

---

# Y33ZUI – Stochastická optimalizace

**Samostatná práce**

*2. úloha*

**Petr Buryan, Y33ZUI - LS 2010**

---

## Problém - Správa komunikací

Toto cvičení předpokládá základní znalost Javy a NetBeans IDE.

### Zadání:

Reorganizujeme správu komunikací a je nutné určit, ve kterých městech je zapotřebí zřídit stálé základny tak, aby tyto základny pokrývaly celou síť cest. Najděte pomocí **základního genetického algoritmu** co nejmenší množinu měst takových, že **všechny silnice** začínají (končí) alespoň v jednom městě z této množiny. Řešení, které by nepokrývalo jakoukoli cestu je tedy nepřijatelné. Úloha je známá jako minimální vrcholové pokrytí grafu (*Minimal Vertex Cover*).

K implementaci algoritmu využijte knihovnu, kterou obdržíte na cvičení. Knihovna obsahuje základní funkce potřebné k načtení, vizualizaci a další práci se zkoumaným problémem. Při řešení problému vytvořte svou vlastní třídu dědící od třídy *AbstractVCCover*. V ní jsou již k dispozici ohodnocovací metody *getNumberOfRoadsCovered* a *getNumberOfCitiesUsed*. K dispozici je i třída *GAOperators*, která obsahuje základní genetické operátory křížení a mutace. Tu můžete využít při tvorbě vlastního genetického algoritmu.

Výstupem Vašeho algoritmu bude pole indexů měst vybraných jako základny (ty se následně vykreslí do předložené mapy). Použijte připravené knihovny uživatelského rozhraní. Můžete se inspirovat třídou *RandomVCCalculator*, která provádí jednoduché náhodné hledávání zadané skupiny měst.

### Ovládání vizualizačního nástroje

Stejně jako v předchozí úloze.

### Pokyny k vypracování:

1. Stáhněte si soubor archivu s distribucí a tento soubor rozbalte.
2. V NetBeans IDE otevřete projekt z místa, kam jste jej rozbalili.
3. Pokuste se spustit ukázkou v metodě *Main.java*, načíst mapu ze souboru *cities.csv* a spustit náhodné hledání cesty mezi zvolenými 2 body.
4. Projděte si **dokumentaci** (javadoc).
5. Vytvořte nový objekt **dědící od třídy *AbstractVCCover*** a do něj vyplňte příslušnou metodu tak, aby plnila požadovanou funkci (hledání min. vrcholového pokrytí pomocí SGA). Inspirovat se můžete třídou *RandomVCCalculator*. Využít můžete i metody realizující základní operace genetického algoritmu, které jsou obsažené v třídě *GAOperators*.

### Pokyny k odevzdání:

Úloha se osobně odevzdává a předvádí cvičícímu na cvičení k tomu určeném.

---

## Poznámky

Předkládaná knihovna je primárně míněna jako pomůcka k výuce. Neošetřuje tedy zcela jistě všechny možné situace, které během obsluhy GUI mohou nastat a zřejmě i obsahuje chyby. Za odhalení a upozornění příp. opravení jakékoli chyby budou autoři cvičení rádi. Z tohoto důvodu je k dispozici i zdrojový kód, který je možné volně adaptovat.