Dynamic Ant: Introducing a New Benchmark for Genetic Programming in Dynamic Environments

David Fagan
Natural Computing Research & Applications Group
University College Dublin
david.fagan@ucd.ie

Miguel Nicolau
Natural Computing Research & Applications Group
University College Dublin
miguel.nicolau@ucd.ie

Erik Hemberg
Natural Computing Research & Applications Group
University College Dublin
erik.hemberg@ucd.ie

Michael O'Neill
Natural Computing Research & Applications Group
University College Dublin
m.oneill@ucd.ie

Anthony Brabazon
Natural Computing Research & Applications Group
University College Dublin
anthony.brabazon@ucd.ie

ABSTRACT

In this paper we present a new variant of the Ant Problem in the Dynamic Problem Domain. This approach presents a functional dynamism to the problem landscape, where by the behaviour of the ant is driven by its ability to explore the search space being constrained. This restriction is designed in such a way as to ensure that no generalised solution to the problem is possible, thus providing a functional change in behaviour.

Categories and Subject Descriptors

I.2 [**Artificial Intelligence**]; I.2.2 [**Automatic Programming**]; F.4.2 [**Grammars and Other Rewriting Systems**]

General Terms

Algorithms, Experimentation

Keywords

Genetic Programming, Artificial Ant, Dynamic Optimisation

1. INTRODUCTION

Explicit studies of the behaviour of Genetic Programming(GP) in dynamic environments are minimal to date [2], and this has been recognised as an open issue for GP [11]. In this study we present a novel benchmark Dynamic Ant, and examine the behaviour of a grammar based form of GP on this benchmark.

The Ant Trail Problem has been viewed by many in Evolutionary Computation(EC) as a complex problem [7]. The most famous and widely used example of such a trail problem is the classic Santa Fe Ant Trail Problem. Harder variants also exist such as Los Altos Trail and San Mateo Trail which build upon the ideas behind the Santa Fe Trail and extend them by adding the need to learn more complex behaviours. These trails are used as common benchmark problems within the GP Field [5], as they provide problems which have been widely implemented and results are available for many different GP Systems. Grammatical Evolution (GE) [10, 2] is a grammar based from of GP [8], that has established itself as one of the most popular and widely used forms of grammar based GP over recent years [9].

It has been seen that solutions evolved with GE in a static environment have performed well when applied to real world dynamic situations. This is of interest as the problems these solutions came from are based in static environments. Current research interest has led to an investigation into applying GE to Dynamic Problem domains, but first a definition of Dynamic must be reached. To aid with this definition and before tackling complex real world problems a strategy of establishing a suite of benchmark problems is desirable. Dempsey et al. [2] makes reference to a spectrum of Dynamism in a very in-depth review of the current state of the dynamic problem domain and examines similar ideas by Branke [1] and DeJong [4] to name a few. The spectrum described is one from a problem where the change is predictable to a problem which is completely random. Due to this spectrum we began to investigate the possibility of taking a standard GP benchmark and using it as inspiration for a dynamic benchmark that could be placed on the spectrum. In [6] Langdon et al. define a Dynamic Ant problem which takes inspiration from the Santa Fe style trail and constructs what they deem to be a dynamic trail, by taking core modules of the Santa Fe style trail and constructing random trails. There is one drawback to this approach, it was observed that a general solution to all trails could be found. Is this dynamic or can this be viewed more as a generalisation problem? In the opinion of the authors a problem can be said to be dynamic if a functional change exists over time, and this change should not allow for a generalised optimal solution for all possible functions, as the goal of a dynamic problem is to adapt and evolve towards the ideal solution at a given time. With this in mind we propose a new Dynamic Ant, which experiences a functional change over time.

GECCO'11, July 12–16, 2011, Dublin, Ireland.
ACM 978-1-4503-0690-4/11/07.

2. DYNAMIC ANT PROBLEM

Dynamic Ant (DA) is a problem which came about after much discussion into what constituted a Dynamic Problem. The reasons for basing our first dynamic problem in the ant domain were twofold, firstly it was based on a classic benchmark problem in the Santa Fe Ant Trail [5], and secondly the ant trail represents a challenging type of problem to solve [7]. Originally the Ant was based upon work by [6] in which a form of DA was proposed where the building block or common modules that made up the Santa Fe trail where extracted and then randomly put together to form a trail. This design did not fit in with the idea that a dynamic problem should involve a functional change and that this change should lead to distinct behaviours. Previous approaches generated a solution which could successfully complete any generated trail as well as the original Santa Fe trail [6].

