An empirical investigation of how and why neutrality affects evolutionary search

2 authors, including:

Riccardo Poli
University of Essex
487 PUBLICATIONS   **16,559** CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Improving Group Decision Making with Collaborative Brain-Computer Interfaces View project

Evolutionary Computation for Image Processing and Analysis View project

An Empirical Investigation of How and Why Neutrality Affects Evolutionary Search

Edgar Galván-López
University of Essex
Colchester, CO4 3SQ
United Kingdom
egalva@essex.ac.uk

Riccardo Poli
University of Essex
Colchester, CO4 3SQ
United Kingdom
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. Some researchers have found neutrality to be beneficial to aid evolution whereas others have argued that the presence of neutrality in the evolutionary process is useless. We believe that this confusion is due to several reasons: 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 more in-depth analysis of population dynamics, studies often consider problems, representations and search algorithms that are relatively complex and so results represent the compositions of multiple effects (e.g., bloat or spurious attractors in genetic programming), there is not a single definition of neutrality and different studies have added neutrality to problems in radically different ways. In this paper, we try to shed some light on neutrality by addressing these problems. That is, we use the simplest possible definition of neutrality (a neutral network of constant fitness, identically distributed in the whole search space), we consider one of the simplest possible algorithms (a mutation based, binary genetic algorithm) applied to two simple problems (a unimodal landscape and a deceptive landscape), and analyse both performance figures and, critically, population flows from and to the neutral network and the basins of attraction of the optima.

Categories and Subject Descriptors

I.2 [**ARTIFICIAL INTELLIGENCE**]: Automatic Programming; D.2.8 [**Software Engineering**]: Metrics—*complexity measures, performance measures*

General Terms

Algorithms

Keywords

Neutrality, Search Space, Genetic Algorithms

1. INTRODUCTION

The effects of neutrality [7] on evolutionary search have been considered in a number of interesting studies.

Harvey and Thompson studied some effects of neutral networks in an evolvable hardware problem [5]. In their work, they defined the concept of potentially useful junk that refers to loci in a genotype that are functionless within the current context, but with different values elsewhere in the genotype they may become functional. Harvey and Thompson argued that is possible to reach a global optimum without worrying about premature convergence if one uses neutrality in the evolutionary process.

Banzhaf [1] proposed an approach where a genotype-phenotype mapping was used in the context of constrained optimisation problems. Banzhaf 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. Keller and Banzhaf extended this work in [6].

Shipman *et al.* explored the benefits of neutrality in the context of a mapping based on an abstraction of a genetic regulatory network — a random boolean network [9]. The mapping used in their experiments provided a very large degree of neutrality. From their experimental results they concluded that neutral drift allowed the discovery of many more phenotypes than would be the case with a direct encoding without redundancy.

Elsewhere [10] the same authors proposed four different redundant mappings to study their effect in the evolutionary process and see how neutrality influences the search. They argued that redundancy was useful in three of their mappings. From this, they concluded that some kind of redundancy (neutrality) is crucial.

Smith *et al.* [11] analysed the effects of the presence of neutral networks on the evolutionary process. They observed how evolvability was affected by the presence of such 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, which allows the evolutionary process to escape from local optima, does not necessarily provide any advantage. This is because the population does not evolve any faster due to inherent neutrality. Later on [12] the same authors,

focused their research on looking at the dynamics of the population rather than looking at just the fitness, and argued that neutrality did not perform a useful role in an evolutionary robotic task.

Yu and Miller [13] showed in their work that neutrality improves the evolutionary search process for a Boolean benchmark problem. They used Miller's Cartesian GP [8] to measure explicit neutrality in the evolutionary process. They have explained that mutation on a genotype that has part of its genes active and others inactive may produce different effects: 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. Yu and Miller extended this work in [14] showing that neutrality was helpful and that there is a relationship between neutral mutations and success rate in a Boolean function induction problem. However, Collins [3] claimed that the conclusion that, in this problem, neutrality is beneficial is flawed.

