Generalized Extremal Optimization: An Attractive Alternative for Test Data Generation

Bruno T. de Abreu*, Eliane Martins
Institute of Computing – UNICAMP
Av. Albert Einstein, 1251 – 13084-971
Campinas, SP, Brazil
{bruno.abreu,eliane}@ic.unicamp.br

Fabiano L. de Sousa
National Institute for Space Research
Av. dos Astronautas, 1758 – 12227-010
São José dos Campos, SP, Brazil
fabiano@dem.inpe.br

Categories and Subject Descriptors

D.2.5 [**Software Engineering**]: Testing and Debugging—*test data generators*

General Terms

Reliability, Verification

Keywords

Software testing, Automatic test data generation, Evolutionary algorithms, Evolutionary testing, Path testing

1. INTRODUCTION

Software testing is an important activity of the software development process, and its automation (specially of test data generation) has been a burgeoning interest of many researchers [1]. It has recently been shown that evolutionary algorithms, such as the Genetic Algorithms (GAs), are valuable tools for test data generation. This work assesses the performance of a recently proposed evolutionary algorithm, the Generalized Extremal Optimization (GEO) [2], on test data generation for programs that have paths with loops. GEO's main advantage in comparison to other stochastic algorithms is that it has only one free parameter to adjust, which eases the process to set it to give its best performance in a given application.

Test data generation consists of generating inputs for the SUT in order to evaluate its internal (white-box testing) or external (black-box testing) behavior. Typically, path testing consists of two steps: (i) select a finite set of paths to be exercised; (ii) generate test data to execute the set of paths. For the first step a criterion is necessary, since testing all execution paths in a program is generally impossible due to the existence of infeasible paths and loops. Here our concern is on step (ii): given a set of paths, how do you generate test data to exercise them?

2. RESULTS AND CONCLUSIONS

Seven well known benchmark programs (BPs) were used as study cases: simplified triangle (BP1), remainder (BP2), product (BP3), linear search (BP4), binary search (BP5), middle value (BP6) and triangle (BP7). The performance of GEO and one of its variations, GEO_{var}, was compared to the one of the Simple Genetic Algorithm (SGA) using three criteria: the average path coverage, the number of program executions (NPE), and the time spent until the end of the test data generation process. The statistical test one-way ANOVA was used to analyze the results of coverage and number of BP executions. Table 1 shows the results.

Table 1: Average path coverage, NPE and time spent during test data generation.

BP	SGA	GEO	GEO_{var}
1	**92.41%** - 253640	89.19% - 268171	91.9% - **226703**
2	100% - **109**	100% - 651	100% - 419
3	100% - 1853	100% - 2277	100% - **765**
4	97.76% - **45343**	**98.49%** - 51721	98.48% - 52229
5	**99.83% - 48286**	99.53% - 51381	99.52% - 50748
6	100% - **100**	100% - 550	100% - 267
7	66.62% - 799764	**66.67% - 799300**	65.58% - 800000

BP	SGA	GEO	GEO_{var}
1	21.59h	16.19h	**16.05h**
2	**0.01h**	0.18h	0.22h
3	9.70h	2.58h	**0.82h**
4	14.10h	**12.51h**	13.99h
5	13.30h	13.53h	**10.16h**
6	**0h**	0.03h	0.01h
7	67.34h	**44.67h**	48.71h

The statistical tests did not show significant differences between SGA and GEO's results in all BPs for coverage percent and number of BP executions (the highest difference was of 3.22% on SP 1 between SGA and GEO). Results also showed that GEO spent significant less time than the SGA to generate test data for the most complex BPs (1 and 7), besides requiring much less computational effort on internal parameter setting than the SGA.

These results indicate that GEO is an attractive option to be used for test data generation. One of the reasons is that for real world applications it is expected that the problems are more complex than the benchmark programs used in this work.

3. REFERENCES

[1] P. McMinn. Search-based software test data generation: a survey. *Software Testing, Verification & Reliability*, 14(2):105–156, 2004.

[2] F. L. Sousa, F. M. Ramos, P. Paglione, and R. M. Girardi. New Stochastic Algorithm for Design Optimization. *AIAA Journal*, 41(9):1808–1818, September 2003.

*Supported by the Brazilian Federal Revenue Service.

GECCO'07, July 7–11, 2007, London, England, United Kingdom.
ACM 978-1-59593-697-4/07/0007.