With this in mind a trail was designed which would remain static and the energy given to the ant or the number of moves possible would be set so that a change in energy would drive a functional change in the ants behaviour. While the trail design appears simple it is so by design, to enable detailed investigation into the behaviour of the ant as we increase the number and difficulty of obstacles that need to be negotiated to maximise food consumption. In order for our DA to fall in line with the idea of a functional change, specific energy levels had to be selected in order for the ants behaviour to be manipulated. The classic behaviour of an ant is to maximise the amount of food it eats. By adjusting the energy available to the ant, it was possible to make it attractive for the ant to skip food, as it would be less efficient to eat food down a certain path. The start position for the ant would be facing east while positioned a the start of the trail in the top left corner. The DA uses a similar grammar to those used in the majority of Santa Fe type trails. The energy levels chosen where picked as a result of testing and manual calculations.

3. EXPERIMENTAL DESIGN

We wish to test the hypothesis that cycling through different functional requirements of the ant problem will not be detrimental to performance of GE in the Dynamic Ant problem and that GE can move within the solution space to optimal solutions for the current period of activity in the problem. We also wish the verify that the optimal solutions for each energy level are distinct and that no universal solution exists for all possible states of the problem.

4. CONCLUSIONS & FUTURE WORK

Upon examination of the results certain observations are apparent. The performance of GE on the problem continues to improve over the 200 generations regardless of the frequency of change within the problem. The average best fitness and average mean fitness over the runs show that the performance levels of the system maintain knowledge as when we cycle the systems worst performance improves pointing towards the population continuing to learn and retain knowledge.

Further examination of the results show that as the system cycles the performance of GE with respect to the time taken to adjust to the new environment decrease with every cycle. The shock of changing environments is reduced with each cycle and the slope of the performance graphs increases and shows a much faster adaptation to the new environment in getting back down to previously seen performance levels and surpassing these levels.

This work has helped to define what we mean by a dynamic problem and has proven a good start to research in the dynamic domain. The first goal of this dynamic research is to establish a taxonomy of dynamism and fit problems into this so that we can establish a suite of benchmark problems. The Dynamic Ant is the first of these problems and it is hoped to use other known static benchmark problems and investigate if a dynamic implementation would meet our functional change criteria and where such problems place within the taxonomy above would be. A full version of this paper is also available [3].

5. ACKNOWLEDGMENTS

This research is based upon works supported by the Science Foundation Ireland under Grant No. 08/IN.1/I1868.

6. REFERENCES

[1] J. Branke. *Evolutionary Optimization in Dynamic Environments*. Kluwer Academic Publishers, 2001.

[2] I. Dempsey, M. O'Neill, and A. Brabazon. *Foundations in Grammatical Evolution for Dynamic Environments*. Studies in Computational Intelligence. Springer, 2009.

[3] D. Fagan, M. Nicolau, E. Hemberg, M. O'Neill, and A. Brabazon. Dynamic ant: Introducing a new benchmark for genetic programming in dynamic environments. Technical Report UCD-CSI-2011-04, School of Computer Science and Informatics, University College Dublin, `http://www.csi.ucd.ie/biblio`, 2011.

[4] K. A. D. Jong. Evolving in a changing world. In *ISMIS*, LNCS 1609, pages 512–519. Springer, 1999.

[5] J. R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, Cambridge, MA, USA, 1992.

[6] W. B. Langdon and R. Poli. Genetic programming bloat with dynamic fitness. In *Genetic Programming*, LNCS 1391. Springer Berlin / Heidelberg, 1998.

[7] W. B. Langdon and R. Poli. Why ants are hard. In *Genetic Programming 1998*, pages 193–201. Morgan Kaufmann, 1998.

[8] R. I. Mckay, N. X. Hoai, P. A. Whigham, Y. Shan, and M. O'Neill. Grammar-based genetic programming: a survey. *Genetic Programming and Evolvable Machines*, 11:365–396, September 2010.

[9] J. F. Miller and R. Poli. Editorial to tenth anniversary issue on progress in genetic programming and evolvable machines. *Genetic Programming and Evolvable Machines*, 11(3-4):247–250, 2010.

[10] M. O'Neill and C. Ryan. *Grammatical Evolution: Evolutionary Automatic Programming in a Arbitrary Language*. Genetic programming. Kluwer Academic Publishers, 2003.

[11] M. O'Neill, L. Vanneschi, S. Gustafson, and W. Banzhaf. Open issues in genetic programming. *Genetic Programming and Evolvable Machines*, 11:339–363, September 2010.