



Provided by the author(s) and University of Galway in accordance with publisher policies. Please cite the published version when available.

Title	Examining the use of a non-trivial fixed genotype-phenotype mapping in genetic algorithms to induce phenotypic variability over deceptive uncertain landscapes
Author(s)	Hill, Seamus; O'Riordan, Colm
Publication Date	2011-06-05
Publication Information	Hill, Seamus, & O' Riordan, Colm. (2011). Examining the use of a non-trivial fixed genotype-phenotype mapping in genetic algorithms to induce phenotypic variability over deceptive uncertain landscapes. Paper presented at the 2011 IEEE Congress of Evolutionary Computation (CEC), New Orleans, LA, USA, 5-8 June, DOI: 10.1109/CEC.2011.5949780
Publisher	IEEE
Link to publisher's version	https://dx.doi.org/10.1109/CEC.2011.5949780
Item record	http://hdl.handle.net/10379/14681
DOI	http://dx.doi.org/10.1109/CEC.2011.5949780

Downloaded 2024-04-26T15:26:06Z

Some rights reserved. For more information, please see the item record link above.



Examining the use of a Non-Trivial Fixed Genotype-Phenotype Mapping in Genetic Algorithms to Induce Phenotypic Variability over Deceptive Uncertain Landscapes

Seamus Hill

College of Engineering and Informatics
National University of Ireland Galway
Email: seamus.hill@nuigalway.ie

Colm O’Riordan

College of Engineering and Informatics
National University of Ireland Galway
Email: colm.oriordan@nuigalway.ie

Abstract—In nature, living organisms can be viewed as the product of their genotype-phenotype mapping (GP-map). This paper presents a GP-map loosely based on the biological phenomena of *transcription* and *translation*, to create a multi-layered GP-map which increases the level of phenotypic variability. The aim of the paper is to examine through the use of a fixed non-trivial GP-map, the impact of increased phenotypic variability, on search over a set of deceptive landscapes. The GP-map allows for a non-injective genotype-phenotype relationship, and the phenotypic variability of a number of phenotypes, introduced by the GP-map, are advanced from the genotypes used to encode them through a basic interpretation of transcription and translation. We attempt to analyse the level of variability by measuring diversity, both at a genotypic and phenotypic level. The multi-layered GP-map is incorporated into a Genetic Algorithm, the multi-layered mapping GA (MMGA), and runs over a number of GA-Hard landscapes. Initial empirical results appear to indicate that over deceptive landscapes, as the level of problem difficulty increases, so too does the benefit of using the proposed GP-map to probe the search space.

I. INTRODUCTION

Living organisms when viewed through the lens of evolution can be seen as the result of the mapping from their genotype to their phenotype. Evolution is the process of searching the phenotypic space through the use of genetic operators such as mutation, recombination and selection. While genetic operators operate at the genotypic level, fitness-based selection operators function at the phenotypic level. Because of this, the relationship between the genotype and the phenotype can be expressed as the *Genotype-Phenotype Map* (GP-map) [1]. The GP-map generates a phenotypic neighbourhood, which very often differs from the idea of proximity which exists between phenotypes when we consider them as a whole [2]. The relationship between the genes and the phenes (phenotypic traits which have an influence on a phenotypes overall fitness) is controlled by the GP-map [3]. Phenotypic plasticity [4] on the other hand, describes how the instructions included in the genotype for the creation of the phenotype are not complete in

that there is an external influence which needs to be taken into account. In other words, a number of phenotypes can result from a given genotype as a result of the environment in which the mapping takes place [5] and there exists an interaction between learning and evolution [6].

When exploring the phenotypic space, it is critical to gain an understanding of the variational topology in trying to determine the shape of the landscape [7]. Many evolutionary algorithms are composed using a fixed variation topology. However, in nature phenotypic variation landscapes are not fixed. These non-fixed phenotypic variation landscapes can be referred to as a non-trivial in terms of their GP-map [7]. A non-trivial GP-map can be viewed as having the following characteristics: firstly, a phenotype can be encoded by many genotypes and secondly, the phenotypic variability of a number of phenotypes will depend on their genotype [3]. Therefore, every genotype can be described as a combination of a phenotype and any kind of neutral traits which exist within the genotype, where different neutral traits associated with the same phenotype give a different representation of the phenotype in question [3]. As was pointed out by Toussaint in [7], the introduction of an adaptable exploration is possible by creating a fixed but non-trivial GP-map, which is achieved by having neutral variations in the genotype space. The choice of genetic representation, the primary structure, used to represent a neutral set which encodes a phenotype will impact on the phenotypic variability.

The motivation for attempting to create a multi-layered GA, comes from the desire to create a fixed non-trivial GP-map which introduces a more flexible phenotypic structure and a higher degree of phenotypic variability through the use of neutrality. The aim of this paper is to put forward a proposed GP-map which includes a number of the features of a fixed non-trivial GP-map which operate using a binary representation and to investigate whether the increased neutrality and level of phenotypic variability can prove beneficial in searching a

series of fully deceptive landscapes. The main contribution is to propose a model for exploring the effects of phenotypic variability using a Genetic Algorithm with a binary representation. The paper is laid out as follows Section II examine past research. Section III outlines the pseudo code and workings of the multi-layered mapping GA (MMGA), while Section IV describes the experiments conducted. Finally Section V discusses the experiments and outlines possible future work.

