Some Steps Towards Understanding How Neutrality Affects Evolutionary Search

Edgar Galván-López and Riccardo Poli

University of Essex, Colchester, CO4 3SQ, UK,
`egalva,rpoli@essex.ac.uk`

Abstract. The effects of neutrality on evolutionary search have been considered in a number of interesting studies, the results of which, however, have been contradictory. We believe that this confusion is due to several reasons. In this paper, we shed some light on neutrality by addressing these problems. That is, we use the simplest possible definition of neutrality, we consider one of the simplest possible algorithms, we apply it to two problems (a unimodal landscape and a deceptive landscape), which we analyse using fitness distance correlation, performance statistics and, critically, tracking the full evolutionary path of individuals within their family tree.

1 Introduction

Natural selection is a powerful theory which can explain the existence of adaptation in nature. However, it is unlikely that natural selection is the only force that directs evolution. Indeed, at molecular scale there is support for the idea that most evolutive variations are neutral [10]. This Neutral theory does not affirm that during evolution the genes are not making something useful, rather it suggests that different forms of the same gene are indistinguishable in their effects. The theory argues that mutations occurring during evolution are neither advantageous nor disadvantageous to the survival and reproduction of individuals, but that such random genetic drift should be considered in the study of the evolutionary process.

Some EC researchers have found neutrality to be beneficial for the evolutionary process while others have found it either useless or worse. We believe there are various reasons of these contradictory results and, by addressing them, we can start clarifying the effects of neutrality. The aims of this study are: (a) to understand how population flows in the search space are affected by the presence of neutrality in the evolutionary process, and (b) to identify under what circumstances neutrality may improve performance.

The paper is organised as follows. In the next section, we review previous work on neutrality. In Section 3 we describe our approach. In Section 4 the fitness distance correlation is computed for landscapes with neutrality. Section 5 provides details on the experimental setup used. In Sections 6 and 7 we present and discuss the results of experiments with unimodal and deceptive landscape problems and draw some conclusions.

2   Previous Work

Harvey and Thompson studied some effects of neutral networks in an evolvable hardware problem [7]. They defined the concept of potentially useful junk that refers to loci in a genotype that are functionless within the current context, but which may become functional with different values elsewhere in the genotype. They argued that with neutrality it is possible to reach a global optimum without worrying about premature convergence.

Banzhaf [2] proposed an approach where a genotype-phenotype mapping was used in the context of constrained optimisation problems. He argued that, very often, constraining the solution space leads to local optima which are difficult to escape from with traditional methods. He used high variability of neutral variants to escape from local optima on saddle surfaces.

Barnett [3] proposed a variant of NK landscapes which he called NKp landscapes. The idea was to introduce a parameter, p, which could vary the degree of neutrality present in the landscape and study the effects of neutrality in the evolutionary process. He claimed that with the presence of neutral networks with certain properties, it is possible to avoid to get stuck in local optima.

Shipman *et al.* [12] explored the benefits of neutrality in the context of a mapping based on an abstraction of genetic regulatory networks — a random boolean network. The mapping used in their experiments provided a very large degree of neutrality. They concluded that neutral drift allowed the discovery of many more phenotypes than would be the case with a direct encoding without redundancy. In [13] they proposed four different redundant mappings to study how neutrality influences the search. They found that redundancy was useful in three of their mappings and concluded that some kind of neutrality is crucial.

Smith *et al.* [14] analysed how evolvability was affected by the presence of neutral networks. For this purpose they used a system with an extremely complex genotype-to-fitness mapping. They concluded that the existence of neutral networks in the search space does not necessarily provide advantages because the population does not evolve any faster with neutrality. In [15] the same authors looked at the dynamics of the population rather than just the fitness, and argued that neutrality did not perform a useful role in an evolutionary robotic task.