Yu and Miller also investigated neutrality using the simple OneMax problem [15]. They attempted a theoretical approach in this work. With their experiments, they showed that neutrality is advantageous because it provides a buffer to absorb destructive mutations.

Chow [2] proposed a method that uses individuals which contains multiple chromosomes instead of a single chromosome. The idea of his approach was to apply genetic operators which do not maintain a one-to-one mapping between a genotypic bit and a phenotypic bit. Chow tested his approach in well known deceptive problems with good results.

Downing [4] added effortless neutral walks in his experiments. For this purpose he used Binary Decision Diagrams (BDDs). His idea was inspired by the fact that BDDs contain a lot of redundancy. Thanks to the presence of neutrality in BDDs, he found good results when solving even-n-parity problems ($7 \leq n \leq 17$).

As it can be seen from the brief summaries provided above, some researches 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 why contradictory results on neutrality have been reported and, by addressing them, we think we can start clarifying the sources of confusion. The aims of this study are:

- To understand how population flows in the search space are affected by the presence of neutrality in the evolutionary process, and, following from this analysis,

- to identify under what circumstances neutrality may help improve performance of the evolutionary process.

The paper is organised as follows. In Section 2, the approach used to carry out our research is described. Section 3 provides details on the experimental setup used. The results presented in this paper are discussed in Section 4 and conclusions are drawn in Section 5.

2. APPROACH

We believe that the confusion regarding neutrality has several reasons:

- 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 more in-depth analysis of population dynamics,

- studies often consider problems, representations and search algorithms that are relatively complex and so results represent the compositions of multiple effects (e.g., bloat or spurious attractors in genetic programming),

- there is not a single definition of neutrality and different studies have added neutrality to problems in radically different ways,

- the features of a problem's landscape change when neutrality is artificially added, but rarely an effort has been made to understand in exactly what ways.

In this paper, we try to shed some light on neutrality by addressing these problems. That is,

- 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 code by adding an extra bit to the representation: when the bit is set 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.

- We consider one of the simplest possible algorithms (a mutation based, binary genetic algorithm without crossover).

- We analyse both performance figures and, critically, population flows from and to the neutral network and the basins of attraction of the optima.

- We use two problems with significantly different landscape features: a unimodal landscape where we expect neutrality to always be detrimental and a multimodal deceptive landscape, where there are conditions where neutrality is more helpful than others.

In the presence of the form of neutrality discussed above, the landscape is therefore divided into two areas of identical size, which we will call the *neutral layer* and the *normal layer*. For bit strings of length l there are 2^l points in each 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. So, 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 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.

Think, for example, of a stochastic hill-climber or a mutation-only genetic algorithm on a unimodal landscape. If we start the search from some random point, we can easily imagine how the hill-climber will generate moves that improve fitness as many times as it will generate moves that make it worse. So, the hill-climber will obtain an improvement of

fitness every other time step, on average. Adding neutrality of the type indicated above to this landscape, however, will change radically the behaviour of the hill-climber. If the current solution has a fitness below the fitness of the neutral network, then the neutral network will be a preferable place to be in, and so, the hill-climber moves there and behaves like a random walk thereafter. If, instead, the fitness of the current point is above the fitness of the neutral network, then the neutral network will be a place to avoid, and so, the hill-climber will only accept improving moves on the normal layer. However, this are now generated with only half the probability they had in the absence of neutrality. So, in both cases the hill-climber cannot benefit from the introduction of a uniform neutral network in an otherwise unimodal landscape.

Neutrality is often reported to help in multimodal landscapes, in that it can create paths which connect local optima, thereby allowing the march towards the global optimum to never come to a complete stand still. 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: 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 our deceptive landscape. Naturally, 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 away from 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.

