

Gramatická evoluce a softwarový projekt AGE

Adam Nohejl

Matematicko-fyzikální fakulta
Univerzita Karlova v Praze
<http://nohejl.name/>

4. 4. 2010

Poznámka: Prezentace založené na variantách těchto slajdů proběhly na MFF UK 10. 11. 2009 v rámci semináře Rozpoznávání a syntaktická analýza a 3. 3. 2010 v rámci přednášky Evoluční algoritmy II (bez úvodní sekce o EA).

Přehled

Evoluční algoritmy, genetické programování

Gramatická evoluce

Softwarový projekt AGE: Algorithms for Grammatical Evolution

Postřehy a nápady

Shrnutí

Motivace: optimalizační problém

Složitý optimalizační problém:

1. prohledávání velkého prostoru s lokálními extrémy,
2. omezený čas.



(Ilustrace z J. R. R. Tolkien: *The Hobbit*, reprodukce v nízké kvalitě v rámci *fair use*.)

Motivace: evoluční algoritmy

Jak *obecně* řešit optimalizační problémy:

1. pro něž není známý algoritmus,
2. pro něž je známý algoritmus pomalý,
3. pro něž se nám nechce nad algoritmem přemýšlet.

Nepožadujeme-li za každou cenu přesné (optimální) řešení, nabízí se použití nějaké **metaheuristiky**: heuristické metody, nebo sady metod, řešící širokou třídu problémů.

Jedna z možností jsou **evoluční algoritmy**.

Evoluční algoritmy, evoluční výpočty: idea

Inspirace Darwinovou evoluční teorií:

1. Pokusy o simulace evoluce na počítači
2. Nápad využití simulované evoluce jako optimalizačního algoritmu:

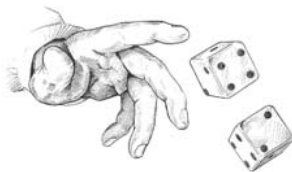
Idea: Zvolíme-li vhodnou reprezentaci kandidátů na řešení, můžeme je křížit, mutovat a podrobovat přirozenému výběru. Věříme-li, že to funguje v přírodě, proč by to nemělo fungovat v počítači. . .
(Lawrence J. Fogel '60s, John Holland '70s.)



Evoluční algoritmus: náčrt kostry

Pracujeme s populací velikosti p jedinců, kandidátů na řešení, po nejvýše n generací:

1. Inicializuj **náhodně** populaci P_0 velikosti p . – **Initilisation**.
2. Opakuj pro $i = 0 \dots n - 1$ nebo dokud není nejlepší řešení dost dobré:
 1. Ohodnoť každého jedince v populaci P_i . – **Fitness evaluation**.
 2. P_{i+1} necht' je pro začátek prázdná.
 3. Opakuj pro $j = 1 \dots p$:
 1. Na základně ohodnocení vyber **náhodně** dva jedince z P_i . – **Selection**.
 2. S **pravděpodobností** p_c je zkříž. – **Crossover**.
 3. S **pravděpodobností** p_m na nich proved' mutaci. – **Mutation**.
 4. Přidej tyto jedince do populace P_{i+1} – **(Replacement.)**



Problém EA č. 1: Co je to fitness?

Fitness má v evolučních algoritmech centrální úlohu, je to pojítka mezi konkrétní aplikací a obecným algoritmem. Co je tedy fitness?

- ▶ **V přírodě:** Úspěšnost jedince v boji o přežití a při reprodukci: počet přeživších potomků. – **Tautologická definice!**
- ▶ **V počítači:** Námi zvolené ohodnocení jedince. – Nabývá významu až ve spojení se selekční metodou a případným škálováním.

Přestože existuje mnoho „vyzkoušených“ technik, volbu metody ohodnocení, škálování a selekce nelze nikdy oddělit od aplikace.

Problém EA č. 2: Reprezentace a operátory

Jedince je třeba nějak reprezentovat a s těmito reprezentacemi umět dále pracovat.