Yu and Miller [18] showed that neutrality improves the evolutionary search process for a Boolean benchmark problem. They used Miller's Cartesian GP to measure explicit neutrality in the evolutionary process. They argued that mutation on active genes is adaptive because it exploits accumulated beneficial mutations, while mutation on inactive genes has a neutral effect on a genotype's fitness, yet it provides exploratory power by maintaining genetic diversity. Furthermore, in [19] they showed that neutrality was helpful and that there is a relationship between neutral mutations and success rate in a Boolean function induction problem. However, Collins [5] claimed that the conclusion that, in this problem, neutrality is beneficial is flawed. In [20] Yu and Miller also investigated neutrality using the simple OneMax problem. They used a theoretical approach and showed that neutrality is advantageous because it provides a buffer to absorb destructive mutations.

Igel and Toussaint [8] claimed that neutrality is necessary for self-adaptation and classified self-adaptation to classical and generalized self-adaptation. Both definitions are inspired from the genotype-phenotype mapping. They argued that neutrality could have benefit when the mapping is done in such a way that desirable phenotypes are represented more often than other ones.

3   Approach

We believe that the confusion regarding neutrality has several sources:(a) many studies have based their conclusions on performance statistics (e.g., on whether or not a system with neutrality could solve a particular problem faster than a system without neutrality) rather than a deep analysis of population dynamics, (b) studies often consider problems, representations and search algorithms that are relatively complex and, so, results represent the composition of multiple effects (e.g., bloat or spurious attractors in genetic programming), (c) there is not a single definition of neutrality and different studies have added neutrality to problems in radically different ways, and, (d) the features of a problem's landscape change when neutrality is artificially added, but rarely an effort has been made to understand exactly how.

In this paper, we shed some light on neutrality by addressing these problems. Firstly, we use the simplest possible definition of neutrality: a neutral network of constant fitness, identically distributed in the whole search space. Neutrality is "plugged into" the original non-redundant representation by adding an extra bit to the representation: when the bit is 1 the individual is on the neutral network (and, so, its fitness has a pre-fixed constant value), when the bit is 0, the fitness of the individual is determined by the coding bits as usual. Secondly, we consider one of the simplest possible algorithms (a mutation-only, binary genetic algorithm). Thirdly, we analyse population flows from and to the neutral network and the basins of attraction of the optima. Fourthly, we compare the percentage of success to find the optimum solution and the difficulty of the problem using fitness distance correlation. Finally, we use two problems with significantly different landscape features: a unimodal landscape (OneMax) where we expect neutrality to always be detrimental and a deceptive landscape (a trap function with different degrees of difficulty), where there are conditions where neutrality is more helpful than others.

In the presence of the form of neutrality discussed previously, the landscape is therefore divided into two areas of identical size: the *neutral layer* and the *normal layer*. However, we still only have one global optimum. So, the addition of neutrality comes at a cost since we are expanding the size of the search space without correspondingly expanding the solution space. Thus, we should expect to see benefits of neutrality (e.g., improved performance) only when neutrality modifies the search bias of an algorithm-problem pair in such a way to make it much more likely to (eventually) sample the global optimum. If this does not happen, or worse, if the original search bias is modified in such a way to make it harder to reach the global optimum, then we can be certain that neutrality will not help.

Neutrality is often reported to help in multimodal landscapes. So, in the case of our multimodal deceptive problem, should we expect a uniform neutral network to increase performance? And what sort of population dynamics should we expect? For analysis purposes, we further divide the normal and neutral layers into two regions depending on which of the two basins of attraction a string belongs to. We will term the resulting four areas "global neutral", "local neutral", "global normal" and "local normal".

Let us now consider whether a uniform neutral network could provide a performance improvement in the case of a trap landscape. We must first consider whether or not the neutral layer acts as an attractor or a repellent and for what proportion of the local and global areas. If, for example, the neutral layer has a very low fitness, then it should become harder for individuals to use it as a "tunnel" between the large basin of attraction of the local optimum and the narrow basin of attraction of the global optimum. In this case, the neutral layers would provide no advantage and, given that it doubles the search space, we should see a marked decrease in performance. If, instead, the neutral layers had a relatively high fitness, we should expect to see more individuals moving towards it. This means that there could be a flow of individuals from one basis of attraction to the other. This, however, would not in itself provide a performance improvement w.r.t. the case where no neutrality is used, because the flow is bidirectional and, so, individuals already in the global area may end up performing a random walk which leads them outside it. In addition, because the search space is still twice as big as the original while the solution spaces has still size 1, in order to beat the performance of the no-neutrality case, neutrality would need to provide a very significant "improvement" in search bias. These considerations have motivated our analysis and experiments. These are described in more detail in the following sections.