There may be situations, however, where the neutral system can provide very dramatic improvements. For example, one can modulate the benefits and drawbacks of neutrality by varying population size and using biased initialisation strategies. The latter are 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, which may be a further reason why certain studies have reported significant benefits when using neutrality (albeit of forms very different
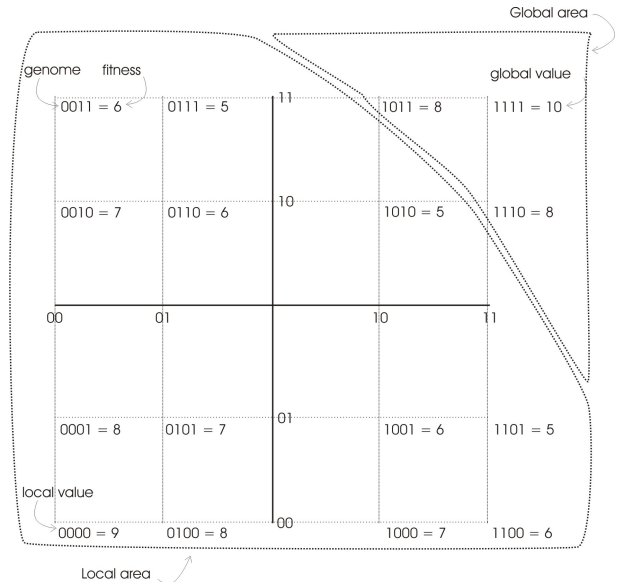

Figure 1: Example of deceptive problem used in our study.

Table 1: Summary of Parameters.

Parameter	Value
Length of the genome	8 (+1 for neutrality)
Population Size	20, 40, 60, 80, 100
Generations	300
Mutation Rate (per bit)	0.02
Independent Runs	100

from the one used here).

These considerations have motivated our experiments. These are described in more detail in the following section.

3. EXPERIMENTAL SETUP

We have used two problems to analyse neutrality. The first one is the OneMax problem. The problem is to maximise:

$$f(x) = \sum_i x_i,$$

where x is a binary string of length l, i.e., $x \in \{0,1\}^l$. Naturally, this problem has only one global optimum in $11 \cdots 1$, and, the landscape is unimodal.

The second problem is similar to a deceptive trap function, but it includes minor variations which make the landscape less symmetric. In this problem we have two optima: a global optimum at position $11 \cdots 1$ and a local optimum a position $00 \cdots 0$. The global optimum is given a fitness n, while the local optimum has fitness $n-1$. The remaining points in the landscape are assigned fitness values that decrease with the distance from one of the optima, in such a way that the basin of attraction of the global optimum is significantly smaller than the basin of attraction for the local optimum. This last feature makes the problem deceptive. Figure 1 shows an example of deceptive problem for the case of $l = 4$ (where we use $n = 10$). In our experiments we used the chromosomes of length $l = 8$ and $n = 40$.

The experiments were conducted using a GA with fitness

Table 2: Average number of generations required to reach the optimal solution for the OneMax problem. Length of the genome $l = 8$.

Population size	Without neutral layer	Value on neutral 7	Value on neutral 5
20	9.6	111.3	17
40	6.5	101.2	11.6
60	5.4	82.3	8.2
80	4.5	64.6	7.5
100	3.5	50.5	6.7

Table 3: Percentage of runs that reached the optimal solution for the Deceptive problem. Length of the genome $l = 8$. Random initialisation.

Population Size	Without neutral layer	Value on neutral 38	Value on neutral 23
20	61%	42%	51%
40	78%	56%	60%
60	81%	67%	72%
80	85%	81%	75%
100	93%	94%	84%

Table 4: Percentage of runs that were able to reach the optimal solution for the Deceptive problem. Length of the genome $l = 8$. Fixed initialisation.

Population Size	Without neutral layer	Value on neutral 38	Value on neutral 23
20	7%	26%	2%
40	9%	48%	6%
60	17%	68%	12%
80	17%	74%	21%
100	31%	86%	23%

proportionate selection and bit-flip mutation, run for 300 generations. To obtain more meaningful results, we performed 100 independent runs for each of the population sizes. Runs were stopped when the maximum number of generations was reached. The parameters we have used are summarised in Table 1.