II. BACKGROUND

Evolutionary Computation can be viewed as an exercise in *hillwalking* over a fitness landscape, where peaks can be seen as phenotypes with high levels of fitness and troughs as phenotypes of low fitness values. Genetic operators such as mutation, function by introducing diversity into the population, whereas selection draws the search towards peaks [8]. By introducing a more complicated GP-map the effect of operators, such as mutation, can be magnified, in that the effect of a single mutation on a phenotype can be increased as many alterations at a genotypic level may occur and not be expressed at the phenotypic level [9]. However, landscapes may contain local peaks, which are sub-optimal points in the fitness landscapes and once a population reaches a local optimum it can prove very difficult to adapt further. Populations can occupy several peaks at a given time and by recombining individuals from different peaks, the search may continue [8].

A. Deception

In order to examine the performance of the multi-layered GA against a simple GA (SGA), a problem environment needs to be chosen to enable a comparison to be made. With regard to testing the performance of GAs a number of authors have used a class of problem considered *GA-Hard*. A problem is considered hard in many cases if it is difficult for the GA to find the optimum, in other words we are looking for single optimum point surrounded by points which have a relatively low fitness level. Problems are often created by taking advantage of the GA so as to allow selection to deliberately lead the search away from the optimum. By using a binary encoding of a solution, GAs can search complex landscapes by sampling hyperplanes in an n-dimensional hypercube. GAs alter the rates of sampling of hyperplanes in relation to encodings of the solution space. In relation to deceptive problems hyperplanes drive the search away from the the global optimum and towards a local optimum [10].

Previous research into the use of deceptive problems with GAs include Goldberg [11], Tanese [12], Grefenstette [13] and Whitley [14]. A Minimal Deceptive Problem was introduced by Goldberg [15], based upon previous work carried out by Bethke [16], in order to develop an enhanced understanding of problems which are likely to prove difficult for a GA to solve. The *schema* theory [17] helps explain how GAs search difficult search spaces by altering the rate of sampling of hyperplanes [17][18] and describes how low level building blocks relate to the hyperplanes which are recombined, moving the search

String	Fitness	String	Fitness
f(000)	28	f(010)	22
f(110)	0	f(101)	0
f(001)	26	f(100)	14
f(011)	0	f(111)	30

TABLE I
FULLY DECEPTIVE ORDER-3 PROBLEM

String	Fitness	String	Fitness
f(111)	28	f(101)	22
f(010)	0	f(100)	0
f(110)	26	f(011)	14
f(001)	0	f(000)	30

TABLE II
REVERSED FULLY DECEPTIVE ORDER-3 PROBLEM

towards a solution which is above the average fitness of the population.

A fully deceptive problem of order-N can be viewed as being deceptive when all of the lower-order hyperplanes lead away from the global optimum and towards a deceptive attractor [14]. To illustrate deception, let us consider a fully deceptive order-3 function, where the information about the hyperplane which the order-1 and order-2 schemata represent in the search space direct the search away from the global optimum towards a deceptive attractor [14]. If the bits 111 represent the global optimum and the bits 000 represent the *deceptive attractor*, then a full order-3 deception would be similar to that defined by Goldberg, Korb and Deb [11] which is illustrated in Table I and shows the fitness values for each bit string.

B. Variational Topology

We can view evolution as operating on the genotype space and when exploring the search space evolution, through recombination and mutation defines the search space's variational topology. When discussing variational topology we need to examine both genotypic variational topology and phenotypic variational topology. Nature uses a complex GP-map to advance a relatively simple genotype space variational topology to an extremely complex phenotypic variational topology. Toussaint [19] argues that the phenotype space is what should in fact be viewed as the search space for evolution rather than the genotype space. With this in mind, the GP-map is the key to understanding phenotypic variability and it also allows us to gain an insight into how evolution can adapt the search on the phenotype space.

The genotype is a collection of items which are inherited and define the possible distribution of offspring. The phenotype on the other hand, comprises all of the phenotypic traits of an individual that contribute to the overall fitness of that individual. The GP-map sets out a blue print for moving from genotype to phenotype. By viewing a phenotype as the combination of all of the traits an organism and the genotype as the genetic encoding of this phenotype, then we will also

assume that there exists a genotypic neighbourhood and a phenotypic neighbourhood. Two genotypes can be considered equivalent if they represent the same phenotype, however their variational topologies can differ [20]. The GP-map “*induces a variational topology on the phenotype space depending on the topology of the genotype space*” [19]. If we allow for a non-injective GP-map, then the same phenotype can be encoded by a number of genotypes. One major advantage of allowing a many-to-one non-trivial GP-map is that it enables the changing of the genotype representation without changing the phenotype.

This can be achieved through *neutral mutations* which allow for the changing in the phenotypic neighbourhood and assists in the search. In other words, neutrality enables phenotypic variability to adapt and vary, whereby the GP-map itself doesn’t need to evolve, neutrality in the phenotypic variational space allows changes to occur in the gene interaction which introduces a different phenotypic variability. Therefore, by introducing neutrality in a non-trivial mapping, the phenotypic neighbourhood depends on the genotypic representation of the phenotype and major alterations of the genotypic representation mean changes in the phenotypic neighbourhood [19]. “*The notion of fitness landscapes is sometimes also used to describe a fitness function over the phenotype space in the case of a non-trivial genotype-phenotype mapping*” [19], but by introducing a fixed (non-trivial) GP-map you can still achieve adaptable exploration through “*neutral variations in the genotype space*” [19].

