

GENETIC PROGRAMMING BASED APPROXIMATION

Ondřej Benda

Master Degree Programme (1), FEEC BUT

E-mail: xbenda00@stud.feec.vutbr.cz

Supervised by: Jan Karásek

E-mail: karasek.jan@phd.feec.vutbr.cz

Abstract: This project aims to implement an application that will demonstrate the operating principles and capabilities of genetic programming. Algorithm with the help of genetic operators try to find a suitable function that will approximate the points defined by the user in the coordinate system.

Keywords: Genetic programming, JAVA, approximation, function

1. ÚVOD

Projekt je zaměřen na implementaci aplikace v jazyce JAVA, která bude studentům oboru Telekomunikační a informační technologie v předmětu Teoretická informatika na Fakultě elektrotechniky a komunikačních technologií demonstrovat možné uplatnění genetického programování, přiblížit jeho princip a ukázat výsledky kterých dosahuje. Účelem aplikace je hledání funkce, která bude nejlépe aproximovat uživatelem zadané body v souřadnicovém systému.

2. IMPLEMENTACE

Genetické programování využívá podobných principů jako biologická genetika. Je definovaný chromozom (jedinec), který nese nějaké řešení problému. Celá populace se potom skládá z množiny řešení. V této množině se hledá jedinec, který reprezentuje nejlepší řešení našeho problému. Dostatečně uspokojivé řešení je definováno tzv. Fitness funkcí. Tato funkce ohodnocuje kvalitu každého jedince na základě nějakého atributu. Jestliže jedinec bude představovat například algebraický výraz, tak fitness funkce může být rozdíl mezi funkční hodnotou výrazu pro nějakou hodnotu neznámé a vertikální souřadnicí bodu, který má funkce aproximovat. Pokud v populaci není nalezeno dostatečně uspokojivé řešení, nastane evoluční proces, při kterém jsou použity tři hlavní genetické operátory – křížení, mutace a selekce. Tyto operátory mají za úkol vybírat vhodné páry jedinců a měnit nebo modifikovat části jejich genetické informace (řešení) a vytvářet tak nová řešení, ze kterých vznikne nová populace. Jelikož jsou genetické operátory založeny na náhodných procesech, je možné že výměna nebo modifikace jedinců nepovede k uspokojivému řešení. Proto je vhodný návrh těchto operátorů stěžejní pro konvergenci populace. [1],[2]

2.1. POPULACE

Algoritmus je řízený gramatikou. Každý jedinec v populaci reprezentuje nějaký algebraický výraz. Tento výraz je vyjádřen definovanou gramatikou a uspořádán ve stromu podle významu jednotlivých symbolů. Gramatika se skládá z pravidel tvořených terminálními a neterminálními symboly (funkcemi). Mezi funkce O patří základní algebraické operace.

$$O \in \{+, -, *, /\}$$

Terminály T jsou reprezentovány přirozenými jednocifernými čísly s nulou a neznámou x .

$$T \in \{0, 1, \dots, 9, x\}$$

Maximální hloubka stromu je tři. To znamená, že maximální počet uzlů ve stromu je sedm, z toho mohou být maximálně čtyři terminály a tři funkce. Tím je omezena vyjadřovací schopnost jedince.

2.2. FITNESS FUNKCE

Fitness funkce je funkce, která u každého jedince určuje, jak moc uspokojivé řešení nese. Neboli jaký je rozdíl mezi řešením, které jedinec nese a minimálním požadovaným řešením.

V aplikaci Fitness funkce u každého jedince spočítá součet rozdílů mezi vertikální souřadnicí zadaných bodů a funkční hodnotou výrazu v těchto bodech. Je-li celkový rozdíl menší než dva, je jedinec prohlášen za hledanou funkci a evoluční proces skončí.

2.3. OPERÁTOR KŘÍŽENÍ

Není-li v aktuální populaci nalezeno minimální uspokojivé řešení, jsou použity operátory mutace a křížení. Operátor křížení vyměňuje mezi dvěma zvolenými jedinci část jejich genetické informace neboli část jejich řešení. V aplikaci jsou implementovány dva algoritmy křížení:

- **2P2CH** - V praxi to probíhá tak, že v prvním jedinci reprezentovaným stromem se vybere náhodným způsobem uzel, následně se v druhém jedinci vybere náhodným způsobem druhý uzel (avšak tak, aby maximální hloubka tří zůstala zachována). Takto vybrané podstromy se vymění, čímž ze dvou rodičů vzniknou dva potomci.
- **2P1CH** - V prvním jedinci se vybere náhodným způsobem uzel, v druhém jedinci také. Následně se vybraný podstrom z druhého jedince nahradí podstromem prvního jedince. Takto vznikne jeden potomek ze dvou rodičů.

Operátor křížení nejvíce ovlivňuje konvergenci populace, proto je použit na její větší část. [1] [2]

2.4. OPERÁTOR MUTACE

Operátor mutace postihuje malou část populace. Jeho funkcí je modifikovat část vybraného stromu. Jsou implementovány dva algoritmy mutace.

