Genotype-Phenotype Mapping in Dynamic Environments with Grammatical Evolution

David Fagan

Natural Computing Research & Applications Group
Complex & Adaptive Systems Lab
School of Computer Science & Informatics
University College Dublin
Ireland

February 2, 2012

Abstract

The application of a genotype-phenotype mapping in Evolutionary Computation is not a new idea, however, how this mapping process is interpreted, and implemented varies wildly. In the majority of cases a very simple abstraction of the biological genotype-phenotype mapping is used, but as our understanding of this process increases, the deficiencies in current approaches become more evident. In this paper, an outline of what approaches have been taken in the investigation of the genotype-phenotype map in Grammatical Evolution are presented and an outline of proposed future work is introduced.

1 Introduction

The adoption of a Genotype-Phenotype Map (GPM) for Genetic Programming (GP) [11, 18] has demonstrated performance advantages over traditional tree-based GP [4, 13, 10, 2]. One of the most popular grammar-based forms of GP [12], Grammatical Evolution (GE), adopts a genotype-phenotype map which has been argued to provide a number of advantages over standard GP [16]. The GPM provides GE with the ability to search both genotypic space and solution space in a many to one relationship, unlike traditional GP which has a one to one mapping. The many-to-one mapping allows for multiple solutions to have the same performance but be structured differently. This feature allows for neutral search which allows the Evolutionary Algorithm(EA) to search with zero impact on performance amongst the different variants of the same solution and has been shown to allow GPM-based variants of GP to resist getting stuck at local optimal solutions [2]. Whilst these results are encouraging it has yet to be established what effect the GPM has on an EA such as GE. How does changing the order in which the mapping is done effect the EA? Can further inspiration be taken from biology which first inspired the GPM in GP to improve the EA. Recently these topics have started to be tackled [8, 15, 6, 7, 1, 14], but many more avenues of exploration remain as the interpretation of mapping used by GE is simplistic and lacking in some of the desired advanced features of the GPM that exist in nature [3].

GP research in dynamic environments was recently noted as an open issue for GP [17]. Within a dynamic environment the ability to adapt to change is crucial. Fast adaptation of a GP system requires diversity to be present within the population. The degeneracy offered by a GPM provides the system with diversity within the population thus making it the ideal choice when applying GP to dynamic environments. However the current usage of GPM has been noted to lack the complexity shown in nature [3] as it is based upon an old model of what was once perceived to take place within Genotype-Phenotype Mappings in the natural world.

The remainder of this paper is as follows. An overview of the main research objective in Sections 2 and 3, followed by a summary of work done in Sections 4 and 5, and finally an outline of the future direction of the research in Section 6.

2 Research Objective

Whilst the idea of the GPM has been around for decades within the field of GP a comprehensive in depth examination of the topic remains absent from the field. Whilst most studies focus on the application of a new or refined mapping process, the outcome of these studies in general, is purely performance based. Most approaches are benchmarked and compared to a standard simplified approach to Genotype-Phenotype Mapping or even standard GP variants such as grammar based GP. These studies lack a broad spectrum of comparison to other approaches. The **no free lunch** theory states no single search algorithm can be the best at all problems and with this in mind it makes sense to find as many successful variations of the Genotype-Phenotype Mapping process to try and ensure a greater spectrum of possible good approaches to mapping. It has also been noted that we need to complexify our approaches to GP to narrow the gap between GP and actual biological systems [17]. Through this proposed investigation, more complex mapping approaches the need for which is further highlighted in [3], will be considered and investigated to try and help narrow the gap between accepted abstractions of GPM and what actually occurs in nature.

Recently it has been noted that the lack of investigation and application of GP to dynamic domains is an open issue within the field of GP [17]. Further focusing on this issue, the investigation of Genotype-Phenotype Mapping behaviour with dynamic environments it is found lacking and in need of in-depth investigation. As understanding of the natural mapping process becomes more widespread, it is seen that mapping is a complex two-way system that adapts at each stage of mapping [3], unlike traditional one way approaches which are the norm in EC. It has already been established that the GPM provides diversity to a system and this diversity is key for fast adaptation to new environments.

