

USING GENETIC PROGRAMMING IN MEDIAN SEARCH

Ondřej JAKŠÍK, Master Degree Programme (5)
Dept. of Computer Systems, FIT, BUT
E-mail: xjaksi00@stud.fit.vutbr.cz

Supervised by: Ing. Lukáš Sekanina, Ph.D.

ABSTRACT

This paper focuses on the application of genetic programming in median search. The author conducted several experiments and compared his results with those obtained by conventional methods.

1 ÚVOD

Genetické programování (GP) je jedním z nejpobulárnějších evolučních algoritmu (EA) posledních let [3]. Úkolem GP je najít program, který řeší daný problém. Během procesu hledání řešení se mění počáteční populace kandidátních řešení do výsledné populace, ve které se nachází hledané řešení. Tento proces připomíná evoluci, jak ji známe v živé přírodě. Jedinci v populaci (počítačové programy) jsou ohodnoceny tzv. *fitness hodnotou*, která odpovídá tomu, jak dobře daný program řeší problém. Programy s vysokou fitness mají větší šanci být vybrány do reprodukčního procesu (křížení a mutace) a jejich potomci tvoří další generaci. Důležitým rysem GP a obecně každého EA je to, že veškeré operace v procesu evoluce jsou náhodné a nejsou řízeny žádnými vnějšími zásahy s ohledem na požadovaný výsledek.

Pomocí GP již bylo řešeno mnoho problémů [1, 2] a v některých případech bylo dosaženo lepších výsledků než při řešení konvenčními metodami nebo byla objevena zcela nová řešení. Pro aplikaci GP je třeba použít nějaký softwarový nástroj. Takových nástrojů existuje již celá řada. Mohou být určeny k řešení konkrétního problému nebo být univerzální. Mezi nejznámější nástroje patří *lil-gp*, *GPC++*, *ECJ*, *OpenBEAGLE*. Pro implementaci bývají nejčastěji použity jazyky *C/C++* nebo *JAVA*.

Cílem tohoto projektu je vyzkoušet knihovnu *GPC++* k aplikaci GP. Jako demonstrační příklad byl vybrán problém hledání mediánu posloupnosti čísel. Cílem projektu není nalezení optimálního řešení, ale pouze otestovat daný nástroj.

2 ANALÝZA PROBLÉMU

Úkolem projektu je nalezení programu pro výpočet mediánu posloupnosti čísel. Tento program bude na vstupu přijímat zadaná čísla a bude vracet hodnotu jejich mediánu.

Výsledný program můžeme reprezentovat stromem, kde listy stromu jsou vstupní hodnoty, v kořenu je hledaná hodnota a vnitřní uzly jsou vhodně zvolené funkce. Pro náš problém byly zvoleny funkce, které vrací *menší* resp. *větší* hodnotu ze dvou vstupních hodnot. Jsou to funkce $min()$ resp. $max()$, jejichž význam odpovídá jejich názvu. Formálně lze problém zapsat takto:

Množina terminálů je

$$T = (x_1, x_2, \dots, x_n), \quad (1)$$

kde n je počet vstupních hodnot a množina funkcí je

$$F = (min, max). \quad (2)$$

Množina funkcí splňuje požadavek jednoduchosti a úplnosti. Jednoduchostí je zde míněna časová složitost, se kterou se daná operace provede. Úplnost znamená, že problém lze vyřešit pouze pomocí uvedených funkcí.

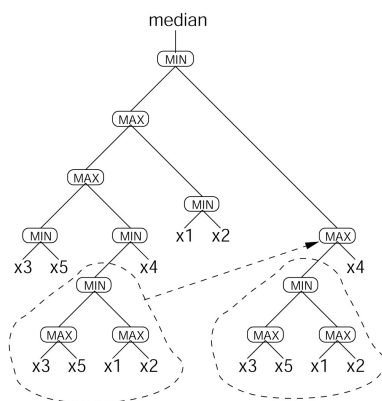
K řešení problému byl zvolen systém *Genetic Programming Kernel* (GPC++) [5]. Jedná se o knihovnu jazyka C++. Její použití je poměrně jednoduché a není potřeba hlubší znalost její vnitřní implementace. Programátor využívá dědičnosti a implementuje potřebné metody. Stejně jako je tomu u většiny systému pro GP, i zde je nutné implementovat zejména funkci pro výpočet *fitness hodnoty* a těla funkcí, ze kterých je vytvářen výsledný program. V našem případě funkce $min()$ a $max()$.

3 EXPERIMENTY

Bylo provedeno několik experimentů. Řešení se hledalo pro 3, 5, 7 a 9 vstupních hodnot. Pro každý počet se provedlo několik běhů, pokaždé s jinými parametry. V tabulce 1 jsou uvedeny nejlepší dosažené výsledky pro jednotlivé počty vstupních hodnot. Příklad jednoho z řešení je uveden na obrázku 1. Pro 11 vstupních hodnot se již použitou metodou řešení nepodařilo najít.

4 ZÁVĚR

Tato práce ukazuje možnosti genetického programování v řešení optimalizačních problémů. Výsledky, kterých bylo dosaženo, zaostávají za výsledky dosahovanými konvenčními algoritmy, avšak cílem práce nebylo nalézt optimálního řešení, ale spíše ukázat, že genetické programování je jednou z možností pro řešení podobných problémů. Jedním ze způsobů, jak dosáhnout lepších výsledků je dodatečná optimalizace nalezeného programu, jak je naznačeno na obrázku 1. Z obrázku a tabulky je vidět, že po provedení naznačené optimalizace je nalezeno optimální řešení pro počet hodnot $N = 5$.



Obrázek 1: Nejlepší řešení programu pro medián z 5. Čárkovaně je naznačena možná optimalizace. Po optimalizaci je program vytvořen pouze z 10 funkcí.

Medián z n	$P(n)$	$P_{best}(n)$	$d(n)$
3	4	4	3
5	13	10	7
7	38	20	8
9	121	30	15

Tabulka 1: Nejlepší dosažené výsledky při experimentech. Kde P je počet funkcí $min()$ a $max()$, ve sloupci P_{best} jsou nejlepší známé výsledky a d je hloubka stromu.

REFERENCE

- [1] Koza, J. R.: Genetic Programming: On the Programming of Computers by Means of Natural Selection. Morgan Kaufmann Publishers, Cambridge, MA 1992.
- [2] Koza, J. R., Bennett III, F. H., Andre, D., Keane, M. A.: Genetic Programming III: Darwinian, Invention and Problem Solving. Morgan Kaufmann Publishers, San Francisco, CA 1999.
- [3] Banzhaf, W., Nordin, P., Keller, R. E., Francone, F. D.: Genetic Programming: An Introduction, Morgan Kaufman Publishers, San Francisco, 1998.
- [4] Kvasnička, V., Pospíchal, J., Tiňo, P.: Evolučné algoritmy, STU Bratislava, 2000.
- [5] Fraser, A., Weinbrenner, T.: Genetic Programming Kernel C++ Class Library. URL <http://www.cs.ucl.ac.uk/staff/W.Langdon/ftp/weinbenner/gp.html> (březen 2005).