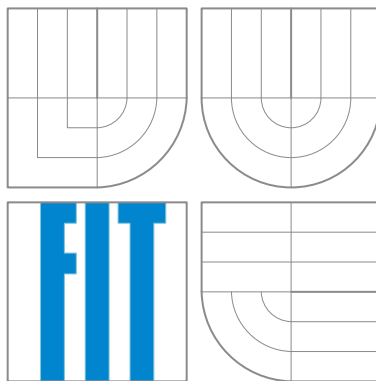


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta informačních technologií



Soft Computing

Java (Aplikace)

Demonstrace jednotlivých kroků GA
2014/2015

10. prosince 2014

Autor: Bc. Radim Kubiš, xkubis03@stud.fit.vutbr.cz

Úvod

Aplikace pro demonstraci jednotlivých kroků genetického algoritmu je napsána v jazyku Java a má grafické uživatelské rozhraní. Používá knihovnu uživatelských prvků *Swing*.

Reprezentace chromozomu

Chromozomy jsou v programu reprezentovány jako seznamy (**ArrayList**) jejich jednotlivých složek/genů. Pro každý typ chromozomu existuje zvláštní třída, která uchovává i jeho další atributy (např. hodnotu fitness). Lze využít 5 typů chromozomů:

- binární (**BinaryChromosome**),
- celočíselný (**IntegerChromosome**),
- reálných čísel (**DoubleChromosome**),
- řetězcový (**StringChromosome**),
- permutační (**PermutationChromosome**).

Populace chromozomů mají vlastní třídy obsahující seznam (**ArrayList**) jedinců, velikost a typ chromozomů. Každá populační třída umožňuje pseudonáhodně generovat složení chromozomů různé délky.

Ohodnocení populace

Typy chromozomů uvedené výše dokáže program ohodnotit minimálně jednou fitness funkcí. Pro binární chromozomy to jsou *OneMax*, *ZeroMax* a binární reprezentace celého kladného čísla (počet správně umístěných nul a jedniček na jednotlivých pozicích chromozomu). Celočíselné chromozomy a chromozomy s reálnými čísly lze hodnotit na základě neklesající, resp. nerostoucí posloupnosti jejich jednotlivých složek. Chromozom řetězcový, obsahující posloupnost uživatelem zadaných řetězců, se hodnotí pouze pomocí počtu v něm obsaženého konkrétního řetězce. U permutačního chromozomu je nutné zadat očekávaný výsledek, z něhož se zjišťuje počet správně umístěných genů.

Ukončovací podmínka

Podmínkou pro ukončení genetického algoritmu je buď nalezení správného řešení, nebo dosažení maximálního zadaného počtu iterací jeho cyklu, když dosud nebylo správné řešení nalezeno.

Program demonstruje postupně průběh jednotlivých kroků jednoho cyklu algoritmu a neumožňuje pokračovat dále pouze pokud je nalezeno správné řešení. Ukončení po dosažení maximálního nastaveného počtu iterací se používá ve funkčním příkladu *OneMax* problému.

Výběr rodičů z populace

Existuje několik způsobů, jak vybrat z populace množinu jedinců, kteří se stanou rodiči nových chromozomů. V programu jsou implementovány čtyři:

- ruleta,
- pravděpodobnostní turnaj,
- výběr elity,
- turnaj.

U většiny z nich je nutné provést přepočet fitness hodnoty podle proporce, nebo podle pořadí jedinců v populaci. Kvalitnější jedinci potom mají při výběru větší pravděpodobnost stát se rodiči. Program umožňuje volbu parametrů, které ovlivňují tento výběr (např. počet vybíraných rodičů, selektivní tlak).

Tvorba potomků (rekombinace)

Potomci vzniknou prováděním různých operací nad dvěma rodiči. Možnosti použití jednotlivých druhů rekombinace jsou závislé na reprezentaci chromozomu. Přehled možností rekombinace pro jednotlivé typy chromozomů v programu:

- **binární:** jednobodové, vícebodové, uniformní,
- **celočíselné:** jednobodové, vícebodové, uniformní,
- **reálné:** jednobodové, vícebodové, uniformní, aritmetické, heuristické,
- **řetězcové:** jednobodové, vícebodové, uniformní,
- **permutační:** indexová tabulka, PMX.