It is from these two arguments above that I see the need to explore the open issues I perceive within GPM in the field of GP. An in-depth review of current mapping ideas is required and a comparison or benchmarking of each approach is needed. The surface has only been scratched with regards to inspiration from biology and further enhancement of the Genotype-Phenotype mapping with this knowledge is worth investigation. Finally all this work needs to be applied to Dynamic Environments and examined for performance and behaviour of the algorithm, as Dynamic Environments are the real world problems we face every day and desire solutions for from GP. It is my hope that a more complex mapping process will result in the ability for a Genetic Algorithm to adjust to Dynamic Environments appropriately and become more resistant to fall-off in performance in highly volatile environments.

3 Research Questions

Do different GPM's impact on GE's performance? With any form of Genetic Programming where a mapping from chromosome to possible solution is performed the way in which this mapping is performed can have a large impact on performance. With Grammatical Evolution a GPM is used in which mapping is performed on the chromosome to a grammar and the phenotype or possible solution is mapped in a left-most first manner. This introduces a bias in the mapping process which can be beneficial with certain problems as noted in [9].

My approach will examine different ways of implementing this mapping from chromosome to solution and investigate the possibility of the existence of mappings that are more suited to certain types of problems, or if a general mapping can be found which exhibits acceptable performance across all problem domains.

Do advanced genetic operations exist that when applied to the new GPMs impact GE's performance? It is not enough to simply investigate new mapping processes and report on findings. Genetic Operations used in many GP systems are reliant upon the mapping process. Applying standard mutation within a system using a different mapping can be tantamount to making the algorithm perform no better than random search as the change is too destructive to allow for any form of evolution or knowledge to be maintained within a chromosome.

Different mappings require genetic operators which are firstly enhanced to deal with the complex and different types of mappings possible. This will allow for the evolutionary process to work, but in certain cases mappings present an opportunity to use advanced genetic operators not possible with standard mapping practices. These operators allow for a great degree of freedom for the system to fine tune the evolutionary search going on and in certain cases adjust to prevent convergence which is a big problem within GP. This idea of mapper specific genetic operations represents a key area of investigation within the area of GPM to enable us to get the best performance out of the various types of GPM.

My approach will examine all the mapping approaches used within this proposed thesis and examine if there exists the possibility of implementing operators to take advantage of degrees of freedom in the mapping to help refine the searching process. As the research moves to the dynamic problem domain the application of dynamically changing operators may prove to provide desired improvement to the performance of the algorithm.

Do observed impacts on performance hold when applied to Dynamic Environments? Currently the majority of work within the field of Genetic Programming is performed on static environments. The main goal of all researchers within the field would be to apply their work to real world problems and as we all know the real world can be very dynamic. However, research within static environments is very useful in that it can provides the ability to design experiments where the behaviour or solution is known and thus allows for focused research on certain aspects of a genetic algorithm, where changes can be observed and performance of the changes to the algorithm can be easily examined and benchmarked.

Once a system has been examined and fine tuned within a static environment the next logical step is to examine these changes within a, similar, dynamic environment. This allows for the examination of the observations made within a static testbed and to see if they translate into the new dynamic testbed environment, or if the addition of the changeable environment will favour different mapping strategies, and genetic operations.