- **Swap** - Náhodným způsobem vybere podstrom a ten nahradí jiným náhodně vygenerovaným podstromem. Tento algoritmus dokáže modifikovat velikost stromu.
- **Change** – Tento algoritmus mění hodnoty náhodně vybraných uzlů. Dokáže změnit celý podstrom nebo i kterýkoliv samostatný uzel. [1]

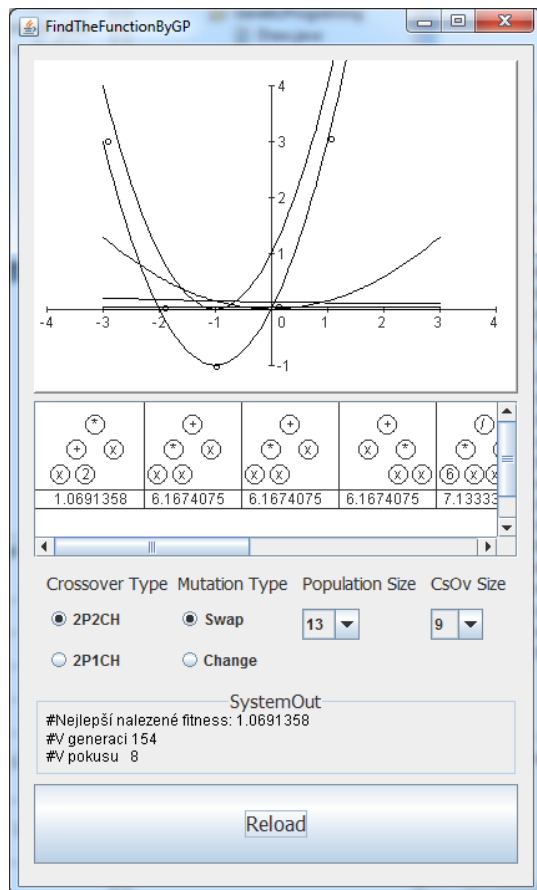
3. APLIKACE

Aplikace se sestává z hlavního panelu, který slouží pro zadávání bodů a vykreslování nalezených funkcí, vedlejšího panelu pro vykreslení jedinců poslední generace, nastavovacích prvků, panelu pro výpis doplňujících informací o výsledku a dvoustavového tlačítka pro spuštění evolučního procesu nebo pro obnovení aplikace do počátečního stavu.

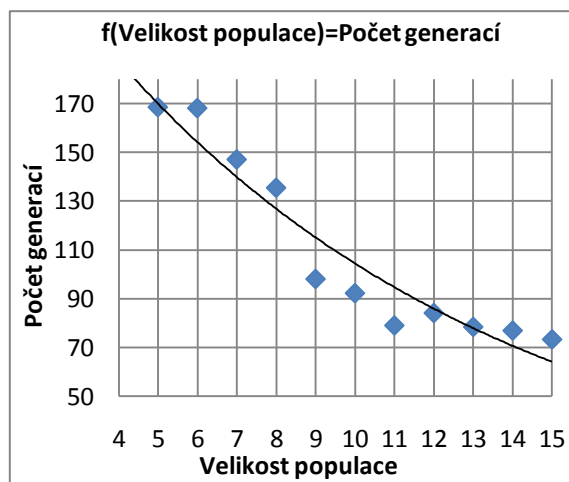
3.1. VÝSTUP

Výstupem programu je několik informací – podoba poslední generace, pořadí generace, ve které byla nalezena výsledná funkce a hlavně podoba výsledné funkce, která se vykresluje v prvním hlavním panelu. Nevykresluje se ale jen ta poslední funkce, ale každou desátou generaci je vykreslována ta aktuálně nejlepší. Protože je za každým tímto vykreslením nastavena časová prodleva 200 ms vytváří evoluční proces názornou animaci toho, jak populace, postupně modifikovaná genetickými operátory, konverguje k hledané funkci. Na obrázku **Obrázek 1: Organizace a vzhled grafického rozhraní** je ukázka hledání funkce $f(x) = x^2 + 2x$. Z výstupu je dobře patrné, že v prvotních populacích byly nalezeny pouhé konstanty, v dalších se objevila klesající přímka, parabola, strmá parabola posunutá po horizontální ose a nakonec hledaná funkce.

Při hledání funkce je dále možné například pozorovat, jakým způsobem se mění pravděpodobnost nalezení funkce při změnách parametrů evolučního algoritmu. Dobře patrná je rychlost nalezení požadované funkce v závislosti na změně velikosti populace. **Obrázek 2:** Závislost počtu generací na velikosti populace ilustruje jmenovanou závislost při měření s rozmístěním bodů odpovídajícím funkci $f(x) = x^2 + 2x$.



Obrázek 1: Organizace a vzhled grafického rozhraní



Obrázek 2: Závislost počtu generací na velikosti populace

4. ZÁVĚR

Aplikace by měla sloužit jako praktická opora k teorii genetického programování vyučovaného v předmětu Teoretická informatika. Pro ještě větší názornost by mohla být rozšířena například o vizualizaci operace křížení na konkrétních jedincích. Pro větší efektivitu algoritmu by bylo nutné rozšířit gramatiku a maximální velikost stromu.

REFERENCE

- [1] B. LANGDON, William, et al. *Studies in Computational Intelligence*. Berlin : Springer, 2008. Genetic Programming: An Introduction and Tutorial, with a Survey of Techniques and Applications , s. 927-1028. ISBN 978-3-540-78292-6.
- [2] MANRIQUE, Daniel, et al. *Grammar Based Crossover Operator in Genetic Programming*. Berlin : Springer, 2005. 252-261 s.