When the neutral layer was used, we used two different values of fitness.

For the deceptive problem we have used two different methods of initialisation. The first method, which we will call *random initialisation*, creates the initial population randomly and uniformly across the whole search space. The second method, which we will call *fixed initialisation*, still creates the initial population at random, but this time individuals can only belong to a pre-fixed area: the basin of attraction of the local optimum. For OneMax we used random initialisation.

4. RESULTS AND ANALYSIS

4.1 Performance comparison

Let's start by analysing the results for the OneMax problem. In Table 2, we show the number of generations required to reach the optimal solution for the OneMax problem. As expected, the number of generations required to reach the optimal solution in the presence of explicit neutrality is bigger than when it is not present. In the case considered here ($l = 8$) the maximum achievable fitness is 8, and so a neutral layer with fitness 7 turns the search into a set of parallel random walks (as we explained earlier). It is not surprising that then, performance decreases so much with neutrality. When, instead, the fitness of the neutral layer is lower, 5, we see that the original character of the search is maintained, but, because half of the moves (mutations) are wasted (in the sense that offspring on the neutral layer are then not selected, and so, produce no progress), performance is inferior to the no-neutrality case by about a factor of 2.

Now, let's consider the second problem – the bimodal deceptive problem. In Tables 3 and 4, we present the percentage of runs that were able to reach the optimal solution with and without neutrality in the evolutionary process.

When using the random initialisation method and being neutrality present, when the fitness of the neutral layer is low (23), the percentage of runs that reached the optimal solution were lower than when neutrality was not present at all population sizes. This is to be expected, since, as we argued above, in this situation we have an increased search space but virtually no "tunnelling" ability (since random walks on the neutral layer can only be rare and very short due to its low fitness). As expected the situation is different when the neutral layer has a high fitness (38, which is only beaten by the global and local optima). For example, with small population sizes it is better to have a neutral layer with low fitness than with high fitness, while the opposite is true for larger populations. Also, interestingly, for large enough populations (e.g., 100), performance is not statistically different from the performance registered in the no-neutrality case. This means that there are complex dynamics going on between layers and regions of the landscapes, and that only by understanding these one can predict and understand the effects of neutrality. We investigate them in the next section.

Before doing that, we let us have a look at what happens when the population is away from the basin of attraction of the global optimum. When we have initialised our population in the local area, the presence of neutrality is beneficial when the fitness of the neutral layer is high enough to allow the easy crossing of the barrier between basins of attraction. As shown in Table 4, if the fitness of the neutral layer is too low, the benefits accrued by the more modest tunnelling ability provide by neutrality in this case are masked by the drop in performance due to an enlarged search space.

4.2 Population flows

First, let us consider where individuals reside and in what proportions in different phases of the search in the presence of neutrality. 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.

In Figure 2, we show how individuals tend to migrate between areas in the case of random initialisation, for two different population sizes and two different fitness values for the neutral layer. When the fitness assigned to the neutral layer is 23, the number of individuals in the neutral layer decreases very rapidly and stays low thereafter. This is easy to explain, being 23 the lowest value in the search space. On the other hand, when the value on the neutral layer is 38,

this layer rapidly acquires individuals, this effectively being the easiest way for the population to increase its fitness. This intake is at the expense of the normal layers which rapidly become less populated. This happens because initially most individuals in that layer have fairly low fitness compared with the neutral layer. However, after a certain number of generations the population is close enough to one of the optima and then being on the normal layer is no longer disadvantageous. Indeed, in the case of populations of size 20, we see that eventually both layers come to have approximately the same number of individuals (Note that plots are averages).

With a population of 100, instead, most runs end up with the population around the global optimum, and this is the reason why, the plot in the bottom right corner of Figure 2, individuals in the normal layer in the local region keep decreasing monotonically.