4 GPM in GE

GPM in GE begins by finding the start symbol in the grammar. This non terminal (NT) in the case of the example grammar shown in Fig. 1, `<e>` is then evaluated using Eq. 1. By taking the first codon value of the GE chromosome (12) and the number of expansions possible for the state `<e>` (2), we get the first expansion of the tree, where `<e>` expands to `<e><o><e>` (12%2) . From this point on the leftmost NT is always expanded first in the derivation process. This action will continue to be performed until no NTs remain to be expanded in the derivation tree. An example of this mapping is shown in Fig. 1 where the order of expansion is indicated by a set of numbers on the arrows between the blocks on the diagram, in the form of 1(12%2) where 1 is the expansion order and 12%2 is the application of Eq. 1.

$$New\ Node\ =\ Codon\ value\ \%\ Num.\ of\ rules\ for\ NT \quad (1)$$

$$NT\ to\ expand\ =\ Codon\ value\ \%\ Num.\ of\ NT's \quad (2)$$

The only difference between standard GE and πGE [15] in its purest form is in the mapping process from genotype to phenotype. πGE's mapping process

```
<e> ::= <e><o><e>
      | <v>
<o> ::= + | *
<v> ::= 0.5 | 5

Chromosome ::=12,8,3,11,
              7,6,11,8,
              4,3,3,11,
              15,7,9,8,
              10,3,7,4
```

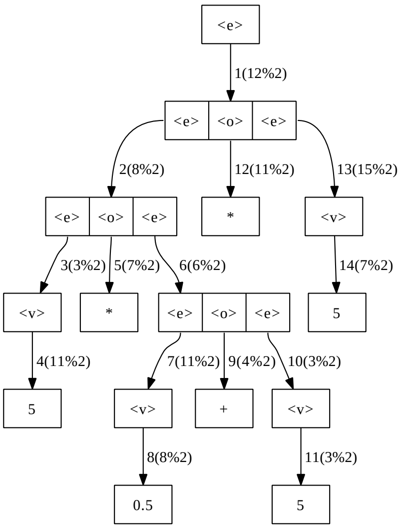

Figure 1: Example Grammar and GE GPM

```
1. (e)-12%1=0
2. (e),o,e-3%3=0
3. o,(e),v-7%3=1
4. o,(v),e,o,e-11%5=1
5. (o),e,o,e-4%4=0
6. (e),o,e-3%3=0
7. (o),e,v-15%3=0
8. e,(v)-9%2=1
9. (e)-10%1=0
10.(v)-7%1=0
```

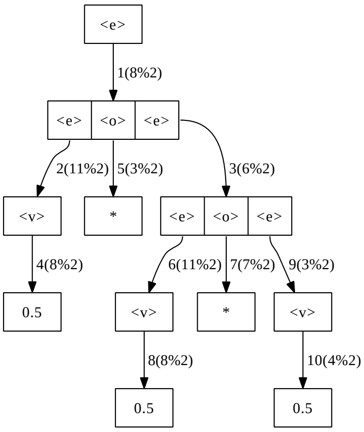

Figure 2: πGE 's Order selection and GPM

differs from that of GE in that each expansion of a NT requires two codons. The standard GE chromosome is essentially split into pairs of values where the first codon of the pair is used to choose which NT to expand and the second is used to choose what to expand the NT to, based on the rules available for a NT of that type. The chromosome shown in Fig. 1 can be viewed as a list of paired values such as `((12,8),(3,11)..)`, where the first value of the pair (The Order Codon) is used to determine the next NT to expand by using Eq. 2 and this will return which NT to choose from a list of unexpanded NTs. Once the NT to be expanded has been chosen, the second codon (Content Codon) is used in conjunction with Eq. 1 (the standard GE expansion rule) to determine what the NT expands to; and if this node happens to be an NT, it is added to the list of unexpanded NTs. Fig. 2 shows the expansion of the example grammar in Fig. 1 using the πGE mapping process. The number associated with each branch of the tree is a reference to the numbered steps shown in Fig. 2 which show how each choice of NT to expand comes about. It is interesting to note the different shape and size of the examples based on just a change in mapping.

The two methods for GPM in GE presented above represent the starting point, and current most advance GPM representation explored. Other variants of the GPM in GE have been looked at as well during this research. In [8], the above GPM and the following are examined. **Breadth-first**, which maps all of the non-terminal symbols at each successive level of the derivation tree, before moving on to the next level down and **Random** control strategy, which randomly selects a NT to expand amongst all of the NTs that currently exist in an expanding derivation sequences.

