A-tokovo kritické grafy – Zimný report

Na začiatku práce na projekte som sa oboznámil so základnou teóriou tokov v grafoch a s definíciou nikde nulového A-toku v abelovských grupách. Cieľom projektu bolo navrhnúť a implementovať jednoduchý algoritmus na zistenie, či je daný graf A-tokovo kritický (či nemá nikde nulový A-tok, ale po identifikácii ľubovoľných dvoch vrcholov už nikde nulový A-tok existuje).

Reprezentácia grafu a orientácia hrán

V implementácii som použil jednoduchú reprezentáciu grafu so zoznamom hrán a predpočítaným zoznamom susedností. Hrany sú reprezentované ako orientované, avšak samotný graf možno chápať ako neorientovaný. Orientácia hrán je použitá len ako technický prostriedok na výpočet hodnoty balance vo vrcholoch, keďže podľa definície A-tok je aj priradenie orientácie.
V algoritme pracujem s fixnou orientáciou hrán, čo je korektné, keďže existencia nikde nulového A-toku nezávisí od počiatočnej orientácie grafu. Pri zmene orientácie hrany je možné priradiť inverzný prvok z abelovskej grupy A, čím sa zachová hodnota balance vo vrcholoch. Ak hrana s hodnotou x prispieva k balance vrchola ako outflow, po zmene orientácie s hodnotou –x bude prispievať ako inflow, pričom výsledná balance vrchola zostane nezmenená.

Voľba grupy A

Hoci je problém formulovaný pre ľubovoľnú abelovskú grupu A, v implementácii som pracoval výhradne s cyklickými grupami Z_k . Toto zjednodušenie je korektné, keďže existencia nikde nulového A-toku závisí iba od počtu prvkov grupy a nie od štruktúry.

Algoritmus a optimalizácie

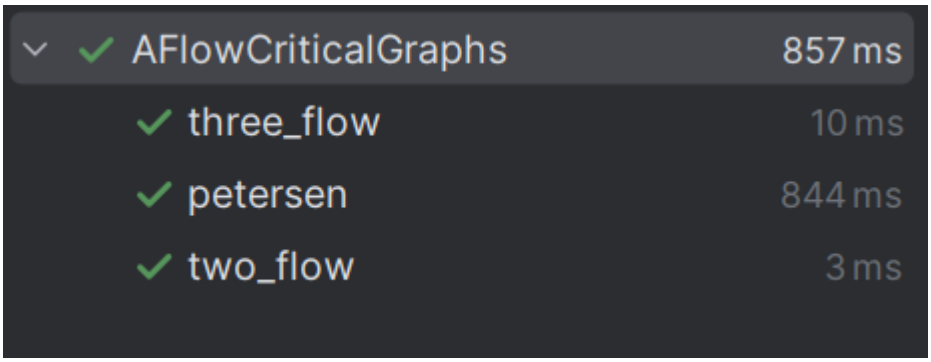
Existencia nikde nulového A-toku je overovaná pomocou backtrackingu, ktorý postupne priraďuje nenulové hodnoty hranám a priebežne aktualizuje hodnotu balance vo vrcholoch. Po spracovaní všetkých hrán sa overí, či je balance každého vrchola rovná nule modulo k.
Základná verzia algoritmu využívala jednoduchý backtracking, ktorý skúšal všetky možné nenulové hodnoty pre jednotlivé hrany. Pri testovaní na väčších grafoch sa však ukázalo, že takýto prístup naráža na výkonnostné obmedzenia. Preto som algoritmus postupne vylepšil pomocou viacerých optimalizácií, ktoré výrazne znížili počet prehľadávaných vetiev.

Hrany sú v algoritme spracovávané v poradí určenom podľa stupňa ich koncových

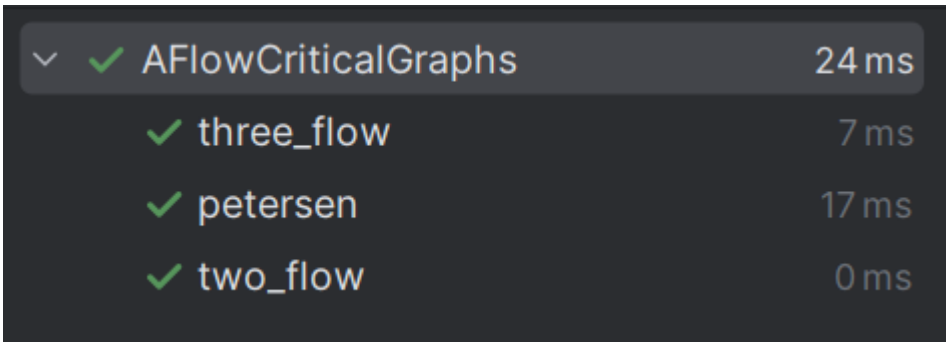
vrcholov. Experimentálne testy ukázali, že spracovanie hrán s vyšším stupňom ako prvých vedie k rýchlejšiemu ukončeniu nevhodných vetiev prehľadávania.

