

# EVO Aplikované evoluční algoritmy

## Graph Coloring Problem

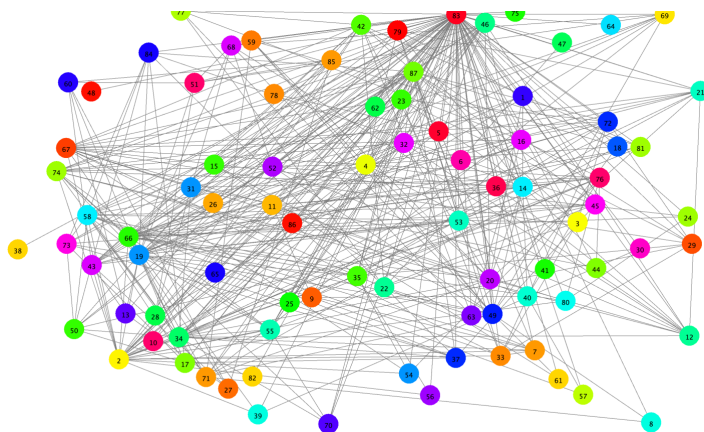
Tadeáš Kachyňa  
xkachy00@stud.fit.vutbr.cz

### Popis problému

Problém barvení grafů by se dal popsat jako problém nalezení takových barev vrcholů, aby dva sousedící vrcholy nesdílely stejnou barvu. Víme, že se jedná o NP-úplný problém, takže nelze efektivně najít jeho optimální řešení. Zadání jsem si zvolil, protože mi přišlo jako jedno z nejzajímavějších a chtěl by se o algoritmu samotném dozvědět více. Barvení grafů se těší mnoha praktickým aplikacím i teoretickým výzvám. Pro příklad užití můžu zmínit známou a populární skládačku Sudoku. Kromě klasických typů problémů lze na graf či barvy přiřadit různá omezení. Barvení grafů je taktéž stále velmi aktivní oblastí výzkumu.

### Implementační prostředí

Jako implemetační prostředí jsem zvolil jazyk Python 3 s podporou knihoven `numpy` pro rychlejší realizaci výpočtů a `matplotlib` pro případné vykreslení grafů (například fitness funkce). Při implementaci genetického algoritmu se budu inspirovat již vytvořeným algoritmem v jazyce C od dr. Bidla<sup>1</sup>, který mi byl v zadání doporučen. Taktéž mi bude inspirací algoritmus<sup>2</sup> přímo řešící tento problém v jazyce Java. Narozdíl od jiných implementací má i pěkné vizuální zpracování, které můžete vidět na obrázku níže.



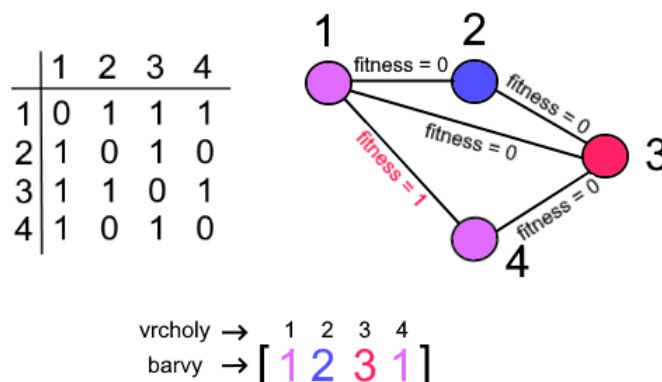
Obrázek 1: Algoritmus realizující i vizuální podobu.

<sup>1</sup>[http://www.fit.vutbr.cz/~bidlom/BIN/BIN\\_GA.tar](http://www.fit.vutbr.cz/~bidlom/BIN/BIN_GA.tar)

<sup>2</sup><https://github.com/soroushj/graph-coloring-genetic-algorithm>

## Způsob řešení

Jak jsem již zmínil v předchozím odstavci, plánuji provádět experimenty na tomto algoritmu v jazyce Python. Algoritmus nejprve inicializuje počáteční populaci. Každý chromozom zde bude vyjádřen vektorem čísel, kde každé číslo bude reprezentovat nějakou barvu. Taktéž bude vygenerována matice vrcholů, kde číslo 1 bude znamenat, že jsou vrcholy spojené. Protože je graf neorientovaný, tak bude matice symetrická. Abych dokázal dané jedince v populaci ohodnotit, tak potřebuji fitness funkci. Ta ohodnotí negativně všechny konflikty barev, které následně sečte. Nejlépe ohodnocený uzel bude mít hodnotu fitness 0.



Obrázek 2: Jednoduchý graf demonstrující fitness funkci a taktéž znázornění vrcholů v matici.

Následně pomocí selekce (*rulety nebo turnaje*) vyberu z populace několik jedinců s vysokou zdatností a aplikuji na ně genetické operátory - *mutaci, křížení* čímž vytvořím novou generaci jedinců, kterou opět ohodnotím fitness funkcí. Takto probíhá algoritmus dokud nenajdu nejlepší řešení nebo nedosáhnu určitého počtu generací dané podmínkou.