Kdy pracujeme přímo s jedinci?

- ▶ při **ohodnocování** (přiřazení *fitness*),
- ▶ při **aplikaci operátorů** (inicializace, mutace, křížení).

Tyto elementy algoritmu jsou tedy nutně alespoň zčásti závislé na aplikaci.

Genetické algoritmy

60. léta: John Holland; později David Goldberg.

Reprezentace: Jedinci jsou vektory bitů (čísels).

Operátory: Náhodná inicializace, jednobodové křížení, bitová mutace.

Zbývá: Interpretovat binární vektory.



A 20x20 grid of binary digits (0s and 1s) representing a population of individuals. The grid is composed of 20 rows and 20 columns. Some digits are highlighted in red, indicating a specific subset of the population or a specific bit position across individuals. The red digits are scattered throughout the grid, with a notable concentration in the middle rows and columns.

Genetické programování

80. léta: prvopočátky; 90. léta: John Koza.

Reprezentace: Jedinci jsou stromy programů (v podmnožině LISPu).



Operátory: Složitější operace na těchto stromech. (Jsou dané sady funkcí a terminálů.)

Zbývá: Interpretovat výstup.

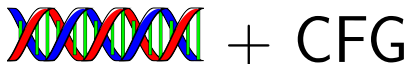
Genetické programování s gramatikou

90. léta: první pokusy o užití gramatik v GP.

- ▶ **Různé motivace:** rozšíření GP, přidání flexibility nebo naopak omezení GP.
- ▶ **Různé prostředky:** doplnění vs. nahrazení stromové struktury.

2003: Michael O'Neill, Conor Ryan: *Grammatical Evolution*:

Reprezentace: Inspirace přírodou ~~přírodou~~ genetikou a molekulární biologii: Oddělení genotypu a fenotypu, ovšem pomocí *formální gramatiky*.



Operátory: Stejně jako v obyčejném GA. (Zkusme to!)

Zbývá: Interpretovat program a jeho výstup.

Přehled

Evoluční algoritmy, genetické programování

Gramatická evoluce

Softwarový projekt AGE: Algorithms for Grammatical Evolution

Postřehy a nápady

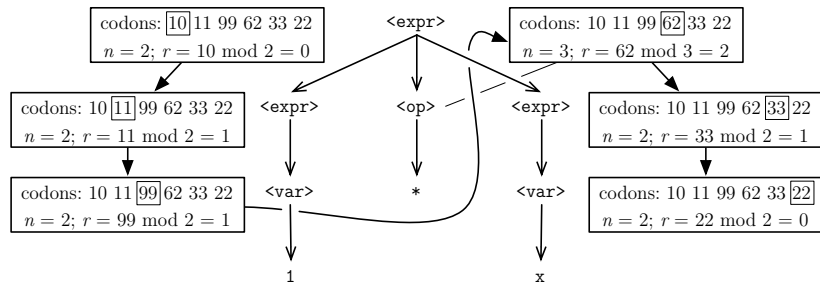
Shrnutí

GE: Příklad mapování

Kodony: 10 11 99 62 33 22 (případné další by byly ignorovány)

Počet pravidel: n

Pravidlo: r



Gramatika:

<code><expr></code>	<code>::= (<expr> <op> <expr>) <var></code>
<code><op></code>	<code>::= + - *</code>
<code><var></code>	<code>::= x 1.0</code>

GE: Zhodnocení

Co jsme získali?

- ▶ Větší volnost formy (a snadnost její specifikace) oproti GA a GP.
- ▶ Snadnou implementaci operátorů nezávisle na aplikaci.

Problémy?

1. Někteří jedinci jsou stále neplatní.
2. Nevyužívá se nutně celá informace jednotlivých kodonů.
3. Nevyužívají se nutně všechny kodony.

GE: Druhý pohled na problémy

1. Někteří jedinci jsou stále neplatní.

Když dorazíme na konec chromosomu, pokračujeme od začátku, jako by byl kruhový, ne lineární. Opakujeme nejvýše n -krát. – **Wrapping** (zarolování).

