

Report - Zimný semester

Cieľom projektu je zefektívniť výpočet cirkulárneho chromatického indexu pre grafy s malým maximálnym stupňom. Pôvodné metódy v knižnici ba-graph narážali na výkonnostné limity pri väčších grafoch alebo špecifických zlomkoch p/q .

Práca v zimnom semestri sa sústredila na tri kľúčové oblasti:

- 1.) Algoritmickú optimalizáciu detekcie cyklov, ktorá je kritickým úzkym hrdlom pri generovaní podmienok.
- 2.) Integráciu moderných solverov (SAT a CP-SAT) pre exaktné riešenie.
- 3.) Implementáciu heuristických metód pre rýchle nájdenie kandidátskych riešení.

1.) Optimalizácia grafových algoritmov

Generovanie podmienok pre solvery vyžaduje enumeráciu kružníc dĺžky 0 (mod p). Pôvodná implementácia funkcie `all_circuits` bola pre praktické použitie príliš pomalá.

- Implementoval som optimalizované verzie `all_circuits_fast` a `all_circuits_fast_mod`.
- Namiesto hrubej sily som využil efektívnejší backtracking, čo umožňuje spracovať aj grafy, kde počet cyklov rastie exponenciálne, v prijateľnom čase pre potreby generovania SAT klauzúl.
- Táto optimalizácia bola nevyhnutným prerekvizitom pre efektívne nasadenie SAT solverov.
- [Link na GitHub](#)

2.) Exaktné metódy: SAT a CP-SAT

Pre rozhodovanie o existencii (p, q) - farbenia boli implementované dva prístupy:

- **SAT Solver (Boolovská formula):** Implementoval som logiku pre detekciu tesných cyklov. Prítomnosť tesných cyklov je nutnou podmienkou pre neexistenciu farbenia, čo nám umožňuje rýchlejšie zamietnuť nevalidné prípady. Refaktoroval som detekciu tak, aby hľadala farbenia bez tesných cyklov s využitím redukcie pomocou najväčšieho spoločného deliteľa na odfiltrovanie irelevantných cyklov.

- **CP-SAT Solver (Google OR-Tools):** Do knižnice ba-graph som integroval solver CP-SAT z balíka Google OR-Tools. Ukázalo sa, že pre určité typy inštancií je constraint programming výrazne efektívnejší než čistý SAT prístup (Kissat). Implementácia podporuje všeobecné cirkulárne vrcholové aj hranové farbenie s možnosťou predfarbenia hrán.
- [Link na GitHub](#)

3.) Heuristické metódy

Keďže exaktný výpočet je NP-ťažký, implementoval som randomizovaný heuristický solver založený na lokálnom prehľadávaní (inšpirovaný metódou CVD).

- Algoritmus sa snaží nájsť validné cirkulárne hranové farbenie pomocou lokálnych zmien a riešenia konfliktov vo vrcholoch.
- Táto metóda slúži na rýchle overenie existencie farbenia pri väčších grafoch, kde by exaktný solver bežal príliš dlho.
- [Link na GitHub](#)

Testovanie a integrácia

Súčasťou práce bolo vytvorenie robustnej sady testov na zabezpečenie správnosti implementácie:

- **Regresné testy:** Pridané testy pre cirkulárny chromatický index a kritickosť grafov.
- **Rozdelenie testov:** Testy boli v štruktúre projektu rozdelené na rýchle a pomalé, čo umožňuje efektívnejší vývoj.
- **Validácia:** Všetky nové komponenty (SAT, CP-SAT, heuristiky) sú pokryté testami overujúcimi ich konzistenciu na známych grafoch.
- [Link na GitHub](#)