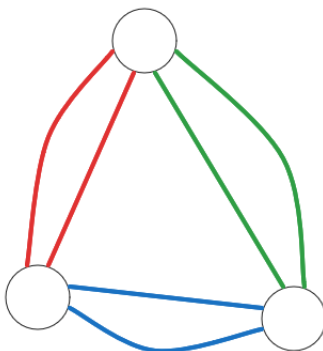


Report ročníkový projekt zimný semester 2024/2025

Alex Diko
diko3@uniba.sk

Pojmy: *Graf* je dvojica $G = (V, E)$, kde V, E sú množiny také, že $E \subseteq V \times V$ a $V \cap E = \emptyset$. Prvky V sa nazývajú *vrcholy* grafu G a prvky E jeho *hrany*. Množinu hrán grafu G označujeme ako $E(G)$. Vrchol v je *incidentný* s hranou e ak $v \in e$. *Stupeň vrcholu* je počet hrán s ním incidentných. Dve hrany sú *susedné* ak existuje vrchol, ktorý je s obidvomi incidentný. Ak všetky vrcholy grafu majú stupeň k , graf sa nazýva *k-regulárny*. *Cesta* je neprázdny graf $P = (V, E)$, kde $V = \{x_0, x_1, \dots, x_k\}$ a $E = \{x_0x_1, x_1x_2, \dots, x_{k-1}x_k\}$, pričom všetky x_i sú rôzne. Túto cestu označíme $x_0x_1 \dots x_k$. Ak $P = x_0 \dots x_{k-1}$ je cesta, potom graf $C := P + x_{k-1}x_0$ sa nazýva *kružnica*. *Dĺžka* kružnice je počet jej hrán. Graf $G'(V', E')$ je *podgrafom* grafu G ak $V' \subseteq V$ a $E' \subseteq E$. *Rozklad* množiny $E(G)$ je množina $R = \{R_1, \dots, R_k\}$, kde R_i sú navzájom disjunktné a zjednotenie $\cup R$ všetkých množín $R_i \in R$ sa rovná $E(G)$. [1]

Kružnicovou dekompozíciou grafu nazveme rozklad $E(G)$ na kružnice, ktoré nezdieľajú žiadnu hranu. Ak každá kružnica tohto rozkladu má párnú dĺžku nazveme tento rozklad *párny* (ďalej *ECD* z anglického *Even cycle decomposition*). Pre *ECD* grafu zafarbíme každú jeho kružnicu tak, aby kružnice so spoločným vrcholom nemali rovnakú farbu. Potom zjednotenie kružníc v každej farebnej triede bude 2-regulárny podgraf pôvodného grafu. Ak najmenší počet farieb potrebných na takéto zafarbenie je m potom *ECD* rozklad má *veľkosť* m . [2]



Obr. 1: ECD rozklad veľkosti 3 na červenú, modrú a zelenú kružnicu

Implementácia: Do grafovej knižnice *ba-graph* som v jazyku c++ naimplementoval backtracking (prehľadávanie s návratom) algoritmus, ktorý nájde veľkosť *ECD* grafu. Pseudokód algoritmu je uvedený nižšie.

```
1: function SKÚSZAČAŤNOVÚKRUŽNICU
2:   if všetky hrany sú priradené farebnej triede then
3:     porovnaj veľkosť tohto ECD s doteraz najmenším, ak je menšie zapamätaj si farby hrán
4:     return
5:   else
6:      $h \leftarrow$  zober nejakú nepriradenú hranu
7:     for all  $f \in$  aktuálne farebné triedy do
8:       skús priradiť  $h$  do  $f$  a zapamätaj si, že  $h$  bude v kružnici na párnej pozícii
9:       SKÚSPOKRAČOVAŤVAKTUÁLNEJKRUŽNICI( $h$ )
10:    end for
11:    skús priradiť  $h$  do novej farebnej triedy a zapamätaj si, že  $h$  bude v kružnici na párnej pozícii
```

```

12:     if počet farebných tried je väčší alebo rovný ako aktuálne najmenšie nájdené ECD then
13:         odmietni aktuálne riešenie
14:         return
15:     end if
16:     SKÚSPOKRAČOVAŤVAKTUÁLNEJKRUŽNICI(h)
17: end if
18: end function
19: function SKÚSPOKRAČOVAŤVAKTUÁLNEJKRUŽNICI(aktuálnaHrana)
20:     ▷ Over či je aktuálnaHrana správne zafarbená. V korektnej ECD musia byť práve 2 susedné hrany
      v rovnakej farebnej triede a na pozíciách inej parity ako aktuálnaHrana <
21:      $p \leftarrow 0$  ▷ Počet susedných hrán v rovnakej farebnej triede ale na pozíciách rôznej parity
22:     for all  $h \in$  hrany susedné s aktuálnaHrana do
23:         if  $h$  je v rovnakej farebnej triede ako aktuálnaHrana then
24:             if  $h$  je na pozícii rovnakej parity ako aktuálnaHrana then
25:                 odmietni aktuálnu kružnicu
26:                 return
27:             else
28:                  $p \leftarrow p + 1$ 
29:             end if
30:         end if
31:     end for
32:     if  $p > 2$  then
33:         odmietni aktuálnu kružnicu
34:         return
35:     else if  $p = 2$  then
36:         SKÚSZAČAŤNOVÚKRUŽNICU ▷ Bez sporu sa nám podarilo vytvoriť kružnicu párnej dĺžky
37:         return
38:     else
39:         for all  $h \in$  hrany susedné s aktuálnaHrana a aktuálne nepriradené do farebnej triedy do
40:             skús priradiť  $h$  do rovnakej farebnej triedy ako aktuálnaHrana ale na pozíciu inej parity
              SKÚSPOKRAČOVAŤVAKTUÁLNEJKRUŽNICI( $h$ )
41:         end for
42:     end if
43: end function

```

K algoritmu su urobené aj testy. Skúšal som viacero variant algoritmu (napríklad, že iterujeme cez všetky možné veľkosti ECD a pre každú veľkosť skúsime nájsť jedno zafarbenie. Ak ho nájdeme, skončíme.), ale tento bol najrýchlejší. Výber hrany na riadku 6. je nedeterministický. Skúšal som program, kde je poradie výberu hrán náhodne a čas rôzne spustenie programu na rovnakých vstupoch trvaly rôzne dlho. Teda to v akom poradí sú tam vyberané hrany ovplyvňuje rýchlosť. Skúšal som heuristiku, kde sa vždy vyberie hrana, ktorá má najviac susedov priradených do nejakej farebnej triedy. Rýchlejší bol však variant, kde sa vrcholy vyberajú na základe poradia, v ktorom sú prehladávané do šírky BFS algoritmom od prvej hrany, ktorá sa zafarbí. Varianty backtrackingu sa dajú pozrieť ako iné git vetvy.

Práca na letný semester: ECD by sa malo dať nájsť pomocou SAT solvera.

Referencie

- [1] Reinhard Diestel. *Graph theory*. Springer (print edition); Reinhard Diestel (eBooks), 2024.
- [2] Analen A. Malnegro a Kenta Ozeki. „H-colorings for 4-regular graphs“. In: *Discrete Mathematics* 347.3 (2024), s. 113844. ISSN: 0012-365X. DOI: <https://doi.org/10.1016/j.disc.2023.113844>. URL: <https://www.sciencedirect.com/science/article/pii/S0012365X23005307>.