLOURINGS

Abstract. The article computes number of n -colourings in a given map/graph. Next to four colours used in the Four Colour Theorem, I tried to use all possible number of colours and compute its number of n -colourings. 22-node graph was still computable.

Keywords. Optimization, R, Sage, Python, Picat, OpenStreetMap.

1. Zpět k R

Rovnou naváží na první část. Při experimentech s programy pro lineární programování kolem výpočtů v sudoku mně vrtalo hlavou, jak v R nechat spočítat model ILP. Knihovna `lp_solve` mě moc nezaujala, při rozhození modelu do sloupců se člověk začne ztrácet. Načtení `.lp` souboru je závislé od jeho formátu. Hledal jsem proto jinou cestu, když mi to na první ani druhou dobrou nejelo.

Masívně užívám program Picat, <https://picat-lang.org/>, kde se lze mezi MIP, SAT, SMT a CP přepínat. V první části tohoto článku je ukázka generování přes Python. Přepínám se z `import sat` na `import mip`. V pozadí se uloží soubor `__tmp.lp`. Ten ještě R neumí zpracovat, resp. jsem cestu nenašel, použil jsem následující udělatko.

V Pythonu přes `pyscipopt` přeukládám do formátu FreeMPS.

```
from pyscipopt import Model
scip = Model()
scip.readProblem(filename="__tmp.lp")
scip.writeProblem(filename="pokonverzi.mps",
                  trans=False, genericnames=False)
```

Takový soubor si lze v R načíst a například přes NEOS server nechat spočítat.

```
library("ROI")
library(ROI.plugin.lpsolve)
```

```
library(ROI.plugin.neos)
(x <- ROI.plugin.lpsolve::read.lp("pokonverzi.mps", type="freemps"))
opt <- ROI_solve(x, solver="neos", method="scip", email="Email-na-NEOSu")
opt; solution(opt)
```

Přebereme výsledky a vykreslíme přes `ggplot2`. Naznačím jádro. Pracovně jsem si okresy zpracoval ze surových dat v bočním souboru. Z výstupů jsem si vybral potřebné údaje a nechal vykreslit. Grafický náhled viz první část tohoto příspěvku.

```
library(ggredist)
library(ggplot2)
library(sf)
source("stefanovi-data.R")
mal <- st_sf(geom=malseznam)
malshort<-solution(opt)[2:150]
ggplot(mal,aes(fill = malshort))+geom_sf()+theme_map()
```