The initial study showed that of the four mapping strategies examined, πGE represented the best avenue to explore as it showed performance on a par with the standard GE GPM except on the Max problem. In [6] investigation into this observed decrease in performance of πGE on the Max problem led to the introduction of some new variants of both the GE and πGE mappings. Some new variations of mappers where required. The experiments planned required a version of πGE, in which the order codons of the mapping were fixed across the whole population, that is not affected by crossover or mutation. This fixed order πGE requires the addition of an order chromosome to standard GE and then an edit to the πGE mapper so it would work with the new desired setup. This setup is referred to as **Fixed-Order** and was necessary to see if the constantly changing order of expansion in the πGE mapping process, was the cause of the performance decrease. The mapping served as an experiment by itself, as it showed if any randomised order might do as well as πGE . The final GPM explored was **Right-Most**, this is a variant of the standard GE GPM. Rather than taking the left-most NT and expanding as in standard GE, in Right-Most, GE always selects the right-most NT for expansion. This way of mapping was required for this study as it provided a similar performance to the standard GE GPM, whilst also furthering insight into what happens within the evolution of a solution to the Max problem.

5 GPM Based Operators

Adopting the πGE GPM allowed for certain degrees of freedom within the mapping process that could be exploited. In [7] one such freedom, which exists in the mutation operation, was used to explore the effect mapping order has on the performance of GE. πGE allows for the mutation of the specific codons that control the order of expansion as well as the choice of expansion. In total four different variants of the mutation operation where tested and the results examined. **Order** Mutation, restricted to codons responsible for determining the mapping order. The results observed on this setup relative to the others allowed us to determine the contribution of the search focused on the order codons towards the success of πGE. **Content** Mutation, restricted to codons responsible for production rule selection. When compared to a standard GE mapping, in effect the mapping order is largely randomised here upon initialisation of the order codons in the first generation. **πGE** Mutation events are allowed on both order and content codons. **Ratio** Mutations were examined where the ratio of order to content mutation events are varied to examine the situation where the search is allowed to continue on both the order and content codons, but at different relative rates. This showed if there may be an advantage in rebalancing the relative rate of codon and order search. Crossover is also subject to some modification under a πGE mapping. The ability to break up the location and content pairs of the chromosome during crossover or preserve this relationship is now possible.

6 Future Work

6.1 Complex Mappings

The first and primary goal of the future work will be to further examine the existing literature for novel approaches to GPM and also try to take further inspiration from nature and examine if different methods to perform Genotype-Phenotype mapping can be derived and implemented. During the process of implementing new variants of the GPM, it must also remain in focus to examine the approach being used to perform GPM and investigate if there exists scope for complex operators. Some operators already established such as those seen in [7] can by dynamically changed to allow for variable performance rates. The use of new operators go hand in hand with new GPM implementations so in the view of this author, the development of these operations go hand in hand with the new GPM. A successful GPM strategy is only as good as the operators that interact with the GPM.

6.2 Dynamic Environments

The main idea behind this research is to apply GPMs to a wide range of Dynamic Problems. Dempsey [5] provided a very in-depth survey of how to classify dynamic problems. Dempsey examined work by many, and put forward his idea of a spectrum or range of dynamism, based on this work, and that all problems within the dynamic domain would have a place upon this spectrum. The spectrum goes from the two perceived extremes of change, one side being the very small predictable change problems and the other being the completely random, large degree of change that was unpredictable. Since the work outlined in previous sections called for a suite of dynamic benchmark problems this will form one of the important tasks that needs to be completed. The goal of this section of proposed research is to establish a suite of problems that will provide a good coverage of this spectrum. This task is currently underway and it is hoped to add to the initial dynamic problems implemented to date.