In Figure 3, we show how individuals migrate between areas in the case of fixed initialisation, for two different population sizes and two different fitness values for the neutral layer. When the fitness assigned to the neutral layer is 23, the number of individuals in the neutral layer decreases very rapidly and stays low thereafter, like for random initialisation. In these conditions, a population of 20 is clearly insufficient to avoid the attractor at the local optimum. With larger populations, the global optimum can be reached given enough mutation steps. When the fitness of the neutral layer is 38, this layer rapidly acquires individuals, as for random initialisation. The individuals in this layer are effectively drifting and so, some reach the neutral layer in the basin of attraction of the global optimum, as indicated by the plots for either population size. The individuals there keep drifting on the neutral layer until they are close enough to the global optimum that offspring in the neutral layer start being reasonably competitive with those on the neutral layer. This multi-stage process is the reason for the slower growth in the number of individuals in the normal layer near the global optimum. This explanation shows that what normally is seen as a simple "tunnelling" effect provided by neutrality may in fact be a much more complicated process.

So far, we have seen that individuals tend to move from one area to another. However, we can not say from where an individual came from being in a specific layer. 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. Due to space limitations, we will report only the data for the Deceptive problem, running the genetic algorithm with a population of size 100 and using the random initialisation method.

We show the results of the analysis of family trees using two different values for the neutral layer: 23 (shown in Figure 4) and 38 (shown in Figure 5). In all plots we can observe that the majority of offspring in an area came from parents already in that area. These are not the only sources, however, as shown in Figure 4 (lower left) where we can see that a small proportion of individuals in the neutral layer near the global optimum actually comes from the corresponding

normal layers (which is, of course, due to mutations to the "neutrality bit"). Another example is in Figure 5 (top right), which shows a small flow of individuals from the neutral local area to the corresponding normal area. More generally the plots confirm the observations made above.

5. 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 considerably 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, by analysing performance figures, population distribution w.r.t. the basins of attraction of optima and the neutral network in the landscape, and 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 itself, 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 a search bias, tunnelling ability, etc. may be insufficient to compensate for the additional search effort required by a reduce density of solutions. We also argue 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.

In the paper we have shown, however, 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, we have shown that 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.

