

# Demonštrácia jednotlivých krokov genetického algoritmu

## Technická správa

Peter Lacko  
xlacko06@stud.fit.vutbr.cz

10. prosince 2014

## 1 Genetické algoritmy

Genetické algoritmy (GA) je označenie algoritmov, ktoré sa pokúšajú nájsť najlepšie riešenie daného problému simulovaním procesu prirodzeného výberu v prírode (evolúcie). Nachádzajú uplatnenie v mnohých typoch riešených úloh, predovšetkým však u problémov s príliš veľkým prehľadávacím priestorom, u obtiažne definovaných problémov resp. tam kde nie je dostupná alebo možná matematická analýza, či v prípadoch keď tradičné metódy zlyhávajú. Príkladom sú optimalizačné úlohy, metódy strojového učenia či automatické programovanie (tiež zvané i Genetické programovanie).

Samotný genetický algoritmus pozostáva z nasledujúcich krokov :

1. Náhodne vygeneruj počiatočnú populáciu,
2. Zisti ohodnotenie každého jedinca z populácie,
3. Vypočítaj pravdepodobnosť výberu pre každého jedinca z populácie na základe jeho ohodnotenia (proporčne vzhľadom k celej populácii),
4. Aplikuj genetické operátory na vytvorenie novej populácie z populácie pôvodnej (reprodukcia),
5. Ak nebola splnená podmienka ukončenia, pokračuj bodom 2.

### 1.1 Reprodukcia

Reprodukcia typicky pozostáva z výberu rodičov, rekombinácie, mutácie a vloženia nových potomkov do pôvodnej populácie, analogicky k procesu v prírode.

**Výber rodičov.** Pre výber rodičov existujú rôzne metódy, popíšem tu tri z nich použité v aplikácii:

- Algoritmus rulety – vygeneruje náhodné číslo v rozsahu  $[0,1)$ , a na základe jeho hodnoty sa vyberie rodič z populácie. Jedinci s vyšším ohodnotením majú teda vyššiu pravdepodobnosť výberu. Proces opakujeme kým nezískame požadovaný počet rodičov.
- Algoritmus výberu elity – z populácie sa vyberú jedinci s najlepším ohodnotením
- Algoritmus turnaja – z populácie je vybraný istý počet jedincov a ten s najlepším ohodnotením sa vyberie ako rodič. Iteratívne opakujeme kým nezískame požadovaný počet rodičov.

**Rekombinácia.** Produktom rekombinácie je nová populácia jedincov vzniknutých náhodným krížením chromozómov práve dvoch náhodne vybraných rodičov, tj. vzájomnou výmenou častí chromozómov medzi jedným alebo viacerými bodmi kríženia.

**Mutácia.** Mutácia je náhodná zmena génu v chromozóme jedného jedinca populácie (napr. pričítaním malej hodnoty či invertovaním bitu). Využíva sa na zachovanie rôznorodosti populácie a napomáha vyhýbaniu sa lokálnemu maximu do ktorého môže GA konvergovať. Mutácia môže byť prevedená invertovaním bitu, pričítaním malej hodnoty k chromozómu, vynásobením hodnoty chromozómu a pod.

**Tvorba novej populácie.** Nová populácia sa vytvára vložením istého množstva novo vzniknutej populácie do pôvodnej populácie. Veľkosť novej populácie môže byť rovnaká, ale i väčšia či menšia ako pôvodná. Záleží taktiež na počte jedincov ktorí budú v populácii nahradení.

## 2 Knapsack problém

Na demonštráciu jednotlivých krokov GA som zvolil tzv. 0/1 Knapsack problém. Je to kombinačná úloha u ktorej máme k dispozícii určité množstvo položiek, kde každá položka má svoj objem a hodnotu, a rucksak (knapsack) ktorého objem je menší ako objem vš. položiek dohromady.