Trivial neutrality can be defined as a situation where “*the evolution of phenotypes can be understood (i.e., modelled) without referring at all to genotypes, in particular, neutral traits are completely irrelevant for the evolution of phenotypes*” [19]. In a non-trivial mapping a single gene is associated with a number of features in the phenotype, when this gene mutates the features vary in accordance [19].

For a GP-map to be non-trivial there are a number of conditions which must be met. Firstly, there is a non-injective relationship between genotype and phenotype and secondly, for a number of the phenotypes, the phenotypic variability generated depends on the genotype from which it has emerged and the genotype contains neutral traits, which implies that you can have different genetic representations for the phenotype [3].

The *locality of representation* defines the degree of correlation which exists between the neighbouring genotypes and neighbouring phenotypes. If there is a high level of correlation between neighbouring genotypes and neighbouring phenotypes, then the locality of representation is high. However, when representations change the distance between corresponding genotypes and phenotypes, they also change the locality of reference [21]. The GP-map proposed by the authors decreases the locality of reference by introducing a form of neutrality as the search moves from the genes to phenes and also increases the level of phenotypic variability, measured in this paper by the level of diversity within the population.

The central dogma of molecular biology states that information which is located in DNA is copied, through a process known as transcription, to RNA. Following this a process known as translation takes place which uses RNA to assemble proteins. Transcription transforms a gene into an RNA molecule that is complementary to one strand of the DNA double helix. Following this, translation uses the information stored in three types of RNA to manufacture a protein by combining specified amino acids. RNA acts as a link between gene and protein and is complementary to one strand of the double helix, called the template strand, with the other strand being the coding strand. The Multi-layered Mapping Genetic Algorithm (MMGA) GP-map moves from genes to phenes by using a basic interpretation of the processes of transcription and translation. The MMGA operates using a binary representation which allows the use of standard genetic operators such as crossover and mutation. The MMGA differs from a simple GA in that it introduces a more flexible phenotypic structure by including a basic interpretation of transcription and translation, which introduces neutrality through the proposed GP-map. This in turn allows for changes in the phenotypic neighbourhood, where phenes are created through various combinations of *amino acids*, based on translation tables (see Tables IV and V) and are in turn combined to create a phenotype. The pseudocode for the process is outlined in Algorithm 1, where P represents the population and g represents the generation.

The GP-map enables the encoding of a phenotype of length l by a genotype of length $l + n$, where n is the number of bits required to represent k symbols, with k being the size of the alphabet being used by the MMGA. As all the genotypes bits have an effect on the evolution of the phenotype there is no redundancy and neutrality is not trivial. The MMGA also offers the ability to alter the alphabet size and to implement a neighbourhood map which allow for different types of neutral traits represented in the genotype to represent the same phenotype.

The MMGA, has the advantage of allowing flexibility to exist in the phenotypic variational topology when moving between genotype and phenotype. To achieve this the size of the alphabet to be created (four in this case) and the number of representations required must be set. The multiple mappings are created by the MMGA and the size of the genome required to achieve this is then calculated by the MMGA. The MMGA randomly initialises the binary genome string and this in turn is converted using a series of mappings to convert genes into a combination of amino acids which are then combined and examined for neighbourhood equivalence to create a phene. An important point to note is that these neighbourhoods are designed to introduce variation into the phenotypic topology. Thereby allowing a phenotype to be encoded by a number of genotypes and that the phenotypic evolution can only be understood by monitoring the neutral traits. The phenes are then combined to form the phenotype.

```

Initialize MMGA;
for Number of runs do
  Initialise Individual Genomes P(g);
  Transcribe Genome to Amino Acids (g);
  Translate Amino Acids to Phenotype P(g);
  Evaluate P(g) (Phenotype fitness);
  for Number of Generations do
    for All members of P do
      Select P(g) from P(g-1);
      Crossover P(g) genotype level;
      Mutation P(g) genotype level;
      Transcribe Genome to Amino Acids P(g);
      Translate Amino Acids to Phenotype P(g);
      Evaluate P(g) (phenotype fitness);
    end
  end
end
End MMGA;

```

Algorithm 1: MMGA Pseudocode

To begin, the basic interpretation of transcription and translation used by the MMGA produces two kinds of phenes, ‘0’ or ‘1’ (as we want to use a binary representation), each made up of combinations of four amino acids represented by A, C, G and U. These phenes are then combined to create a phenotype which is evaluated for fitness. Beginning with a binary string, which represents the genotype, the MMGA maps pairs of binary bits into one of four characters A, C, G or T. These characters represent the first phase of the transcription phase where a *template* strand is created. Following this, the template strand maps onto a *coding* strand and the final stage of the transcription phase maps the coding strand onto RNA. Using a four letter alphabet (A, C, G and T), the RNA sequence is complementary to that of the DNA template strand and therefore is the same sequence as the DNA coding strand, with uracil (U) in place of thymine (T). Table III outlines the mapping process used by the MMGA to implement the transcription stage.