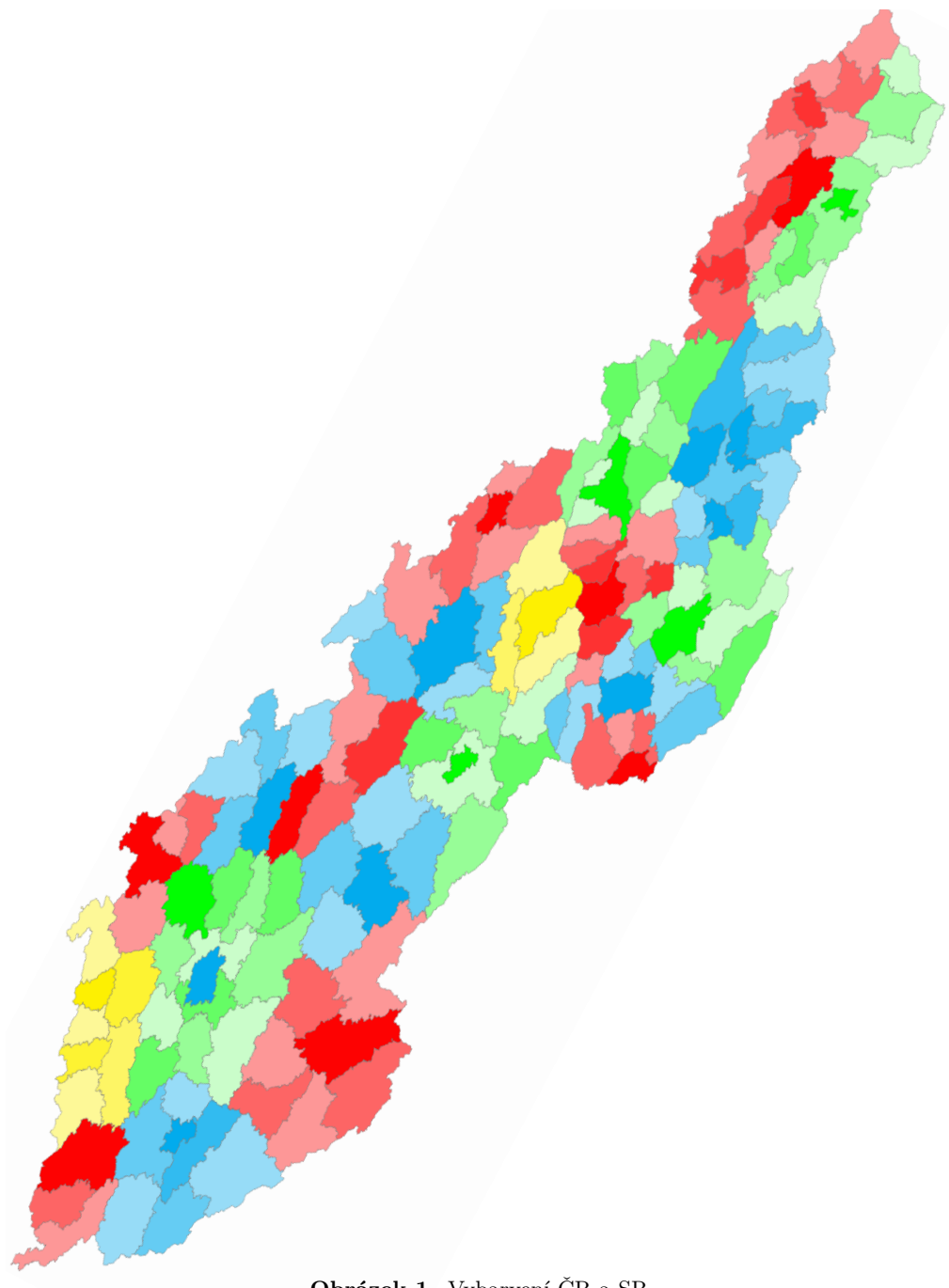
2. Zpět k Pythonu: případ Sage

Podobného vykreslení ze surových dat lze dosáhnout i v Pythonu, např. přes knihovnu `matplotlib`. Jen naznačím strukturu.

```
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.patches import Polygon
combined=Polygon([ (0,1), (1,2), (2,1), (1,0) ])
fig, ax = plt.subplots()
ax.add_patch(combined)
plt.ylim(0,3)
plt.xlim(0,3)
plt.show()
```

Poněvadž jsem se nedopracoval výsledku přes knihovnu `networkx`, hledal jsem další cesty. Zkusil jsem si SageMathCell, <https://sagecell.sagemath.org/>, či <https://cocalc.com/>. Řada věcí je placených, ale na testy to stačilo. Za pozornost stojí možnost týmové práce, diskuze i videohovoru. Náš problém spadá v dokumentaci do této části, viz https://doc.sagemath.org/html/en/reference/graphs/sage/graphs/graph_coloring.html.

```
from sage.graphs.graph_coloring import number_of_n_colorings
from sage.graphs.graph_plot import GraphPlot
# Jihomoravský kraj
G = Graph({
    17: [67,68],
    32: [61,68],
    49: [61,67],
    61: [32,49,66,67,68],
```



Obrázek 1. Vybarvení ČR a SR

```

66: [61],
67: [17,49,61,68],
68: [17,32,61,67],
})
for n in range(8): # [4]
    print(number_of_n_colorings(G,n))
G.graphplot(layout='graphviz', prog="twopi").show()

```

A právě během toho testu jsem si zkusil čtyři barvy, ale pak i jiný počet barev. A dostal jsem nápad propočítat všechny možnosti pro okresy a kraje. U krajů je největší počet uzlů (22), což byla výpočetní výzva. Sage začal kolabovat u šesti barev. Dostupných 45 minut na výpočty by určitě nestačilo. Nestačil by na to v rozumném čase ani jeden počítač, bylo potřeba výpočty distribuovat a podívat se po nějaké úspoře, spisovně po lepším algoritmu.

3. Zpět k rekurzi

Od dvojíček, Rudolfa Blaška a Aleše Kozubíka, prakticky ve stejnou chvíli, došla prosba, jestli by nebylo možné vybarvit kraje (hlavní barva) a poté vybarvit okresy uvnitř kraje (odstín hlavní barvy). Na vše požadavek, aby zkoumaní sousedi neměli stejnou barvu. Poněvadž jsou počty krajů a okresů v řádech desítek, mohli jsme zkusit až 6 barev.

Při nejnižším možném počtu barev jsem vybral první obarvení a užil. Jako bonus se krajské město (vyjma Středočeského kraje) nastavilo nejsytější barvou. Na obrázku 1 je závěrečná ukázka. Kdyby se šlo na úroveň ohrazení měst, dokáží si představit užití šrafování v dané barvě a intenzitě.