2. Nevyužívá se nutně celá informace jednotlivých kodonů.

Díky rolování můžeme kodon interpretovat vícekrát: vzhledem k různým sadám pravidel.

3. Nevyužívají se nutně všechny kodony.

Nepoužité kodony, i různé části informace obsažené v těch použitých, se mohou uplatnit v dalších generacích po aplikaci genetických operátorů.

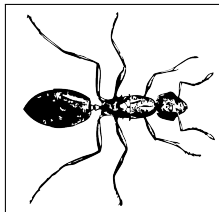
Paralely s „přírodou“: *gene overlapping, genetic code degeneracy.*

GE: Základní postřehy

- ▶ Nešikovnost/pomalost implementací (modul existujícího EA prostředí, interpretace).
- ▶ Nedaří se reprodukovat pěkné výsledky autorů.
- ▶ Stále není úplně jasné, jaký efekt mají (mít) operátory a jiné parametry.

GE: Aplikace na problém Santa Fe ant trail (1)

Cíl: Navigovat umělého mravence aby našel všechno jídlo ležící na čtvercové mřížce do časového limitu.

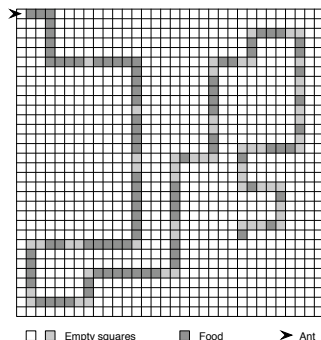


Akce:

- ▶ LEFT otočí mravence doleva o 90° bez pohybu,
- ▶ RIGHT otočí mravence doprava o by 90° bez pohybu,
- ▶ MOVE pohne mravencem vpřed na pole ve směru, kterým se dívá, a sní případné jídlo na něm,
- ▶ FOOD-AHEAD testuje jídlo na poli, na které se mravenec dívá.

GE: Aplikace na problém Santa Fe ant trail (2)

Všechny akce vyjma testu zaberou jednotku času. Mřížka je nabalená kolem toroidu, aby mravenec nespadl z okraje. Počáteční konfigurace Santa Fe ant trail:



Pozorování: Posloupnost mezer a zatáček má stoupající náročnost.

GE: Aplikace na problém Santa Fe ant trail (3)

Různé gramatiky mohou dávat různé výsledky:

Gramatika 1:

```
<code>          ::= <line> | <code> <line>
<line>          ::= <if-statement> | <op>
<if-statement>  ::= if (foodAhead(h)==1) then <line>
                  else <line> end
<op>            ::= left(h) | right(h) | move(h)
```

Gramatika 2:

```
<prog>          ::= <code>
<code>          ::= <line> | <code> <line>
<line>          ::= <condition> | <op>
<condition>     ::= if (foodAhead(h)==1) then <opcode>
                  else <opcode> end
<op>            ::= left(h) | right(h) | move(h)
<opcode>        ::= <op> | <opcode> <op>
```

Přehled

Evoluční algoritmy, genetické programování

Gramatická evoluce

Softwarový projekt AGE: Algorithms for Grammatical Evolution

Postřehy a nápady

Shrnutí

Cíle softwarového projektu AGE

Východisko: Literatura. Existující implementace.

Cíle:

- ▶ Čistá, ucelená implementace standardních algoritmů.
- ▶ Modularita.
- ▶ Kvalitní dokumentace.
- ▶ Dobře využitelné výstupy.
- ▶ Reprodukovatelné výsledky.
- ▶ Přijatelný výkon.