Transcription Stage		
Template Strand Map	Coding Strand Map	RNA Map
00 → A	A → T	T → U
01 → C	C → G	G → G
10 → G	G → C	C → C
11 → T	T → A	A → A

TABLE III

TRANSCRIPTION STAGE CONSISTING OF TEMPLATE MAP, CODING MAP & RNA MAP

Once the transcription phase is completed and the alphabetic characters have been combined, the translation stage uses the information and maps it against the translation tables created by the MMGA to examine neighbourhood equivalence to manufacture a phene, the MMGA can adjust the number of bits required to make up a phene, however in this paper the

authors have used a simplistic interpretation whereby a phene is represented by either a 0 or a 1. Tables IV and V are extracts from the the translation tables created by the MMGA. Table IV contains translation mappings for phene 0, while Table V contains translation mappings for phene 1. The number of different translation phase mappings used by the MMGA in this paper are one hundred and twenty eight for each phene (this can be adjusted if required, as one may wish to alter the level of neutrality present). The translation tables created by the MMGA introduce the phenotypic variability by ensuring that neutral traits exist and that each genotype can be viewed as a pairing of a phenotype and a genotypes neutral traits which allows for different neutral traits to be associated with the same phenotype.

Extract of Translation Table for Phene 0				
AAAA	AAAC	AAAG	AAAU	AACA
AACC	AACG	AACU	AAGA	AAGC
AAGG	AAGU	AAUA	AAUC	AAUG
ACAA	ACAU	ACCA	ACUG	...

TABLE IV

EXTRACT OF TRANSLATION TABLE FOR PHENE 0

Extract of Translation Table for Phene 1				
CAAA	CAAC	CAAG	CAAU	CACA
CACC	CACG	CACU	CAGA	CAGC
CAGU	CAUA	CAUC	CAUG	CAUU
CCAC	CCAG	CCAU	CCCC	...

TABLE V

EXTRACT OF TRANSLATION TABLE FOR PHENE 1

Table VI illustrates the mappings from genotype strings which are transcribed and translated into phenes for a three bit problem. We begin with a genome string, which is mapped to the template strand using the template mappings in Table III. This is then converted to derive the coding strand using the coding mappings in Table III. Once the coding strand has been created the next phase in the transcription stage is to create RNA. This is achieved by using the RNA mappings in Table III. Once we have created the RNA, the transcription stage is now complete. Translation now begins and the RNA is compared to the values contained in the translation Tables IV and V, which check for neighbourhood equivalence and then map to phenes. Once the phenes have been created they are then combined together to create a three-bit phenotype as illustrated Table VI .

One interesting feature of the transcription phase, is that the mapping is a form of complementing. This is an attempt to, in a basis way, recreate the bridging between the gene and the phene, as in natural RNA which is complementary to the template strand. Therefore, the mappings are included to mimic more closely, but still at a basic level, naturally occurring phenomenon and will be explored more in future work. The translation phase on the other hand, takes the output

Transcription & Translation - A Basic Interpretation			
Genome	11100001	11101100	10110001
Template Strand	TGAC	TGTA	GTAC
Coding Strand	ACTG	ACAT	CATG
RNA	ACUG	ACAU	CAUG
Phene	0	0	1
Phenotype	001		

TABLE VI
TRANSCRIPTION-TRANSLATION MAPPING

from transcription and maps it to a phene through neighbourhood equivalence with the translation tables generated by the MMGA. This allows for neutral traits to be included and are of importance as they allow increased redundancy and induce phenotypic variability.

IV. EXPERIMENTS

To examine the performance differences between a SGA and the MMGA, experiments were conducted over a number of fully deceptive landscapes which fall into the class of GA-hard problems. The landscapes chosen consist of a three-bit minimal deceptive problem as outlined by Goldberg in [15], a variation of the thirty-bit deceptive problem as described by Goldberg & Bridges in [11] which expands the three-bit problem into ten three-bit deceptive order-three subfunctions. And finally, a thirty-bit deceptive problem, which also expands the three-bit problem into ten three-bit deceptive order-three subfunctions, over a changing landscape as outlined in [22]. The experiments were conducted with a population of 200 (apart from the 3-bit deceptive problem which had a population of 20) over 200 runs; a crossover rate of 0.7 and a mutation rate of $1/l$, where l is the length of the chromosome. We chose a population of 200 to ensure adequate exploration of the search space and give both GAs a reasonable chance of success. The diagrams below indicate the average best performance of both GAs over the deceptive landscapes. For each group of experiments diversity is calculated at both a genotypic and phenotypic level using pair-wise hamming distance in order to gain an understanding to the level of variability present in the topology. For comparison purposes we normalised the data on the hamming distance measurements and use a scale of $[0 - 1]$, with 0 indicating convergence and 1 indicating maximum diversity.

