Investigation on Artificial Ant using Analytic Programming

Zuzana Oplatková
Tomas Bata University in Zlin
Faculty of Applied Informatics
Nad Stranemi 4511, 763 02 Zlín
Czech Republic
+420 57 603 5190
oplatkova@fai.utb.cz

Ivan Zelinka
Tomas Bata University in Zlin
Faculty of Applied Informatics
Nad Stranemi 4511, 763 02 Zlín
Czech Republic
+420 57 603 5190
zelinka@fai.utb.cz

ABSTRACT

The paper deals with a alternative tool for symbolic regression – Analytic Programming which is able to solve various problems from the symbolic regression domain. In this contribution main principles of Analytic Programming are described and explained. Then follows how Analytic Programming was used for setting an optimal trajectory for an artificial ant according to Koza. An ability to create so called programs, as well as Genetic Programming or Grammatical Evolution do, is shown in that part. Analytic Programming is a superstructure of evolutionary algorithms which are necessary to run Analytic Programming. In this contribution SelfOrganizing Migrating Algorithm and Differential Evolution as two evolutionary algorithms were used to carry simulations out.

Categories and Subject Descriptors

Genetic Programming: Application, Optimization, Genetic Programming, Analytic Programming

General Terms

Algorithms, Measurement, Design, Experimentation, Verification.

Keywords

Analytic Programming, Self-Organizing Migrating Algorithm, Differential Evolution, symbolic regression, Santa Fe trail

1. INTRODUCTION

The term "symbolic regression" represents a process during which measured data is fitted and a suitable mathematical formula is obtained in an analytical way. This process is well known for mathematicians. They used this process when they need a mathematical model of unknown data. For long time symbolic regression was a domain of humans but in few last decades computers have gone to foreground of interest in this field. Firstly, the idea of symbolic regression done by means of computer was proposed in Genetic Programming (GP) by John Koza [1-2]. The other two approaches are Grammatical Evolution (GE) developed by Conor Ryan [3-4] and here described Analytic Programming (AP) designed in [5 - 8].

GP was the first tool for symbolic regression done by means of computer instead of humans. The main idea comes from genetic algorithms (GA) which John Koza uses in his GP. The ability to solve very difficult problems was proved many times.

The other tool is GE which was developed in last decade of 20th century by Conor Ryan. GE has one advantage compared to GP and this is ability to use arbitrary programming language not only LISP as is in the case of GP. In contrast to other evolutionary algorithms, GE was used only with a few search strategies, with a binary representation of the populations [3].

This contribution demonstrates use of methods which is independent on computer platform (as author of AP suggests), programming language and can use any evolutionary algorithm (as demonstrated by [5 - 8]) to find an optimal solution of required task.

2. ANALYTIC PROGRAMMING

2.1 Description

Basic principles of the AP were developed in the 2001 and stands for synthesis of analytical solution by means of any evolutionary algorithms independently on individual representation [5 - 8].

AP is based as well as GP on the set of functions, operators and so-called terminals, which are usually constants or independent variables like for example: functions(Sin, Tan,, And, Or), operators (+, -, *, /, dt), terminals (2.73, 3.14, t)

All these "mathematical" objects create a set which AP tries to synthesize the appropriate solution from. Main core is based on Discrete set handling (DSH) which shows itself as universal interface between EA and symbolically solved problem. That is why AP can be used almost by any evolutionary algorithm [5 - 8].

Briefly said, in AP individuals consist of non-numerical expressions (operators, functions,…) as described above, which are in evolutionary process represented by their integer indexes. This index then serves like a pointer into this set of expressions and AP uses it to synthesize resulting function-program for cost function evaluation. List of expression can be mathematical or also linguistic terms as in this case which represents mobility of a robot. To find a final formula, use of evolutionary algorithm as an optimization tool which finds the best solution according to value of cost function is necessary as was mentioned in the introduction.

GECCO'06, July 8–12, 2006, Seattle, Washington, USA.
ACM 1-59593-186-4/06/0007.

3. PROBLEM DESIGN

3.1 Santa Fe description

The Santa Fe trail, demonstrated, was chosen from [1, 9] to make an enlarged and comparative study with the same problem which was solved by Koza in Genetic Programming.

The SantaFe trail is defined as a 32 x 31 fields where food is set out. The aim of the task is that an artificial ant should go through defined trail and eat all food what is there.

3.2 Set of functions

Functions used for movements of the ant are Left, Right, Move for turning left and right and moving one field forward. Two arguments function IfFoodAhead was for indicating food infront of the ant. The last 2 function Prog2 and Prog3 were for doing two action in together. There were necessary to create whole complex formula.