Cieľom je vybrať také položky pre vloženie do knapsacku, ktorých celkový objem je menší, nanajvyš rovný objemu knapsacku a súčasne ich hodnota čo najvyššia. Existuje mnoho variácií problému a spôsobov jeho riešenia. U varianty 0/1 môže byť každá položka z množiny položiek umiestnená v knapsacku nula alebo jedenkrát. Na knapsack problém je možné pozerať aj ako na problém pridelovania obmedzeného počtu zdrojov.

## 3 Popis činnosti aplikácie

V demonštračnej aplikácii je obsah knapsacku reprezentovaný jedným bytom, kde hodnoty jednotlivých bitov značia príslušnosť danej položky v knapsacku (1 – položka sa nachádza v knapsacku, 0 – nenachádza). Maximálny počet položiek v knapsacku je teda 8, čo nám dáva celkovo 256 rôznych zložení knapsacku. Je teda zjavné že na výpočet optimálneho zloženia knapsacku nie je GA v tomto prípade najvhodnejšie riešenie, avšak výborne poslúži na ilustráciu jeho činnosti a to najmä vďaka svojej názornosti.

Objem a hodnota položiek sa generuje náhodne pri spustení programu (je možné generovať i manuálne po spustení) pričom hodnota i objem každej položky sa pohybuje v rozpätí 0 – 256 a maximálny objem knapsacku je daný ako polovica celkového objemu všetkých položiek. Počiatočná populácia sa generuje náhodne pričom je možné zvoliť jej veľkosť a to v rozmedzí 2–10. Do počiatočnej populácie sú vybrané iba také kombinácie položiek (jedinici), ktorých celkový objem je menší alebo rovný maximálnemu objemu knapsacku. Populácia má v každej generácii rovnaký počet jedincov.

Podobne je možné zvoliť i počet cyklov reprodukcie a to medzi 1 – 50, čo je zároveň jediná ukončovacia podmienka algoritmu. Počet rodičov je stanovený ako  $\lfloor |population|/2 \rfloor$  a je možné si zvoliť jeden z troch spôsobov ich výberu popísaných vyššie.

V aplikácii je možnosť výberu medzi jedno, dvoj a trojbodovým krížením a uniformným krížením. U prvých troch možnosti sú body kríženia vybrané náhodne, avšak pre všetkých jedincov novo vytvorenej populácie sú rovnaké. Pravdepodobnosť výberu prvého i druhého rodiča je rovnaká pre všetkých rodičov (tzn. nový jedinec môže vzniknúť i krížením dvoch rovnakých rodičov). V prípade že objem nového jedinca (kombinácia položiek) je väčší ako objem knapsacku, náhodne sa vyberie bit s hodnotou 1 a invertuje na 0. Proces sa opakuje kým celkový objem nie je menší ako objem knapsacku.

Mutácia je vykonávaná invertovaním náhodného bitu v náhodne vybranom chromozóme, pričom invertovaný môže byť len bit, ktorého zmena nezväčší celkový objem položiek nad maximálny objem. Typicky je mutácia prevedená iba s veľmi malou pravdepodobnosťou, no v tejto aplikácii je mutácia kôli názornosti prevedená v každej iterácii algoritmu.

Počet nahradených jedincov starej populácie je taktiež možné si ľubovoľne zvoliť a to od 1, kedy je nahradený iba člen s najnižšou hodnotou až po plné nahradenie starej populácie novou.

## 4 Spustenie a ovládanie aplikácie

Programovacím jazykom aplikácie je Java vo verzii 7, grafické rozhranie je vybudované pomocou knižnice Swing, štandardne prítomnou v inštalácii Javy. Archív obsahuje súbory `dokumentace.pdf`, `GADemo.jar` a adresár `src` so zdrojovými textami aplikácie. Pre správny beh aplikácie je potrebné ju spustiť na počítači s grafickým užívateľským rozhraním a nainštalovanou Javou vo verzii 7 a vyššie.