Druhou věc, kterou jsem si zkusil bylo vygenerovat počty různých obarvení při daném počtu barev. Obvody Prahy a okresy Bratislavy a Košic jsem si sloučil dohromady.

Co mě během bádání zaujalo je, že můžeme použít menší počet barev než 4, 4 barvami lze oblasti vždy vybarvit, ale nic nás neomezuje v tom užít i víc barev. Matematická teorie hovoří o tom, že pokud je víc barev než oblastí, počet obarvení je nula, angl. number of n -colourings. Lze si to představit tak, že obarvíme všechny oblasti, ale když nám některé barvy zůstanou, pak to není obarvení přesně n barvami.

Při obarvování můžeme barvy prohazovat (Štefan Peško upřesnil, že se jedná o automorfismus), pokud tedy najdeme počet neautomorfních obarvení násobíme zjištěný počet faktoriálem počtu barev. Výsledky jsou zmíněny v následujících dvou tabulkách. V první tabulce 1 jsou počty neautomorfních obarvení, ve druhé tabulce 2 pak celkové počty obarvení.

Zvláštní kategorii tvoří kraje, kde vychází obrovský počet obarvení, proto jsem je z okresů vyčlenil do samostatné tabulky.

Tabulka 1. Počty neautomorfních obarvení mapy ČR a SR

Zkoumaná oblast	Uzlů	Počet barev													
		min	1	2	3	4	5	6	7	8	9	10	11	12	13
Hlavní město Praha Středočeský kraj Jihočeský kraj Plzeňský kraj Karlovarský kraj Ústecký kraj Liberecký kraj Královéhradecký kraj Pardubický kraj Kraj Vysočina Jihomoravský kraj Olomoucký kraj Moravskoslezský kraj Zlínský kraj Bratislavský kraj Trnavský kraj Trenčínský kraj Nitranský kraj Žilinský kraj Banskobystrický kraj Prešovský kraj Košický kraj	1	1	–	–	–	–	–	–	–	–	–	–	–	–	–
	12	4	–	–	230	5 320	22 547	31 283	18 254	5 036	681	43	1	–	–
	7	3	–	–	1	15	25	10	1	–	–	–	–	–	–
	7	4	–	–	–	8	19	9	1	–	–	–	–	–	–
	3	3	–	–	1	–	–	–	–	–	–	–	–	–	–
	7	3	–	–	1	15	25	10	1	–	–	–	–	–	–
	4	3	–	–	2	1	–	–	–	–	–	–	–	–	–
	5	3	–	–	1	3	1	–	–	–	–	–	–	–	–
	4	3	–	–	1	1	–	–	–	–	–	–	–	–	–
	5	3	–	–	1	3	1	–	–	–	–	–	–	–	–
	7	3	–	–	2	22	31	–	1	–	–	–	–	–	–
	5	3	–	–	4	5	1	–	–	–	–	–	–	–	–
	6	3	–	–	2	7	6	1	–	–	–	–	–	–	–
	4	3	–	–	1	1	–	–	–	–	–	–	–	–	–
	4	3	–	–	1	1	–	–	–	–	–	–	–	–	–
	7	3	–	–	8	46	46	13	1	–	–	–	–	–	–
	9	3	–	–	4	140	506	490	171	23	1	–	–	–	–
	7	3	–	–	1	15	25	10	1	–	–	–	–	–	–
	11	4	–	–	–	252	3 024	7 770	6 951	2 646	462	36	1	–	–
	13	4	–	–	–	300	12 273	77 681	152 670	124 688	48 877	9 837	1 026	52	1
13	3	–	–	2	1 150	29 467	147 451	247 780	179 753	64 015	11 881	1 155	55	1	
8	3	–	–	16	146	230	111	19	1	–	–	–	–	–	