4   Fitness Distance Correlation

The fitness distance correlation (fdc) [9] measures the hardness of a landscape according to the correlation between the distance from the optimum and the fitness of solutions. The definition of fdc is quite simple: given a set $F = \{f_1, f_2, ..., f_n\}$ of fitness values of n individuals and the corresponding set $D = \{d_1, d_2, ..., d_n\}$ of distances to the nearest optimum, we compute the correlation coefficient r, as:

$$r = \frac{C_{FD}}{\sigma_F \sigma_D},$$

where:

$$C_{FD} = \frac{1}{n} \sum_{i=1}^{n} (f_i - \overline{f})(d_i - \overline{d})$$

is the covariance of F and D, and σ_F, σ_D, $\overline{f}$ and $\overline{d}$ are the standard deviations and means of F and D, respectively.

According to [9] a problem can be classified in one of three classes, depending of the value of r: (1) **misleading** ($r \geq 0.15$), in which fitness tends to increase with the distance from the global optimum, (2) **difficult** ($-0.15 < r < 0.15$),

for which there is no correlation between fitness and distance, and (3) **easy** ($r \leq -0.15$), in which fitness increases as the global optimum approaches.

There are some known weakness in the *fdc* as a measure of problem hardness [1, 11]. However, it is fair to say that the method has been generally very successful [9, 4, 17, 16]. The distance used in the calculations is, for binary search spaces, the Hamming distance.

In this work we will use *fdc* to evaluate problem difficulty with and without neutrality. Since we only consider problems where the fitness function is a function of unitation, we can rewrite C_{FD} in a more useful form.

For a search space of binary strings of length l, if we sample the whole search space in order to compute C_{FD}, we have:

$$C_{FD_f} = \frac{1}{2^l} \sum_{u=0}^{l} \binom{l}{u} (f(u) - \overline{f}_f)(u - \overline{u}_f)$$

where:

$$\overline{f}_f = \frac{\sum_{u=0}^{l} \binom{l}{u} f(u)}{2^l}$$

$$\overline{u}_f = \frac{l}{2}$$

where u represent the unitation class of strings.

As mentioned in the previous section, the form of neutrality we consider here is one where an extra bit is added to the representation. When the bit is set we say that an individual is in the neutral layer and its fitness is the constant value f_{layer}. So, when neutrality is present the size of the landscape is 2^{l+1}. Now C_{FD} is given by:

$$C_{FD_{neu}} = \frac{1}{2^{l+1}} \sum_{u=0}^{l} \binom{l}{u} \Big[(f(u) - \overline{f}_{neu})(u - \overline{u}_{neu}) + (f_{layer} - \overline{f}_{neu})(u + 1 - \overline{u}_{neu}) \Big]$$

where:

$$\overline{f}_{neu} = \frac{\frac{\sum_{u=0}^{l} \binom{l}{u} f(u)}{2^l} + f_{layer}}{2}$$

$$\overline{u}_{neu} = \frac{l+1}{2}$$

These calculations indicate that the introduction of neutrality does not necessarily imply a reduction of *fdc*. So, whether or not a problem is easier with neutrality depends on landscapes features and on f_{layer}.

5 Experimental Setup

We have used two problems to analyse neutrality. The first one is the OneMax problem which consist in maximizing the number of ones of a bitstring. Seen as a function of unitation the problem is represented by $f(u) = u$.

Table 1. Parameters

Parameter	Value
Length of the genome	10, 14 (+1 for neutrality)
Population Size	80
Generations	100
Mutation Rate (per bit)	0.02
Independent Runs	1,000

The second problem is a trap function, which is a deceptive function of unitation [6]. For this example, we have used the function:

$$f(X) = \begin{cases} \frac{a}{z}(z - u(X)) & \text{if } u(X) \leq z, \\ \frac{b}{k-z}(u(X) - z) & \text{otherwise} \end{cases}$$

where a is the deceptive optimum, b is the global optimum, and z is the slope-change location. Basically the idea is that there are two optima, a and b, and by varying the parameters k and z, we can make the problem easier or harder.