6. REFERENCES
[1] 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, Jerusalem, 9-14 Oct. 1994. Springer-Verlag.
[2] R. Chow. Evolving genotype to phenotype mappings with a multiple-chromosome genetic algorithm. In
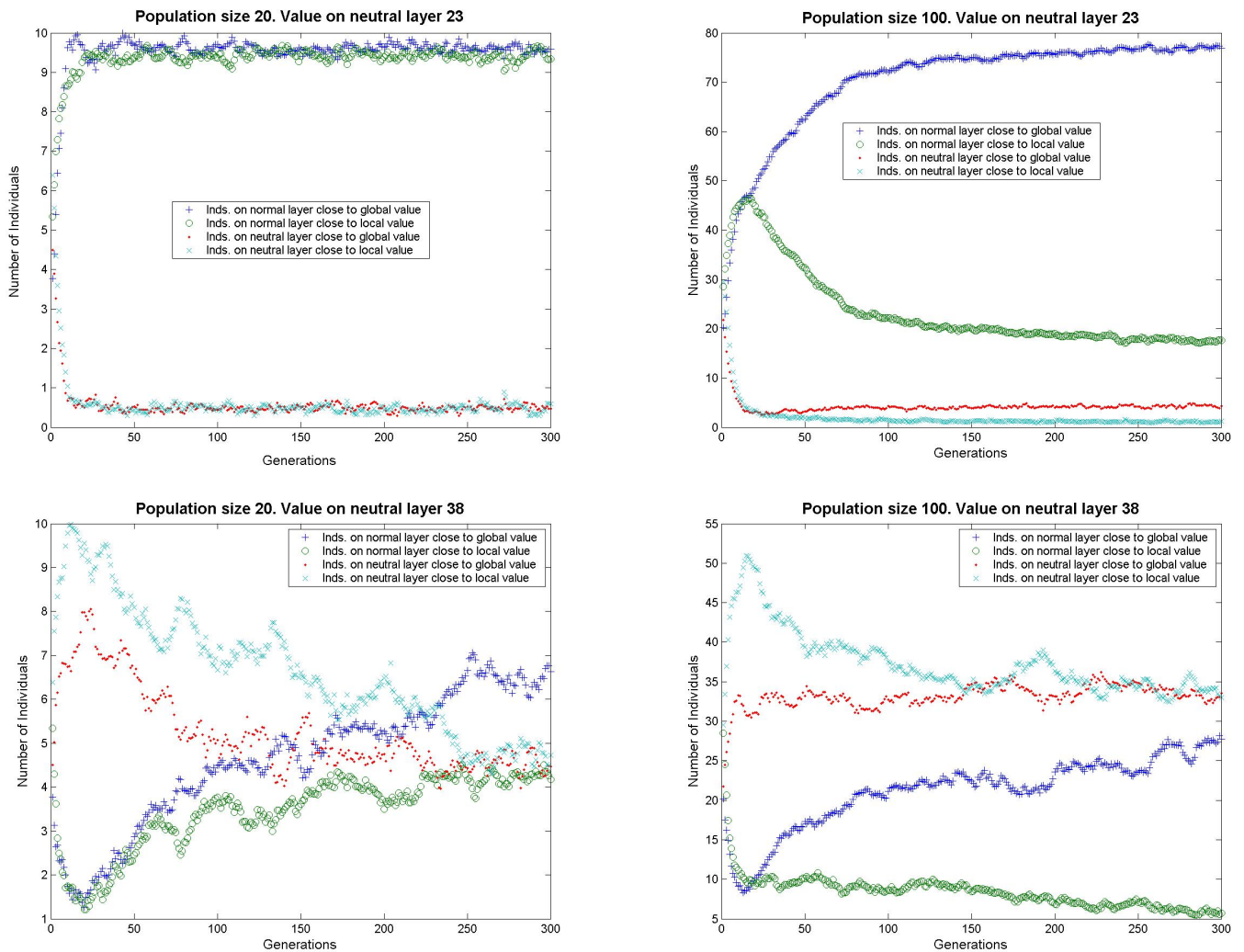
Figure 2: Number of individuals situated in one of the four parts of the landscape. Random initialisation. Fitness of neutral layer 23 (top) and 38 (bottom).

K. D. *et al.*, editor, *GECCO 2004: Proceedings of the 2004 Conference on Genetic and Evolutionary Computation*, volume 1, pages 1006–1017, Seattle WA, USA, 26-30 June 2004. Springer-Verlag.

[3] M. Collins. Finding needles in haystacks is harder with neutrality. In H.-G. Beyer, U.-M. O'Reilly, D. V. Arnold, W. Banzhaf, C. Blum, E. W. Bonabeau, E. Cantu-Paz, D. Dasgupta, K. Deb, J. A. Foster, E. D. de Jong, H. Lipson, X. Llora, S. Mancoridis, M. Pelikan, G. R. Raidl, T. Soule, A. M. Tyrrell, J.-P. Watson, and E. Zitzler, editors, *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.

[4] R. M. Downing. Evolving binary decision diagrams using implicit neutrality. In *CEC 2005: IEEE Congress on Evolutionary Computation (CEC'2005)*, pages 5–11, Edinburgh, Scotland, 2005.

[5] 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.

[6] R. E. Keller and W. Banzhaf. Genetic programming using genotype-phenotype mapping from linear genomes into linear phenotypes. In J. R. Koza, D. E. Goldberg, D. B. Fogel, and R. L. Riolo, editors, *Genetic Programming 1996: Proceedings of the First Annual Conference*, pages 116–122, Stanford University, CA, USA, 28–31 July 1996. MIT Press.