3.3 Used evolutionary algorithms

Based on reached results in [8] where authors of AP used four evolutionary algorithms, the same set of algorithms was used in this experiment, i.e. algorithms used in this study were SOMA and DE. (Simulated Annealing (SA) and Genetic Algorithms (GA) are in process now).

4. CONLUSIONS

This contribution deals with a tool for symbolic regression. This study shows that this tool is suitable not only for mathematical regression but also for setting of optimal trajectory for artificial ant which can be replaced by robots in real world, in industry.

To compare with standard GP it can be stated on the basic results above that AP can solve this kind of problems in shorter times as cost function evaluations are counted. The number is lowered by decreasing of size of population which is 300 in both algorithms. GP uses 4000 – 16000 mentioned in [1]. The time could be decreased by parallelization of the process, which is one of further plans, as Koza did in GP. He uses in GP activity computer-cluster consisting of hundreds PCs [10]. But in our case we used only **1** computer.

The aim of this study was not to show that AP is better or worse than GP (or GE when compared) but that AP is also a powerful tool for symbolic regression with support of different evolutionary algorithms.

The main object of the paper was to show that symbolic regression done by AP is able to solve also cases where linguistic terms as for example commands for movement of artificial ant or robots in real world are. For this purpose and because of time consuming calculations only 28 simulations for SOMA and DE were carried out. During these tests the best solution was found by DE which needs only 387 steps to complete the task.

Reached results:

- all 28 simulations were with positive result, i. e. all 28 simulations found solution which accomplished the required tasks thus Analytic Programming is able to solve such kind of problems in symbolic regression
- mutual comparison – SOMA and DE gave satisfactory results. On the other hand, all parameters were set up heuristically. Maybe some better setting could be to improve or speed up the convergence to right solution

Future research is key activity in this field. The first step is to finished simulation with other evolutionary algorithms (SA, GA) which are in process now. The second step is to try some other class of problems to show that Analytic Programming is powerful tool as Genetic Programming or Grammatical Evolution are.

5. ACKNOWLEDGMENTS

This work was supported by the grant NO. MSM 7088352101 of the Ministry of Education of the Czech Republic and by grants of Grant Agency of Czech Republic GACR 102/06/1132 and GACR 102/05/0271.

6. REFERENCES

[1] Koza J.R.: Genetic Programming, MIT Press, ISBN 0-262-11189-6, 1998.

[2] Koza J.R., Bennet F. H., Andre D., Keane M.: Genetic Programming III, Morgan Kaufnamm pub., ISBN 1-55860-543-6, 1999

[3] O'Neill M. and Ryan C. Grammatical Evolution. Evolutionary Automatic Programming in an Arbitrary Language. Kluwer Academic Publishers, ISBN 1402074441.

[4] O'Sullivan John, Conor Ryan, 2002, An Investigation into the Use of Different Search Strategies with Grammatical Evolution Proceedings of the 5th European Conference on Genetic Programming, p.268 - 277, 2002, Springer-Verlag London, UK, ISBN:3-540-43378-3

[5] Zelinka I.: Analytic programming by Means of Soma Algorithm. Mendel '02, In: Proc. 8th International Conference on Soft Computing Mendel'02, Brno, Czech Republic, 2002, 93-101., ISBN 80-214-2135-5.

[6] Zelinka I., Oplatkova Z.: Analytic programming – Comparative Study. CIRAS'03, The second International Conference on Computational Intelligence, Robotics, and Autonomous Systems, Singapore, 2003, ISSN 0219-6131

[7] www.ft.utb.cz/people/zelinka/ap

[8] Zelinka I.,Oplatkova Z, Nolle L..:Boolean Symmetry Function Synthesis by Means of Arbitrary Evolutionary Algorithms-Comparative Study, International Journal of Simulation Systems, Science and Technology, Volume 6, Number 9, August 2005, pages 44 - 56, ISSN: 1473-8031, online http://ducati.doc.ntu.ac.uk/uksim/journal/Vol-6/No.9/cover.htm, ISSN: 1473-804x

[9] Oplatkova Z., Optimal Trajectory of Robots Using Symbolic Regression, In: CD-ROM of Proc. 56th International Astronautical Congress 2005, Fukuoka, Japan, 2005, paper nr. IAC-05-C1.4.07

[10] J. R. Koza, M. A. Keane, M. J. Streeter, Evolving Inventions, ScientificAmerican, February 2003, p. 40-47, ISSN 0036-8733, (online www.sciam.com)