6.3 Apply research to Dynamic Environments

Upon establishment of a suite of Dynamic Benchmark Problems the next set of experiments proposed will be a repeat of those performed within the static benchmark problems. Once completed it will be possible to investigate if the observed behaviour of the mapping process on the static benchmarks translates into the dynamic domain. This will then lead to a series of experiments that will be needed to see why we may observe a positive or negative impact on performance. This should lead in nicely to the next research topic which is that of dynamic mappers and operators.

6.4 Dynamic GPM and Genetic Operators

With the introduction of Dynamic Environments to this research the idea of having dynamic GPM's and operators could prove to be a very desirable feature. The idea that certain GPM's could be better suited to certain types of situations that could exist in the environment such as, one type of GPM could be very good when the environment is highly changeable, while another might be very good at times of limited change, leads to the conclusion that some form of dynamic approach to mapping may lead to an as yet unseen performance advantage. In the same way, having operators that can adjust the degree of change within an algorithm can also be of benefit. Being able to inject diversity into a converged population after a prolonged period of time in a certain environment would certainly be desirable.

Acknowledgments

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

References

[1] R. M. A. Azad. *A Position Independent Representation for Evolutionary Automatic Programming Algorithms - The Chorus System.* PhD thesis, University of Limerick, Ireland, Dec. 2003.

[2] W. Banzhaf. Genotype-phenotype-mapping and neutral variation – A case study in genetic programming. In *Parallel Problem Solving from Nature III*, LNCS866. Springer, 1994.

[3] W. Banzhaf, G. Beslon, S. Christensen, J. A. Foster, F. Kepes, V. Lefort, J. F. Miller, M. Radman, and J. J. Ramsden. Guidelines: From artificial

evolution to computational evolution: a research agenda. *Nature Reviews Genetics*, 7(9):729–735, Aug. 2006.

[4] M. F. Brameier and W. Banzhaf. *Linear Genetic Programming.* Springer, 2007.

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

[6] D. Fagan, M. Nicolau, E. Hemberg, M. O'Neill, A. Brabazon, and S. McGarraghy. Investigation of the performance of different mapping orders for GE on the max problem. In *Proc. of EuroGP 2011*, LNCS 6621, Turin, Italy, 27-29 Apr. 2011. Springer.

[7] D. Fagan, M. Nicolau, M. O'Neill, E. Galvan-Lopez, A. Brabazon, and S. McGarraghy. Investigating mapping order in piGE. In *2010 IEEE World Congress on Computational Intelligence*, pages 3058–3064, Barcelona, Spain, 18-23 July 2010. IEEE Press.

[8] D. Fagan, M. O'Neill, E. Galván-López, A. Brabazon, and S. McGarraghy. An Analysis of Genotype-Phenotype Maps in Grammatical Evolution. *Genetic Programming*, pages 62–73, 2010.

[9] E. Hemberg. *An Exploration of Grammars in Grammatical Evolution.* PhD thesis, University College Dublin, Ireland, 17 Sept. 2010.

[10] R. E. Keller and W. Banzhaf. Genetic programming using genotype-phenotype mapping from linear genomes into linear phenotypes. In *Genetic Programming 1996.* MIT Press, 1996.

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

[12] J. F. Miller and R. Poli. Editorial. *Genetic Programming and Evolvable Machines*, 11(3/4):247–250, Sept. 2010.

[13] J. F. Miller and S. L. Smith. Redundancy and computational efficiency in cartesian genetic programming. *IEEE Transactions on Evolutionary Computation*, 10(2):167–174, Apr. 2006.

[14] M. Nicolau. *Genetic Algorithms using Grammatical Evolution.* PhD thesis, University Of Limerick, 2006.

[15] M. O'Neill, A. Brabazon, M. Nicolau, S. M. Garraghy, and P. Keenan. πgrammatical evolution. In *GECCO-2004, Part II*, LNCS3103. Springer, 2004.

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

[17] 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.

[18] R. Poli, W. B. Langdon, and N. F. McPhee. *A field guide to genetic programming.* Published via `http://lulu.com` and freely available at `http://www.gp-field-guide.org.uk`, 2008.