Opět mají metody rekombinace možnosti nastavení několika parametrů, jedná se především o volbu pevného/náhodného místa rekombinace, počet míst rekombinace apod.

Mutace potomků

Mutace spočívá v provedení umělých změn náhodně vybraného genu náhodně vybraného jedince. Hlavní parametr, který ovlivňuje mutaci, je její pravděpodobnost. Program umožňuje nastavit rozsah pravděpodobnosti od 0 do 100 % s krokem 0,1. Menší krok bohužel nešlo v grafické komponentě **JSpinner** nastavit.

Všech pět typů chromozomu lze zmutovat. V případě binárního chromozomu se neguje hodnota na určité jedné pozici. Celočíselný chromozom mutuje záměnou jednoho genu za jeho jinou alelu. U chromozomu s reálnými čísly se k některému z jeho genů přičte, resp. odečte velmi malá hodnota (maximálně 10 % původní hodnoty). Řetězcový chromozom také mutuje záměnou genu za jinou alelu. Pro mutaci permutačního chromozomu je potřeba zvolit dvě pozice, na kterých dojde k prohození genů.

Tvorba nové populace

Po rekombinaci a mutaci se provede výběr jedinců do nové populace. Lze vybírat z původní populace a zároveň i z množiny potomků. Nová populace může opět vznikat mnoha způsoby, některé se podobají způsobům výběru rodičů. Uživatel má na výběr tyto:

- **generační:** úplná náhrada původní populace potomky,
- **inkrementální:** nejhorší jedinec původní populace je nahrazen nejlepším potomkem,
- **úplná náhrada:** stejná jako generační,
- **uniformní náhrada:** potomci nahradí náhodně vybrané jedince původní populace,
- **elitní náhrada:** celá množina potomků nahradí stejně velkou množinu nejhorších jedinců původní populace,
- **turnajová náhrada:** z náhodně vybraného rodiče a náhodně vybraného potomka se do nové populace začlení ten s vyšší fitness.

Spuštění programu

Ve složce `app` je binární `JAR` soubor, který se při správně nastaveném systému spustí poklikáním. Pokud se nespustí tímto způsobem, lze program spustit z příkazové řádky:

```
java -jar <název souboru>.jar
```

nebo otevřít projekt v *NetBeans IDE*.

Funkčnost byla ověřena na studentském serveru `merlin` a na počítačích v CVT (Windows i Linux).