A. Three-Bit Deceptive Problem

To gain an initial understanding of what effect the MMGA GP-map would have on variability we monitor the fitness levels of both GAs and the levels of both genotype and phenotype diversity present. The first set of experiments were run over a three-bit fully deceptive problem. Figure 1 illustrates the level of genotypic and phenotypic diversity in the population for both the MMGA and a SGA. However, the three-bit deceptive problem is relatively easy for both algorithms to solve, this is illustrated in Figure 2, which shows the average best fitness for both the SGA and the MMGA, with both variations of the

GA locating the global optimum (maximum fitness level of 30) quite quickly.

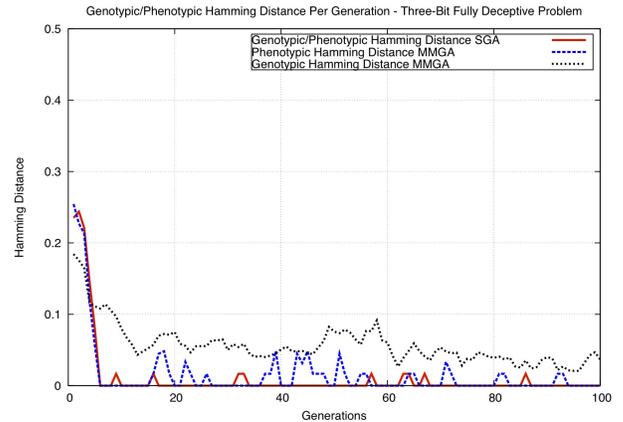


Fig. 1. Genotype/Phenotype Hamming Distance 3-Bit Deceptive Problem

However, looking again at Figure 1, the SGA converges very quickly and diversity, both genotypic and phenotypic, as they are one and the same when a standard binary representation GP-map is being used, disappear apart from the efforts of the mutation operator, indicating the lack of phenotypic variability. The MMGA on the other hand, through the non-trivial GP-map, maintains a level of diversity both at a genotypic level and to a lesser extent, at a phenotypic level indicating the presence of phenotypic variability. A t-test was used to determine if the results of the hamming distance experiments conducted were statistically significant. For each experiment, the results of the t-test analysis with a 95% confidence interval, indicated that the result of the three-bit deceptive problem hamming distance experiments were statistically significant with $P < 0.001$. Overall, when we examine Figures 1 and 2 it is difficult to argue for an increase in the level of phenotypic variability given the extra computational overhead associated with implementing the GP-map proposed. So although the MMGA exhibits the ability to solve the problem, in the case of the three-bit deceptive problem, it would appear that this landscape is relatively easy to solve and therefore we need to examine a more difficult landscape to see if there is any benefit associated with an increase in phenotypic variability.

B. Thirty-Bit Deceptive Problem

One failing of the 3-bit fully deceptive problem is that it is too small to really demonstrate a search strategy. The thirty-bit problem as outlined in [11] expands the three-bit problem into the sum of ten three-bit deceptive order-three subfunctions. The effect of this is to make it difficult for the SGA to solve as it tends to converge prematurely, with the subfunctions being drawn towards the deceptive attractor rather than the global optimum [11]. However, in Goldberg's thirty-bit function each three-bit subfunction is associated with the adjacent three-bit