Na základe odporúčania školiteľa som nakoniec implementoval optimalizáciu, ktorá pre vrcholy s iba jednou zostávajúcou nespracovanou hranou priamo určí hodnotu toku tak, aby sa ich balance vyrovnala na nulu. Následné testy ukázali, že táto optimalizácia je veľmi efektívna, a to aj pre grafy s 18–20 vrcholmi, vrátane prípadov, keď graf nie je A-tokovo kritický.

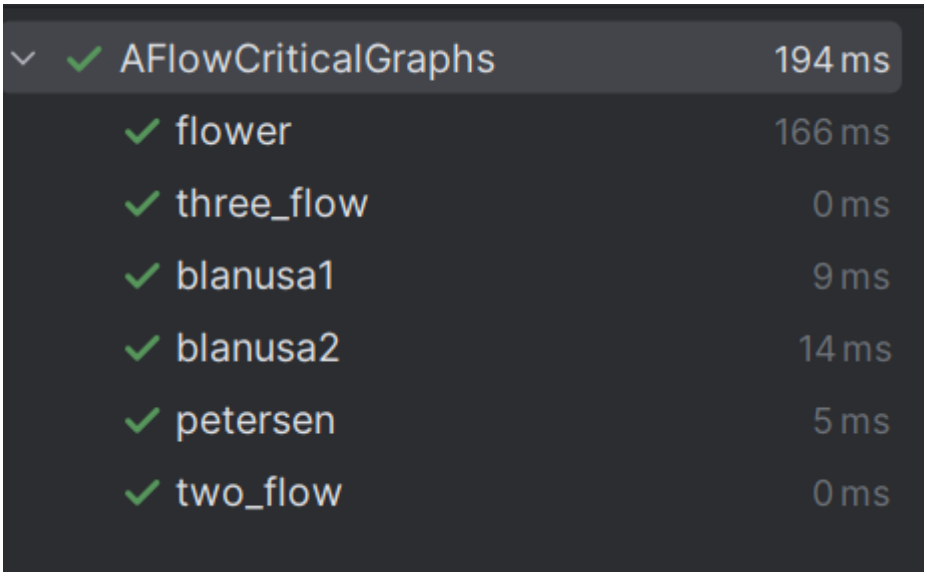
Obr. 1-4 porovnávajú čas behu algoritmu pred a po aplikovaní optimalizácií:



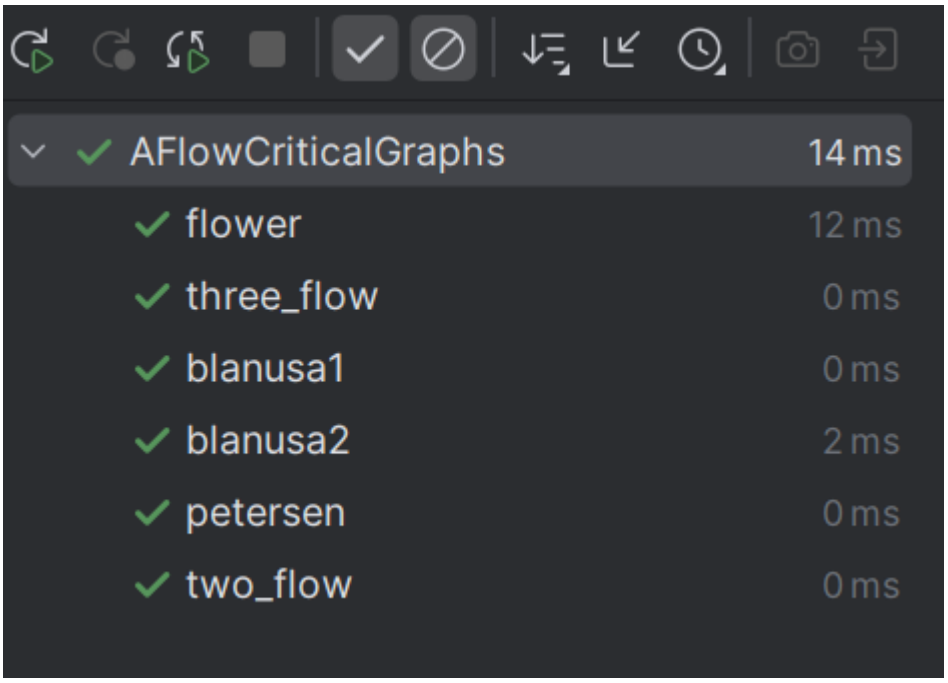
Obr.1 - triviálny backtrack



Obr.2



Obr.3 - pridanie komplexných grafov



Obr.4 - aktuálny algoritmus