For the OneMax problem we have used chromosomes of length $l = 10$ while for the trap function we have used chromosomes of length $l = 14$, $k = 14$, $z = \{8, 9, 10, 11, 12, 13\}$, $a = 39$, $b = 40$, and sample size 4,000 to calculate fdc.

The experiments were conducted using a GA with fitness proportionate selection and bit-flip mutation. Runs were stopped when the maximum number of generations was reached. The parameters used are given in Table 1.

6   Results and Analysis

6.1   Performance comparison

In this section, we describe empirical evidence which corroborates the discussion presented above. Let's start by analysing the results for the OneMax problem. In Table 2 we show the fdc, the number of generations required to reach the optimum solution and the percentage of success in finding the optimum. As expected the problem is more difficult in the presence of neutrality. However, the degree of difficulty varies f_{layer}. fdc is a good heuristic measure of difficulty as one can see by comparing the fdc against the percentages of successes for different values on the neutral layer. In the case considered here ($l = 10$) the maximum achievable fitness is 10, and so a neutral layer with fitness 9 turns the search into a set of parallel random walks. It is not surprising then that, performance decreases so much with neutrality.

Now, let's consider the second problem - the trap problem. In this problem, the length of the genome is 14. As shown in Table 3, the bigger the value of the slope-change location z the harder the problem. When the neutral layer is present, regardless the value of f_{layer}, the number of generations required to reach the global optimum is bigger than when it is not present. This is easy to explain if we consider that the search space without neutrality is of size 2^l whereas with the presence of it is 2^{l+1}. When $8 \leq z \leq 11$, the percentage of

Table 2. Statistical information on the OneMax problem.

f_{layer}	fdc	Avg. Generations	% of Success
Not present	-1	8.07	100
6	-0.4922	10.68	100
7	-0.3010	12.72	100
8	-0.1604	21.73	94.7
9	-0.0650	35.02	34.2

Table 3. Statistical information on the trap problem.

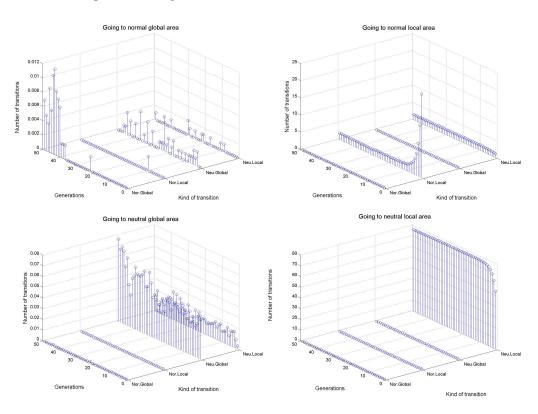| Value of z	fdc			Avr. Generations			% of Success		
	No neutral layer	f_{layer} 30	f_{layer} 38	No neutral layer	f_{layer} 30	f_{layer} 38	No neutral layer	f_{layer} 30	f_{layer} 38
8	0.42	0.35	0.33	10.81	40.25	29.60	38.7	19.8	1.7
9	0.74	0.45	0.40	8.65	31.26	24.50	17.5	12.1	1.3
10	0.90	0.51	0.45	6.83	12.45	22.60	7.7	1.3	1.9
11	0.96	0.55	0.45	3.85	16.75	17.20	1.7	1.1	1.2
12	0.99	0.57	0.48	0.25	6.20	7.55	0.2	0.7	0.7
13	0.99	0.59	0.49	-	7.90	24.30	0	0.6	0.9

runs that reached the optimum solution is bigger when neutrality is not present. However, the opposite happens when $12 \leq z \leq 13$. Moreover, when neutrality is not present the solution is either found after few generations or is not found at all. This does not hold when neutrality is present, as can be seen in Table 3. This means that there are complex dynamics going on between layers and regions of the landscapes, and that only by understanding these one can understand the effects of neutrality. We investigate them in the next section.