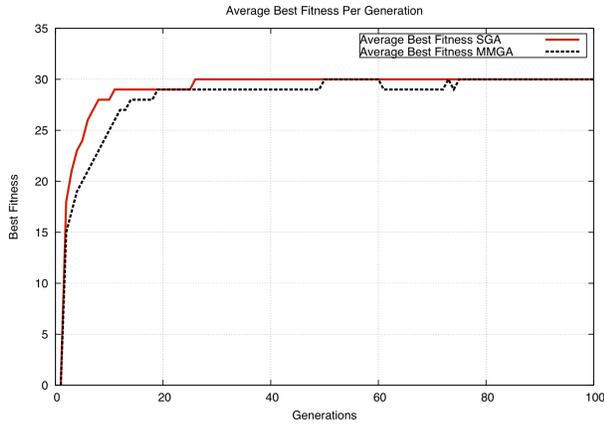


Fig. 2. Average Best Fitness 3-Bit Deceptive Problem

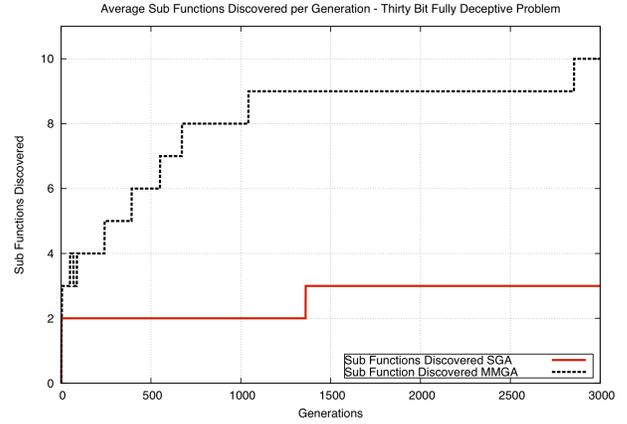


Fig. 5. Sub-Functions Discovered 30-Bit Deceptive Problem

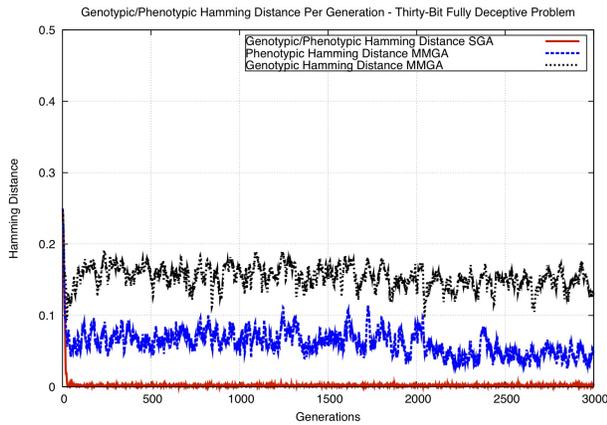


Fig. 3. Genotype/Phenotype Diversity 30-Bit Deceptive Problem

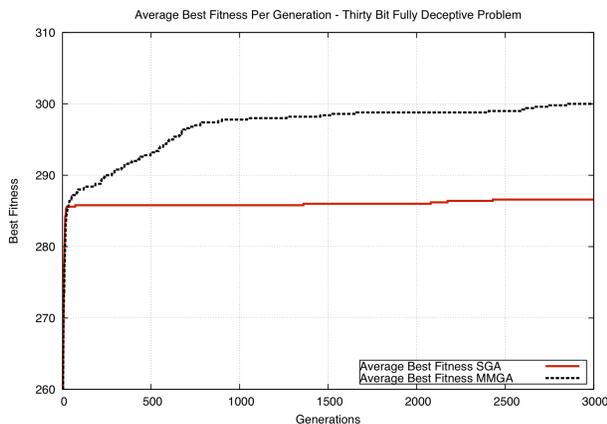


Fig. 4. Average Best Fitness 30-Bit Deceptive Problem

subfunction. So subfunction one is associated with bits one to three and subfunction two is associated with bits four to six and so on. This function is relatively difficult using most functional optimisation standards and the search space is quite large. To increase the level of difficulty we include a *loose ordering*, which makes the problem fully deceptive. This is achieved by increasing the defining length to twenty, where the defining length is the maximum distance between two defining symbols in a schema. For example, rather than having bits one to three linked together to create a subfunction, increasing the defining length to twenty means our subfunction now consists of bits one, eleven and twenty one for the first subfunction and bits two, twelve and twenty two for the second subfunction etc. We also carry out a count of the number of subfunctions discovered throughout the search, with ten being the maximum number achievable. By increasing the three-bit problem into ten three-bit sub-functions, as outlined, we create a thirty-bit fully deceptive landscape which through the increase in defining length for each subfunction, dramatically increases the level of problem difficulty. Through the fixed non-trivial GP-map, we increase the phenotypic variational topology, but we are still interested in understanding whether increasing phenotypic variation is beneficial in searching the deceptive landscape provided by the function.

Figure 3, highlights both the genotypic and phenotypic diversity for both the SGA and MMGA. Again, because of the type of representation and GP-map present in the SGA, both the genotypic and phenotypic diversity are the same. The SGA's genotypic and phenotypic diversity converges early on as it is drawn towards the deceptive attractor, indicating a lack of variability. The MMGA's performance differs from that of the SGA, as a level of diversity, both phenotypic and to a greater extent genotypic, is maintained throughout the search as the variational topology changes. The ability to increase the level of phenotypic variability through the GP-map enables the maintenance of diversity, particularly genotypic diversity

appears to assist in searching the landscape as the MMGA exhibits the ability to repeatedly locate the optimum solution where the SGA fails. Figure 4, illustrates the average best fitness achieved by both the MMGA and SGA per generation, with a fitness of 300 indicating optimisation. Figure 5, emphasises this from a different perspective by outlining the number of sub-functions discovered by both the SGA and MMGA. We can see that the SGA never manages to locate the entire ten sub-functions, whereas the MMGA succeeds in discovering all ten sub-functions, which is the global optimum. To ensure that the results were statistically significant we carried out a t-test analysis with a 95% confidence interval, the result of which indicated were statistical significance, with a p-values score < 0.001 . Overall from the fitness values outlined in Figure 4, the increase in phenotypic variability associated with the use of a non-trivial GP-map appears to be beneficial in searching this particular class of problem. To continue testing the effect of increased phenotypic variability we now introduce a dynamic landscape and examine the performance of the MMGA.

C. Changing Landscape Thirty-Bit Problem Deceptive Problem

As seen from the experiments outlined above, the MMGA possesses the capability to solve problems over a relatively difficult deceptive landscape. In order to continue to observe the performance of both the SGA and the MMGA, we use a landscape which changes the fitness function at a predefined point in the search. To obtain the effects of a dynamic environment we alter the fitness function during the search, which changes the landscape and creates a new global optimum. Table II outlines the changes made to the fitness function and shows our new deceptive attractor and new global optimum. In other words to change the landscape, at a predetermined point we change the fitness function from that shown in Table I to the fitness values shown in Table II.

In testing the MMGA over a changing deceptive landscape we are attempting to increase our understanding of the impact the multi-layered GP-map has as the structure of this landscape should give an advantage to the SGA. The reason for this is because the SGA gets drawn towards the deceptive attractor in the first stage of the search and when the landscape changes, the deceptive attractor becomes the new global optimum, therefore the SGA begins the search closer to this point. In Figure 6 we can see that both the SGA's genotypic and phenotypic diversity are drawn towards convergence. However, with the MMGA, we have increased phenotypic variability through the GP-map and maintain a level of diversity in the population. Again, a t-test analysis with a 95% confidence interval was used and the results of the hamming distance experiments were shown to be statistically significant.