Tabulka 2. Celkové počty obarvení mapy ČR a SR

	Počet barev												
	1	3	4	5	6	7	8	9	10	11	12	13	
PA	1	–	–	–	–	–	–	–	–	–	–	–	
SC	–	–	5 520	638 400	16 233 840	157 666 320	736 001 280	1 827 463 680	2 471 212 800	1 716 422 400	479 001 600	–	
JC	–	6	360	3 000	7 200	5 040	–	–	–	–	–	–	
PM	–	–	192	2 280	6 480	5 040	–	–	–	–	–	–	
KV	–	6	–	–	–	–	–	–	–	–	–	–	
UL	–	6	360	3 000	7 200	5 040	–	–	–	–	–	–	
LB	–	12	24	–	–	–	–	–	–	–	–	–	
HK	–	6	72	120	–	–	–	–	–	–	–	–	
PU	–	6	24	–	–	–	–	–	–	–	–	–	
VY	–	6	72	120	–	–	–	–	–	–	–	–	
JM	–	12	528	3 720	7 920	5 040	–	–	–	–	–	–	
OL	–	24	120	120	–	–	–	–	–	–	–	–	
MS	–	12	168	720	720	–	–	–	–	–	–	–	
ZL	–	6	24	–	–	–	–	–	–	–	–	–	
BA	–	6	24	–	–	–	–	–	–	–	–	–	
TT	–	48	1 104	5 520	9 360	5 040	–	–	–	–	–	–	
TN	–	24	3 360	60 720	352 800	861 840	927 360	362 880	–	–	–	–	
NR	–	6	360	3 000	7 200	5 040	–	–	–	–	–	–	
ZA	–	–	6 048	362 880	5 594 400	35 033 040	106 686 720	167 650 560	130 636 800	39 916 800	–	–	
BB	–	–	7 200	1 472 760	55 930 320	769 456 800	5 027 420 160	17 736 485 760	35 696 505 600	40 954 636 800	24 908 083 200	6 227 020 800	
PO	–	12	27 600	3 536 040	106 164 720	1 248 811 200	7 247 640 960	23 229 763 200	43 113 772 800	46 103 904 000	26 345 088 000	6 227 020 800	
KE	–	96	3 504	27 600	79 920	95 760	40 320	–	–	–	–	–	

PA – Hlavní město Praha, SC – Středočeský kraj, JC – Jihočeský kraj, PM – Plzeňský kraj, KV – Karlovarský kraj,

UL – Ústecký kraj, LB – Liberecký kraj, HK – Královéhradecký kraj, PU – Pardubický kraj, VY – Kraj Vysočina,

JM – Jihomoravský kraj, OL – Olomoucký kraj, MS – Moravskoslezský kraj, ZL – Zlínský kraj, BA – Bratislavský kraj,

TT – Třinec, TN – Trenčinský kraj, NR – Nitrianský kraj, ZA – Žilinský kraj,

BB – Banskobystrický kraj, PO – Prešovský kraj, KE – Košický kraj

4. Technické pozadí výpočtů

Jak však tato obarvení pro tak velký počet uzlů najít v rekurzi s komplikovanou stromovou strukturou?

V matematice, speciálně v teorii grup a kombinatorice, se hledají symetrie a podobné vztahy, které by výpočty usnadnily. Zde se mi podařilo v rekurzi stromovou strukturu obarvení uzlů snížit na hledání jen neautomorfních větví. Pracovně to nazývám nalézt tzv. „prvoobarvení“. Je to rostoucí sekvence prvních výskytů barev, kdy každá barva je zastoupena alespoň jednou. Počet dalších výskytů barev nesmí překročit hranici počet uzlů minus počet barev, takovou větev jsem v rekurzi vyřadil.

Zmíním několik ukázek a zdůvodnění vyřazení větve z dalšího zkoumání v rekurzi pro 4 barvy, 7 uzlů, graf nechť je smyšlený.

B	Na prvním místě nemůže být druhá barva v pořadí, nedostali bychom uspořádanou n -tici. Podobně by se vyřadily C a D.
ABABD	Na páté pozici mohou být barvy A až C, protože nejvyšší předchodí je barva B. Lze připustit jen o jednu vyšší pozici (barva C).
ABABCB	Pokud by se jednalo o vybarvení 4 barvami, chybí na poslední pozici barva D. Pokud rekurze ještě běží, D lze užít, žádný ze sousedů barvu D nemá.
AAABBBC	Vyplnilo se sice 7 uzlů, ale neušly se všechny barvy (barva D).
ABBABA	Zbývá vybarvit poslední uzel, ale potřebujeme užít ještě barvy C a D, na ty není už dost uzlů. Volné uzly jsou 3. až 6., tedy 4 uzly, ale mohou být jen tři (7 uzlů minus 4 barvy je maximum).
ABCDBCD	Tato situace by v závislosti na grafu mohla být platným obarvením, ale prvoobarvení by skončilo už při ABCD.
DCBA	Toto obarvení se získá z nejkratšího možného prvoobarvení ABCD permutací, počet nalezených obarvení se tedy násobí 4! (tj. 24). Technicky měla být větev vyřazena již na první pozici u barvy D.