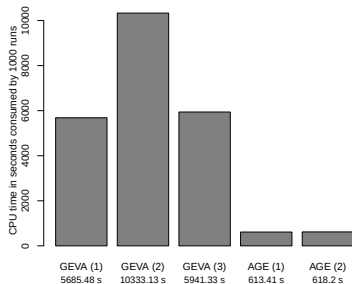
Výsledky softwarového projektu

Cíle splněny. 😊

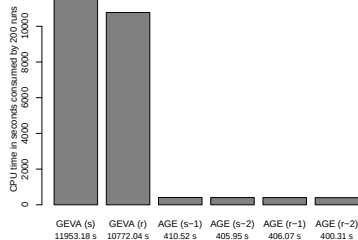
- ▶ **Implementace algoritmů:** Ve větším rozsahu a kvalitě než GEVA. Konsistentnost. Citace zdrojů.
- ▶ **Modularita:** I na uživatelské úrovni. Demonstrovaná implementovanými elementy algoritmů.
- ▶ **Dokumentace:** Pokrývá uživatelské rozhraní včetně API, použité algoritmy i podstatné implementační detaily.
- ▶ **Výstup:** Textový. Mezivýsledky (user feedback), konečné výstupy v XML, statistiky jako textová tabulka.
- ▶ **Reprodukovatelnost:** Viz výše; náhodný generátor; přenositelnost; ukládání paramerů.
- ▶ **Výkon:** Možnost více paralelních běhů. Rychlý interpret (Lua). V testech řádově vyšší výkon než GEVA.

Srovnání výkonu

Benchmarkové aplikace symbolická regrese a „mravenčí cesta“ (Santa Fe ant trail): 10–29krát rychlejší se stejným nastavením. Výsledky stejné nebo lepší.



Symbolická regrese (časy)



Santa Fe ant trail (časy)

Web projektu

Výsledky projektu, včetně softwaru ke stažení, najdete na webu:

`http://nohejl.name/age/`

Přehled

Evoluční algoritmy, genetické programování

Gramatická evoluce

Softwarový projekt AGE: Algorithms for Grammatical Evolution

Postřehy a nápady

Shrnutí

GE: Inicializace

Problém:

Náhodná inicializace nedává dost reprezentativní vzorek různě košatých stromů.

Řešení:

Možnost inspirace GP: Co odpovídá funkcím a terminálům?
Rekurzivní a nerekurzivní pravidla.

GE: Mutace

Jak se mutace projeví ve fenotypu?

- ▶ Efekt „buď a nebo“ (není lokální).
- ▶ Po bitech: Efektivní pravděpodobnost mutace závisí na degeneraci (redundanci).
- ▶ Po kodonech?

GE: Crossover

Problém(?):

„Vlnkový“ efekt. Destruktivní?

Řešení:

Lepší analýza? Možnost inspirace GP?

GE: Možnosti práce s gramatikou

Pozorování: Ekvivalentní gramatiky kódují preferenci určitých struktur a podstatně ovlivňují chování operátorů.

Jak se s tím vypořádat nebo toho využít?

- ▶ Dynamická koevoluce gramatik.
- ▶ Vynucení nějakého předem daného tvaru gramatiky.
- ▶ Obojí.

GE: Proces mapování

Možnosti úpravy:

- ▶ Přímo: Představený algoritmus nemá žádné teoretické opodstatnění.
- ▶ Nepřímo prostřednictvím gramatiky (a velikosti kodonů).

Přehled

Evoluční algoritmy, genetické programování

Gramatická evoluce

Softwarový projekt AGE: Algorithms for Grammatical Evolution

Postřehy a nápady

Shrnutí

Shrnutí

- ▶ **GE je praktická.** – Může nám usnadnit práci v mnoha aplikacích vhodných pro EA.
- ▶ **GE nemá žádnou použitelnou implementaci.** – To už není pravda. 😊
- ▶ **Pěkné výsledky autorů nelze reprodukovat.** – Škoda. Můžeme teď ale zkusit získat jiné, zajímavější.
- ▶ **Užití GE může mít netriviální důsledky.** – Dobře, že jsme si toho aspoň všimli.
- ▶ **Je mnoho, co zlepšovat.** – Mám o čem psát diplomovou práci.
- ▶ Další vývoj projektu můžete sledovat na <http://nohejl.name/age/>.