Figure 7, shows the average best fitness per generation for both the SGA and the MMGA. Once the search reaches the half way point the fitness function changes and the landscape shifts where the global optimum prior to the changing of fitness functions, switches and becomes the deceptive attractor of the new fitness function. At this point the SGA is closer to

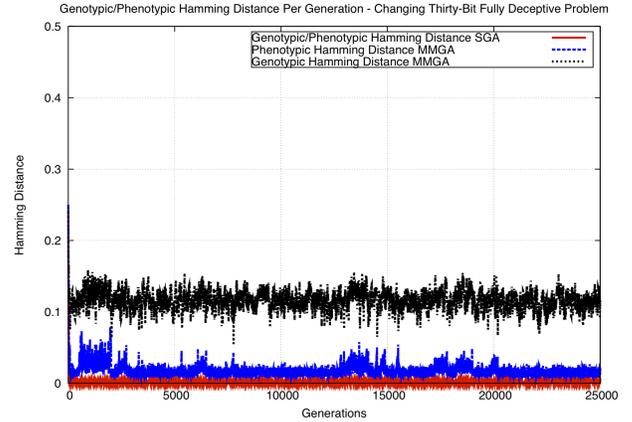


Fig. 6. Genotype/Phenotype Hamming Distance Thirty-Bit Changing Landscape

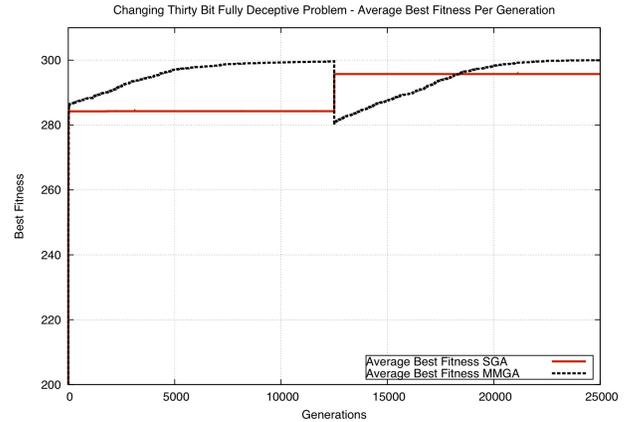


Fig. 7. Average Best Fitness Per Generation Thirty-Bit Changing Landscape

the new global optimum, while the MMGA is located directly on the deceptive attractor. As the search continues the SGA continues to converge and fails to locate the optimum (maximum fitness value of 300). The MMGA however, succeeds in locating the optimum, both before and after the landscape changes.

Figure 8, illustrates that in the first half of the search, the SGA is drawn away from the global optimum and converges on the deceptive attractor, solving only, on average, two sub-functions. The MMGA on the other hand, continues probing the landscape and discovers the global optimum, solving all ten sub-functions. When the landscape changes although the SGA is closer to the global optimum it still converges prematurely and fails to locate all ten sub-functions. While the MMGA continues to search after the landscape has changed and succeeds in optimising the problem and solving all ten

sub-functions. Thus indicating that the non-trivial GP-map can through an increase in phenotypic variability, succeed in solving a changing deceptive landscape. These results indicate the usefulness of an increase in phenotypic variability over this particular class of problem.

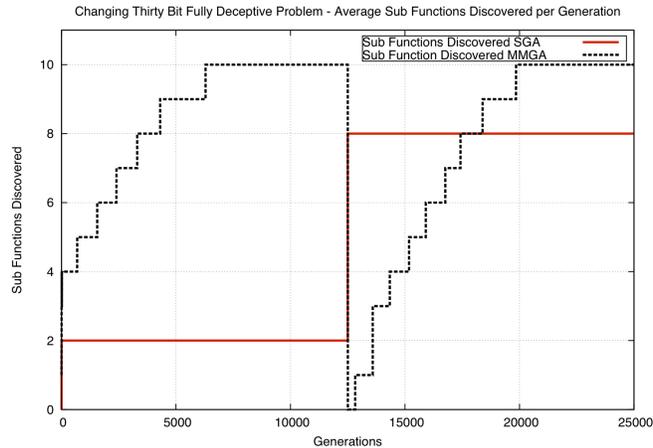


Fig. 8. Sub-functions Discovered Thirty-Bit Changing Landscape

V. CONCLUSION AND FUTURE WORK

The results indicate that the proposed multi-layered GP-map is effective in solving the deceptive problems presented, with its benefits increasing as the level of problem difficulty increases. It illustrates that neutrality or redundancy can have an critical impact on phenotypic variability in the case of a non-trivial GP-map. The non-injective GP-map allows different neighbourhoods that map, quite possibly, to different neighbourhoods of the same phenotype and thereby increase the phenotypic variability, similar to that outlined in [3]. By measuring the level of diversity at both the genotypic and phenotypic level, we attempted to quantify the levels or variability present. The results of the experiments appear to indicate that the introduction of a flexible phenotypic variational topology enhances the ability of a genetic algorithm to search more complex fully deceptive landscapes. By introducing a more flexible genotypic-phenotypic relationship through the use of a series of mappings, loosely based on the biological processes of transcription and translation, the MMGA repeatedly optimised both the fixed deceptive landscape and the changing deceptive landscape. This paper presents a non-trivial fixed GP-map, which includes a flexible phenotypic topology for mapping genes to phenes. The GP-map proposed allows the use of standard operators within a GA, without having to modify them and illustrates the benefits of the GP-map over a set of deceptive problems. Possible future work includes; further examination and analysis on phenotypic variability topologies and further investigation on how a multi-layered GP-map influence these topologies within the phenotypic