Po rozbalení archívu spustíte aplikáciu príkazom

```
java -jar GADemo.jar
```

čím sa otvorí hlavné okno programu. Horná časť okna obsahuje stručný popis knapsack problému a súčasné hodnoty možných položiek v knapsacku. Tieto hodnoty je možné zmeniť po kliknutí na tlačítko *Vygeneruj nový obsah*. Dolná časť okna obsahuje prvky pre nastavenie parametrov genetického algoritmu. Hlavné okno a východzie hodnoty parametrov sú znázornené na obrázku 1. Po kliknutí na tlačítko *Start simulation* sa otvorí simulačné okno v prvom kroku, tj. výber rodičov. Je možné spustiť viacero simulácií s rovnakými hodnotami položiek a rôznymi parametrami a sledovať aký vplyv má výber parametrov na dosiahnuté výsledky.

V simulačnom okne je možné sa posúvať medzi jednotlivými krokmi algoritmu (výber rodičov, rekombinácia, mutácia, tvorba novej populácie) pomocou tlačítiek *Next* a *Prev*. a pomocou *Next cycle* resp. *Prev cycle* zasa prechádzať algoritmom po iteráciách. *Skip to results* slúži na okamžité zobrazenie výsledkov algoritmu a *Exit simulation* na ukončenie simulácie. Súčasný krok a číslo iterácie sú zobrazené v prvom hornom rohu a relevantné hodnoty v jednotlivých krokoch simulácie sú prehľadne zobrazené v tabuľkách tak, ako je znázornené na obr. 2.

## 5 Záver

Cieľom projektu bolo vytvoriť program na demonštráciu jednotlivých krokov genetického algoritmu. Na tento účel bol vybraný 0/1 knapsack problém, čo je kombinačná úloha ktorej cieľom je naplniť knapsack položkami s celkovým objemom nepresahujúcim kapacitu knapsacku pri ich čo najvyššej hodnote. Výstupom projektu je plne funkčná aplikácia s grafickým užívateľským rozhraním, ktorá zobrazuje príslušné hodnoty v jednotlivých krokoch algoritmu a umožňuje medzi nimi ľubovoľne prechádzať.

### Knapsack problém

Knapsack problém je úloha, kde máme k dispozícii určité množstvo položiek, každú s určitým objemom a hodnotou, a knapsack kam chceme tieto položky uložiť. Cieľom je vybrať také položky aby ich celkový objem nebol väčší ako objem knapsacku a v súčte mali čo najvyššiu možnú hodnotu.

**Max objem knapsacku: 465**

Položka	Objem	Hodnota
1.	24	81
2.	71	5
3.	61	95
4.	27	6
5.	237	121
6.	239	88
7.	216	17
8.	55	101

Vygeneruj nový obsah

### Nastavenie genetického algoritmu

Veľkosť populácie:

Počet cyklov reprodukcie:

Výber rodiča:

Body rekombinácie:

Počet nahradení:

Exit program      Start simulation

Obrázek 1: Hlavné okno aplikácie.

### Parent Selection

Iterácia 1  
Roulette selection

Populácia

ID	Populácia	Objem	Hodnota	Proporčne	Kumulatívne
1	00100001	245.0	199.0	0.08	0.08
2	11001001	506.0	289.0	0.12	0.20
3	10000111	465.0	363.0	0.15	0.35
4	00001110	534.0	302.0	0.12	0.48
5	01000011	389.0	151.0	0.06	0.54
6	01011001	611.0	284.0	0.12	0.65
7	10100100	469.0	507.0	0.21	0.86
8	01000110	521.0	332.0	0.14	1.00

Vybraní rodičia

Náhodná hodnota	ID	Rodič	Hodnota
0.01	1	00100001	199.0
0.66	7	10100100	507.0
0.54	5	01000011	151.0
0.86	7	10100100	507.0

< Prev (Reinsertion)      Next (Recombination) >

<< Prev cycle      Next cycle >>

Exit Simulation      Skip to results

Obrázek 2: Ukážka simulácie v prvej iterácii pri výbere rodičov