

Report ročníkový projekt 2024/2025

Alex Diko
diko3@uniba.sk

Práca za letný semester:

Do knižnice **ba-graph** som doprogramoval hľadanie ECD pomocou SAT solvera. Nech riešime úlohu, kde sa pýtame, či graf $G = (V, E)$ obsahuje ECD veľkosti k .

Použijeme nasledujúce binárne premenné:

$\forall e \in E \forall c \in \{1, \dots, k\} : col_{e,c} = 1 \iff e$ patrí do i -tej farebnej triedy

$\forall e \in E : even_e = 1 \iff e$ je vo svojej farebnej triede na párnej pozícii

SAT solveru dáme nasledujúce klauzuly:

$\forall e \in E : \bigvee_{1 \leq i \leq k} col_{e,i}$ (každá hrana patrí do aspoň jednej farebnej triedy)

$\forall e \in E \forall c_1 \in \{1, \dots, k\} \forall c_2 \in \{c_1 + 1, \dots, k\} : \neg col_{e,c_1} \vee \neg col_{e,c_2}$

(každá hrana patrí do najviac jednej farebnej triedy)

Nech $NE(e)$ označuje množinu hrán susedných s hranou e a $NE(v)$ množinu hrán susedných s vrcholom v

$\forall e \in E \forall e_2 \in NE(e) \forall c \in \{1, \dots, k\} : \neg col_{e,c} \vee \neg col_{e_2,c} \vee \neg even_e \vee \neg even_{e_2}$ a

$\neg col_{e,c} \vee \neg col_{e_2,c} \vee even_e \vee even_{e_2}$

(ak sú susedné hrany v rovnakej farebnej triede, musia byť na pozíciach inej parity)

$\forall (v_1, v_2) \in E \forall c \in \{1, \dots, k\} \forall u \in \{v_1, v_2\} : \neg col_{(v_1, v_2), c} \vee \bigvee_{e_2 \in (NE(u) - \{(v_1, v_2)\})} col_{e_2, c}$

(každý vrchol hrany musí byť susedný s aspoň jednou hranou v rovnakej farebnej triede)

Pre CNF v takejto podobe môže existovať veľa symetrických riešení. Na rozbitie symetrie som skúšal použiť nejaké vlastné podmienky, ale nakoniec efektívnejšie bolo použitie knižnice **breakid**. Zároveň som vylepšil testy testujúce programy na hľadanie ECD. Ďalej som prešiel literatúru a spísal zoznam známych faktov o ECD (dá sa nájsť tu: <https://www.overleaf.com/read/hxbcdhzbngcd#fe15ca>).

Práca za zimný semester:

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

```
1: function NÁJDIECD(graf)
2:   inicializuj počet farebných tried na 0
3:   SKÚSZAČAĽNOVÚKRUŽNICU
4:   if žiadne ECD sa nenašlo then
5:     | return neexistuje ECD
6:   else
7:     return najmenšie ECD
```

```

8:   end if
9: end function
10:
11: function SKÚSZAČAĽNOVÚKRUŽNICU
12:   if všetky hrany sú priradené farebnej triede then
13:     porovnaj veľkosť tohto ECD s doteraz najmenším nájdeným, ak je menšie zapamätaj si prira-
        denie hrán do farebných tried
14:     return
15:   else
16:      $h \leftarrow$  zober nejakú nepriradenú hranu
17:     for all  $f \in$  aktuálne farebné triedy do
18:       skús priradiť  $h$  do  $f$  a zapamätaj si, že  $h$  bude v kružnici na párnej pozícii
19:       SKÚSPOKRAČOVAŤVAKTUÁLNEJKRUŽNICI( $h$ )
20:     end for
21:     skús priradiť  $h$  do novej farebnej triedy a zapamätaj si, že  $h$  bude v kružnici na párnej pozícii
22:     if počet farebných tried je väčší alebo rovný ako aktuálne najmenšie nájdené ECD then
23:       odmietni aktuálne riešenie
24:       return
25:     end if
26:     SKÚSPOKRAČOVAŤVAKTUÁLNEJKRUŽNICI( $h$ )
27:   end if
28: end function
29:
30: function SKÚSPOKRAČOVAŤVAKTUÁLNEJKRUŽNICI(aktuálnaHrana)
31:   ▷ 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 <
32:    $p \leftarrow 0$            ▷ Počet susedných hrán v rovnakej farebnej triede ale na pozíciách rôznej parity
33:   for all  $h \in$  hrany susedné s aktuálnaHrana do
34:     if  $h$  je v rovnakej farebnej triede ako aktuálnaHrana then
35:       if  $h$  je na pozícii rovnakej parity ako aktuálnaHrana then
36:         odmietni aktuálnu kružnicu
37:         return
38:       else
39:          $p \leftarrow p + 1$ 
40:       end if
41:     end if
42:   end for
43:   if  $p > 2$  then
44:     odmietni aktuálnu kružnicu
45:     return
46:   else if  $p = 2$  then
47:     SKÚSZAČAĽNOVÚKRUŽNICU           ▷ Bez sporu sa nám podarilo vytvoriť kružnicu párnej dĺžky
48:     return
49:   else
50:     for all  $h \in$  hrany susedné s aktuálnaHrana a aktuálne nepriradené do farebnej triedy do
51:       skús priradiť  $h$  do rovnakej farebnej triedy ako aktuálnaHrana ale na pozíciu inej parity
        SKÚSPOKRAČOVAŤVAKTUÁLNEJKRUŽNICI( $h$ )
52:     end for
53:   end if
54: end function

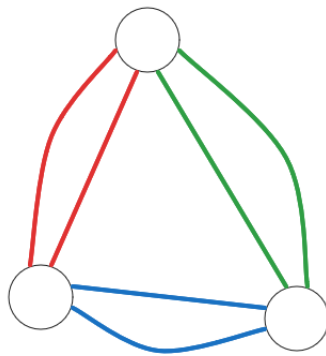
```

K algoritmu sú urobené aj testy (`test_ecd.cpp`). Skúšal som viacero variant algoritmu (napríklad, že iterujeme cez všetky možné veľkosti ECD od najmenej po najväčšiu a pre každú veľkosť skúsime nájsť jedno zafarbenie. Ak ho nájdeme, skončíme.), ale tento bol najrýchlejší. Pri výbere hrany na riadku 16 máme viacero možností. Skúšal som program, kde je poradie výberu hrán náhodne a spustenia programu na rovnakých vstupoch trvali 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 prehľadávané do šírky BFS algoritmom od prvej hrany, ktorá sa zafarbí. Varianty backtrackingu sa dajú pozrieť ako rôzne `git` vetvy.

Pojmy:

Graf je dvojica $G = (V, E)$ disjunktných množín V a E spolu so zobrazením $f : E \rightarrow V \cup \binom{V}{2}$. Prvky V sa nazývajú *vrcholy* grafu G a prvky E jeho *hrany*. f priraduje každej hrane jej *konce* (teda povoluujeme multigrafy). Ak je hrane priradený iba jeden vrchol, nazveme ju *slučka*. Hranu s koncovými vrcholmi x, y označíme xy . Množinu hrán grafu G označujeme ako $E(G)$. Vrchol v je *incidentný* s hranou e , ak $v \in f(e)$. *Stupeň vrcholu* je počet hrán s ním incidentných, pričom slučka pridáva k stupni vrchola 2. 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 nezdediajú 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

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>.