[7] M. Kimura. Evolutionary rate at the molecular level. In *Nature*, volume 217, pages 624–626, 1968.

[8] J. F. Miller and P. Thomson. Cartesian genetic programming. In R. Poli, W. Banzhaf, W. B. Langdon, J. F. Miller, P. Nordin, and T. C. Fogarty, editors, *Genetic Programming, Proceedings of EuroGP'2000*, volume 1802 of *LNCS*, pages 121–132, Edinburgh, 15-16 Apr. 2000. Springer-Verlag.
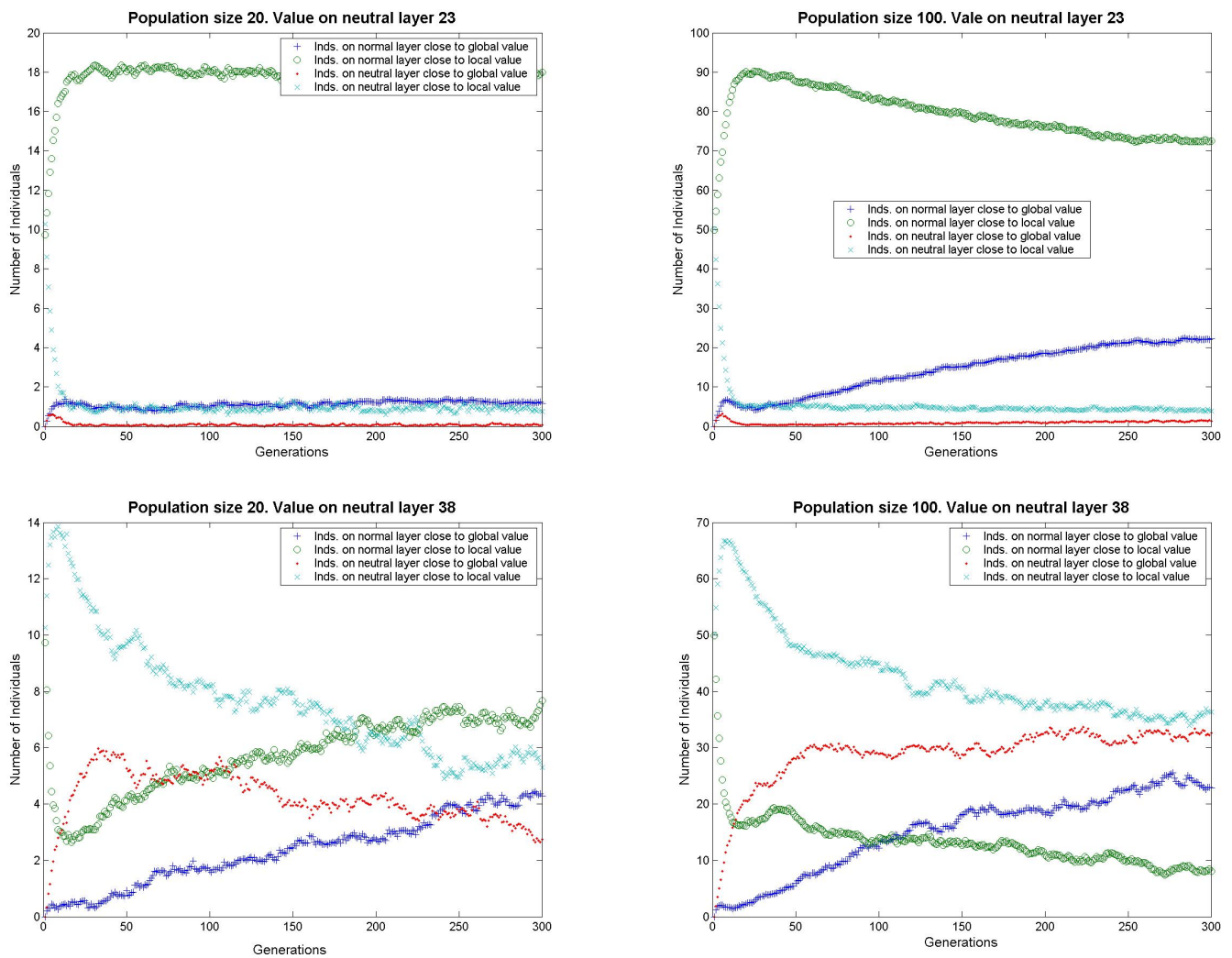
Figure 3: Number of individuals situated in one of the four parts of the landscape. Fixed initialisation. Fitness of neutral layer 23 (top) and 38 (bottom).