Zde je ukázka prvoobarvení v plné délce pro 22 uzlů krajů a 4 barvy. Každá sekvence začíná první barvou A a končí poslední barvou D. Barva vždy splňuje podmínku, že je někde v rozsahu od A až nejvyšší barva z levých uzlů + 1 pozice. Tučným fontem červeně naznačuji první výskyt dané barvy. V této malé ukázce je 15 prvoobarvení. Více jich není.

22 krajů ve 4 barvách		Počet obarvení ve větvi	
Pořadí	Prvoobarvení	neautomorfní	celkem (krát 24)
1.	AB ACAD	22 032	528 768
2.	AB AC B ACACD	1 296	31 104

22 krajů ve 4 barvách		Počet obarvení ve větvi	
Pořadí	Prvoobarvení	neautomorfní	celkem (krát 24)
3.	ABACBACAD	1 296	31 104
4.	ABACBACD	6 480	155 520
5.	ABACBAD	9 072	217 728
6.	ABACBD	22 032	528 768
7.	ABACD	18 144	435 456
8.	ABCABCACAD	1 296	31 104
9.	ABCABCACD	1 296	31 104
10.	ABCABCAD	6 480	155 520
11.	ABCABCD	9 072	217 728
12.	ABCABD	22 032	528 768
13.	ABCACD	22 032	528 768
14.	ABCAD	18 144	435 456
15.	ABCD	80 352	1 928 448
	Součet	241 056	5 785 344

Možná si kladete otázku, jaktože neexistuje sekvence typu **AAAAABCD**. Je to dané zadáním úlohy, že sousedící uzly nesmí mít stejnou barvu. Uzly si můžeme různě přecíslovat (nemusí to být cesta ve stylu známém z teorie grafů), ale podmínku sousedících uzlů nesmíme opominout. Například u Košického kraje z okresů nelze Eulerovu cestu, tzv. jednotahovku, ani najít. Obecně však taková sekvence může být platná, záleží na konkrétním grafu a užitém pořadí uzlů.

Poté hledáme platná obarvení z těchto sekvencí rekurzí od určité hloubky a jejich součet nám dá počet neautomorfních obarvení. To pak stačí vynásobit faktoriálem počtu barev. Samotná prvoobarvení nebyla na rozložení rekurze vhodná, některé sekvence byly spočítány rychle, jiné pomalu. Nápad tedy byl, vzít sice jen tato prvoobarvení, ale do určité hloubky. To se nakonec užilo. Post mortem řečeno: ještě lepší varianta by byla, aby se kratší prvoobarvení než nastavená hloubka rekurze nechaly rozvinout do nastavené hloubky. To by bylo ještě víc časově rovnoměrné. Třeba užiji v budoucnu.

Touto fintou se mi podařilo eliminovat větve s proházenými barvami v duchu kombinatoriky. Následně jsem rozhodl rekurzivní výpočty na několik stovek jader na cca tři dny. A bylo spočteno! V průměru jsem užil 17000 úloh na barvu, to bylo při hloubce 11 uzlů. Lepší bylo užít hloubku 9 či 10 uzlů, těch úloh bylo zbytečně moc.

Poznámka. Nejsem zběhlý v C++, zdrojové kódy v Pythonu, především užité datové struktury, jsem si nechal převádět v <https://products.codeporting.ai/convert/python-to-cpp>. Věřím tomu, že podobných nástrojů, ať už s nebo

bez AI asistence, se rodí nespočet. Bez C++ to bylo nepočítatelné, Python není na výpočty optimalizován. Možná by zabral ještě Rust či Go, ale nezkoušel jsem je. Poctivě se učím C++. Dorazil mi tip, že Python nemám vzdávat, ale užít knihovny `numpy` a `pandas`. Zůstává mi otevřené s otazníkem.

5. Postřeh

Při výpisu počtu obarvení na uzlech v rekurzi to vypadá, že by se dalo uspořít ještě víc, protože počty na větvích na mnoha místech vycházely stejné. Chce se to však víc ponořit do teorie grafů, možná do kombinatoriky. Rekurzi se nejspíš vyhnout nedá, ale naděje na úsporu umírá poslední. Po objevení takového Svatého Grálu by se dal počet obarvení spočítat i pro větší grafy/mapy. Jak však velké je těžké odhadovat. Dávám si mezi otevřené problémy a důležité úkoly po OSSConf.

Kontaktní adresa

Ing. Pavel Stríž, Ph.D., Nakladatelství Martin Stríž, U Škol 940, Bučovice, okres Vyškov, 685 01, Česká republika,
E-mailová adresa: `pavel@striz.cz`