6.2   Family Tree

In a particular generation each individual can be in one of four areas: normal layer close to the global value, normal layer close to the local value, neutral layer close to the global value and neutral layer close to the local value. However, so far we have not studied where an individual in a specific layer came from. Fortunately, in a mutation based genetic algorithm each individual has only one parent. This makes it possible to track the origin of a sample point, and, in fact, the full evolutionary path of an individual within its family tree. This has allowed us to collect detailed statistics of population flows from one layer and region to another. To perform a full analysis we need to look at $2^4 = 16$ different parent/offspring transitions: a parent could be in any of four areas and his offspring could be in any of the same four areas.

In Figure 1 we show the result of the analysis of family trees for the trap function using $f_{layer} = 38$, $l = 14$ and $z = 13$. In all plots we can observe that the majority of offspring in an area came from parents already in that area. These

Fig. 1. Number of transitions to the normal global area (top left), normal local area (top right), neutral global area (bottom left) and neutral local area (bottom right), when the fitness of the neutral layer is 38.

are not the only sources, however, as shown in Figure 1 where can see that a small proportion of individuals in the neutral layer near the global optimum actually comes from neutral local area, indicating the presence of tunnelling.

7   Conclusions

There is considerable controversy on whether or not neutrality helps or hinders evolutionary search. In this paper we have highlighted some possible reasons for this situation. A particularly serious problem is that many studies are only based on performance statistics, rather than more in-depth investigations, and there is considerable variability in the problems, algorithms and representations used for benchmarking purposes. Also, there is neither a single definition of neutrality nor a unified approach to add neutrality to a representation. In this paper, we have made an effort to address these problems. We used *fdc* to assess if a problem gets easier or harder in the presence of neutrality. We complemented this with statistical information (e.g. average number of generations required to

solve a problem). We also recorded parent-offspring flows from and to the neutral network and the basins of attraction of the optima.

We argue that neutrality *may* be beneficial in some cases, but when it comes at the cost of an increased size of the search space without a corresponding expansion of the solution space, then any benefits it may bring via search bias, tunnelling ability, etc. may be insufficient to compensate for the additional search effort required by a reduced density of solutions. It is clear that the modifications in the original search bias of an algorithm produced by the addition of neutrality (at least of the form we have discussed here) are not always beneficial. We brought, for instance, the example of a unimodal landscape, where, as confirmed also experimentally, it is very hard to imagine any advantages in adding neutrality. Neutrality-induced bias, may, however, be very beneficial (so much so to fully overcome the inefficiencies due to an extended search space) in certain circumstances, like, for example, when the population is initialised in the wrong part of the search space. This is particularly common when dealing with infinitely large search spaces (e.g., the space of variable length strings and the space of computer programs), where it is impossible to initialise the population uniformly at random across the whole search space. This may be a further reason why certain studies have reported significant benefits when using neutrality (albeit of forms very different from the one used here).

We have shown that it is very difficult to infer the effects (or benefits) of neutrality without getting under the bonnet and looking at the population flows induced by the presence of neutrality. For example, as we have shown, in exactly the same conditions, a neutral network of low fitness changes the behaviour of a genetic algorithm in very different ways than a high-fitness neutral network.

Acknowledgments

References

1. L. Altenberg. Fitness distance correlation analysis: An instructive counterexample. In *Proceedings of the Seventh International Conference on Genetic Algorithms*, pages 57–64, San Francisco, CA, USA, 1997. Morgan Kaufmann Publishers Inc.
2. W. Banzhaf. Genotype-phenotype-mapping and neutral variation – A case study in genetic programming. In Y. D. *et al.*, editor, *Parallel Problem Solving from Nature III*, volume 866 of *LNCS*, pages 322–332. Springer-Verlag, 9-14 Oct. 1994.
3. L. Barnett. Ruggedness and neutrality: The nkp family of fitness landscapes. In R. A. *et al.*, editor, *Artificial Live VI: Proceedings of the Sixth International Conference on Artificial Life*, pages 18–27. MIT Press, 1998.
4. M. Clergue, P. Collard, M. Tomassini, and L. Vanneschi. Fitness distance correlation and problem difficulty for genetic programming. In W. B. L. *et al.*, editor, *GECCO 2002: Proceedings of the Genetic and Evolutionary Computation Conference*, pages 724–732, New York, 9-13 July 2002. Morgan Kaufmann Publishers.