space. Comparisons of the performance of the MMGA with dynamic landscape algorithms such as SORIGA and ADMGA. Investigations into the use of operators within the transcription phase and altering the levels of neutrality and examine the influence this has in terms of changes in the locality of reference.

REFERENCES

- [1] P. F. Stadler, "Genotype-phenotype maps," *BIOLOGICAL THEORY*, vol. 2, 2006.
- [2] W. Fontana and P. Schuster, "Continuity in evolution: On the nature of transitions." in *Science* 280:14511455, 1998.
- [3] M. Toussaint, "Demonstrating the evolution of complex genetic representations: An evolution of artificial plants," in *In Proceedings of the 2003 Genetic and Evolutionary Computation Conference (GECCO 2003)*. Springer-Verlag, 2003, pp. 86–97.
- [4] S. M. Scheiner, "Genetics and evolution of phenotypic plasticity," in *Annual Review of Ecology and Systematics*, 1993, pp. 35–68.
- [5] S. Nolfi, O. Miglino, and D. Parisi, "Phenotypic plasticity in evolving neural networks," in *From Perception to Action Conference, 1994. Proceedings*, 2002, pp. 146–157.
- [6] F. Gruau, L. C. B. Lyon I, O. A. D. D. Doctorat, M. J. Demongeot, E. M. M. Cosnard, M. J. Mazoyer, M. P. Peretto, and M. D. Whitley, "Neural network synthesis using cellular encoding and the genetic algorithm." 1994.
- [7] M. Toussaint, "On the evolution of phenotypic exploration distributions," in *Foundations of Genetic Algorithms 7 (FOGA VII)*. Morgan Kaufmann, 2003, pp. 169–182.
- [8] R. Shipman, M. Shackleton, and I. Harvey, "The use of neutral genotype-phenotype mappings for improved evolutionary search," *BT Technology Journal*, vol. 18, pp. 103–111, October 2000.
- [9] A. Cangelosi, D. Parisi, and S. Nolfi, "Cell division and migration in a 'genotype' for neural networks (cell division and migration in neural networks)," *Network*, vol. 5, pp. 497–515, 1994.
- [10] D. E. Goldberg and C. L. Bridges, "An analysis of a reordering operator on a GA-hard problem," TCGA Report No. 88005, 1988.
- [11] D. E. Goldberg, B. Korb, and K. Deb, "Messy genetic algorithms: Motivation, analysis, and first results," *Complex Systems*, vol. 3, no. 5, pp. 493–530, 1990.
- [12] R. Tanese, "Distributed genetic algorithms," in *Proc. of the Third Int. Conf. on Genetic Algorithms*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1989, pp. 434–439.
- [13] J. J. Grefenstette and J. Baker, "How genetic algorithms work: a critical look at implicit parallelism." in *Proc. of the Third. International. Conf. on Genetic Algorithms*. Morgan Kaufmann, 1989.
- [14] L. D. Whitley, "Fundamental principles of deception in genetic search," in *Foundations of genetic algorithms*, G. J. Rawlins, Ed. San Mateo, CA: Morgan Kaufmann, 1991, pp. 221–241.
- [15] D. E. Goldberg, "Simple genetic algorithms and the minimal deceptive problem," *Genetic Algorithms and Simulated Annealing*, pp. 74–88, 1987.
- [16] A. D. Bethke, "Genetic algorithms as functional optimisers." Ph.D. dissertation, (Doctoral dissertation, University of Michigan) Dissertation Abstracts International, 41(9),3503B. (University Microfilms No. 81-06101), 1981.
- [17] J. H. Holland, *Adaptation in natural artificial systems*. Ann Arbor: University of Michigan Press, 1975.
- [18] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*. Reading, MA: Addison-Wesley Publishing Company, Inc., 1989.
- [19] M. Toussaint, G. Prof, D. ing Werner Seelen, P. Dr, and K. Goeke, "The evolution of genetic representations and modular adaptation," Ph.D. dissertation, 2003.
- [20] B. M. R. Stadler, P. F. Stadler, G. Wagner, and W. Fontana, "The topology of the possible: Formal spaces underlying patterns of evolutionary change," *Journal of Theoretical Biology*, vol. 2, no. 213, pp. 241–274, 2001.
- [21] F. Rothlauf, "Representations for evolutionary algorithms," in *GECCO (Companion)*, 2010, pp. 2535–2556.
- [22] S. Hill and C. O'Riordan, "Solving fully deceptive problems in changing environments," in *Artificial Intelligence Cognitive Studies (AICS)*, 2010, pp. 87–95.