Ovládání programu

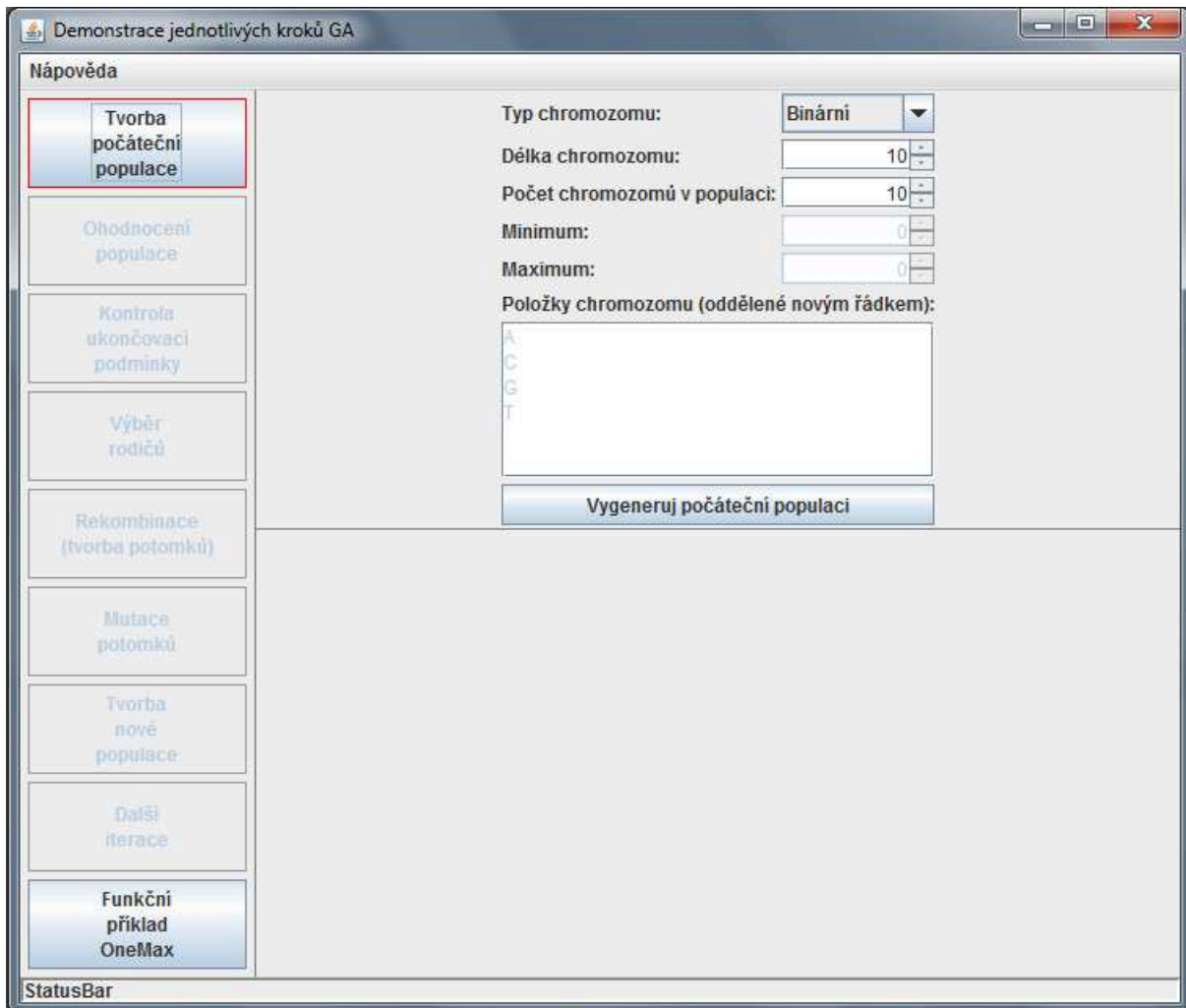
Aplikace rozděluje cyklus genetického algoritmu na 8 jeho kroků:

1. Tvorba počáteční populace.
2. Ohodnocení populace.
3. Kontrola ukončovací podmínky.
4. Výběr rodičů.
5. Rekombinace (tvorba potomků).
6. Mutace potomků.
7. Tvorba nové populace.
8. Skok na další iteraci (bod 2).

Začíná se tvorbou počáteční populace, která je k dispozici stále. Ostatní kroky jsou dostupné pouze po provedení kroků předchozích. Lze se však libovolně vysoko vrátit k jakémukoliv jinému kroku a po jeho zopakování (např. s jinými parametry) pokračovat opět směrem dolů.

Dokud není stisknuto tlačítko „Další iterace“, pracuje program stále s původní populací jedinců. Až při použití tohoto tlačítka se provede náhrada původní (předchozí) populace nově vygenerovanou za použití potomků.

Funkční příklad OneMax je demonstrací řešení OneMax problému, kdy je požadováno dosažení binárního chromozomu obsahujícího pouze bity v hodnotě 1. Uživatel může nastavit všechny dostupné a potřebné parametry, díky kterým zjistí, jak moc ovlivňují nalezení žádoucího řešení.



Obrázek 1: Grafické uživatelské rozhraní aplikace.

Použité softwarové prostředky

- NetBeans IDE 8.0.2
- Java SE 1.8.0_25

Použité informační zdroje

- Přednáškové materiály předmětů BIN, EVO, SFC
- Návody, rady a dokumentace na internetu