5. M. Collins. Finding needles in haystacks is harder with neutrality. In H.-G. B. *et al.*, editor, *GECCO 2005: Proceedings of the 2005 conference on Genetic and evolutionary computation*, volume 2, pages 1613–1618, Washington DC, USA, 25-29 June 2005. ACM Press.
6. D. E. Goldberg. Construction of high-order deceptive functions using low-order walsh coefficients. *Ann. Math. Artif. Intell.*, 5(1):35–47, 1992.
7. I. Harvey and A. Thompson. Through the labyrinth evolution finds a way: A silicon ridge. In *Proceedings of the First International Conference on Evolvable Systems: From Biology to Hardware (ICES)*, pages 406–422. Springer-Verlag, 1996.
8. C. Igel and M. Toussaint. Neutrality and self-adaptation. In *Natural Computing*, pages 117–132, 2003.
9. T. Jones and S. Forrest. Fitness distance correlation as a measure of problem difficulty for genetic algorithms. In *Proceedings of the 6th International Conference on GA*, pages 184–192, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc.
10. M. Kimura. Evolutionary rate at the molecular level. In *Nature*, volume 217, pages 624–626, 1968.
11. R. J. Quick, V. J. Rayward-Smith, and G. D. Smith. Fitness distance correlation and ridge functions. In *PPSN V: Proceedings of the 5th International Conference on Parallel Problem Solving from Nature*, pages 77–86. Springer-Verlag, 1998.
12. R. Shipman, M. Schackleton, and I. Harvey. The use of neutral genotype-phenotype mappings for improved evolutionary search. *BT. Technology Journal*, 18(4):103–111, October. ISSSN 2000.
13. R. Shipman, M. Schakleton, M. Ebner, and R. Watson. Neutral search spaces for artificial evolution: A lesson from life. In M. B. *et al.*, editor, *Artificial Life: Proceedings of the Seventh International Conference on Artificial Evolution*, pages 162–169. MIT Press, 2000.
14. T. Smith, P. Husbands, and M. O'Shea. Neutral networks and evolvability with complex genotype-phenotype mapping. *Lecture Notes in Computer Science*, 2159:272–282, 2001.
15. T. Smith, P. Husbands, and M. O'Shea. Neutral networks in an evolutionary robotics search space. In *Congress on Evolutionary Computation: CEC 2001*, pages 136–145. IEEE Press, 2001.
16. M. Tomassini, L. Vanneschi, P. Collard, and M. Clergue. A study of fitness distance correlation as a difficulty measure in genetic programming. *Evolutionary Computation*, 13(2):213–239, Summer 2005.
17. L. Vanneschi, M. Tomassini, P. Collard, and M. Clergue. Fitness distance correlation in structural mutation genetic programming. In C. R. *et al.*, editor, *Genetic Programming, Proceedings of EuroGP'2003*, volume 2610 of *LNCS*, pages 455–464, Essex, 14-16 Apr. 2003. Springer-Verlag.
18. T. Yu and J. Miller. Neutrality and the evolvability of boolean function landscape. In *Fourth European Conference on GP*, pages 204–211. Springer-Verlag, 2001.
19. T. Yu and J. F. Miller. Needles in haystacks are not hard to find with neutrality. In J. A. F. *et al.*, editor, *Genetic Programming, Proceedings of the 5th European Conference, EuroGP 2002*, volume 2278 of *LNCS*, pages 13–25, Kinsale, Ireland, 3-5 Apr. 2002. Springer-Verlag.
20. T. Yu and J. F. Miller. The role of neutral and adaptive mutation in an evolutionary search on the onemax problem. In E. Cantú-Paz, editor, *Late Breaking Papers at the Genetic and Evolutionary Computation Conference (GECCO-2002)*, pages 512–519, New York, NY, July 2002. AAAI.