[9] 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.

[10] R. Shipman, M. Schakleton, M. Ebner, and R. Watson. Neutral search spaces for artificial evolution: A lesson from life. In M. Bedau, S. Rasmussen, J. McCaskill, and N. Packard, editors, *Artificial Life: Proceedings of the Seventh International Conference on Artificial Evolution*, pages 162–169. MIT Press, 2000.

[11] 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.

[12] 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.

[13] T. Yu and J. Miller. Neutrality and the evolvability of boolean function landscape. In *Fourth European Conference on Genetic Programming*, pages 204–211. Springer-Verlag, 2001.

[14] T. Yu and J. F. Miller. Needles in haystacks are not hard to find with neutrality. In J. A. Foster, E. Lutton, J. Miller, C. Ryan, and A. G. B. Tettamanzi, editors, *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.

[15] 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.
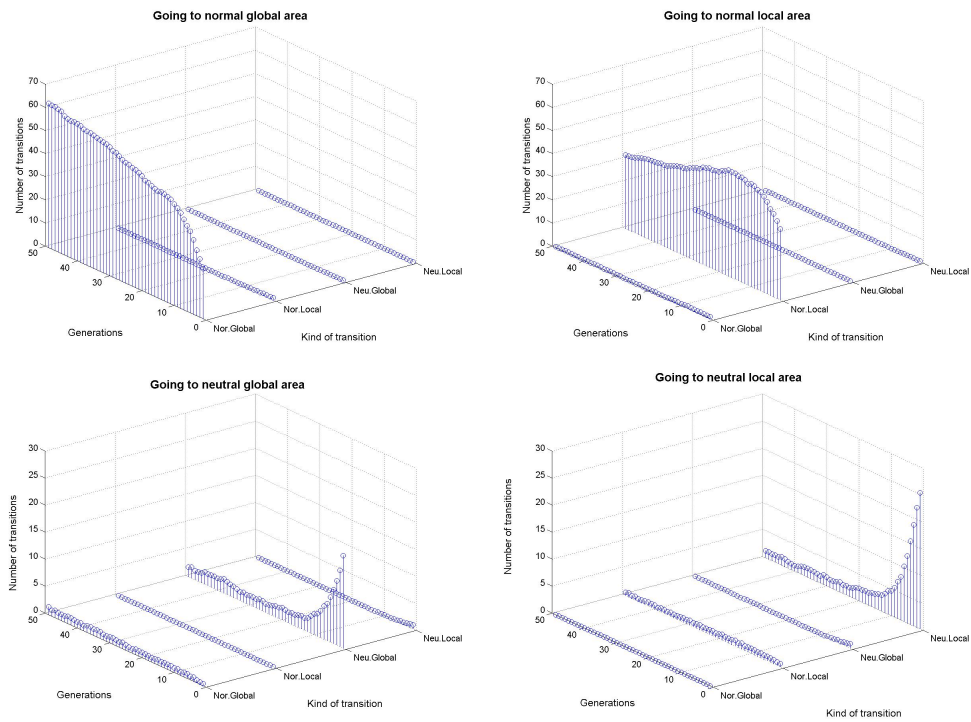
Figure 4: Number of transitions to the normal global area (top left), normal local area (top right), neutral global area (bottom left) and neutral local area (bottom right), when the fitness of the neutral layer is **23**. Random initialisation and population size 100.
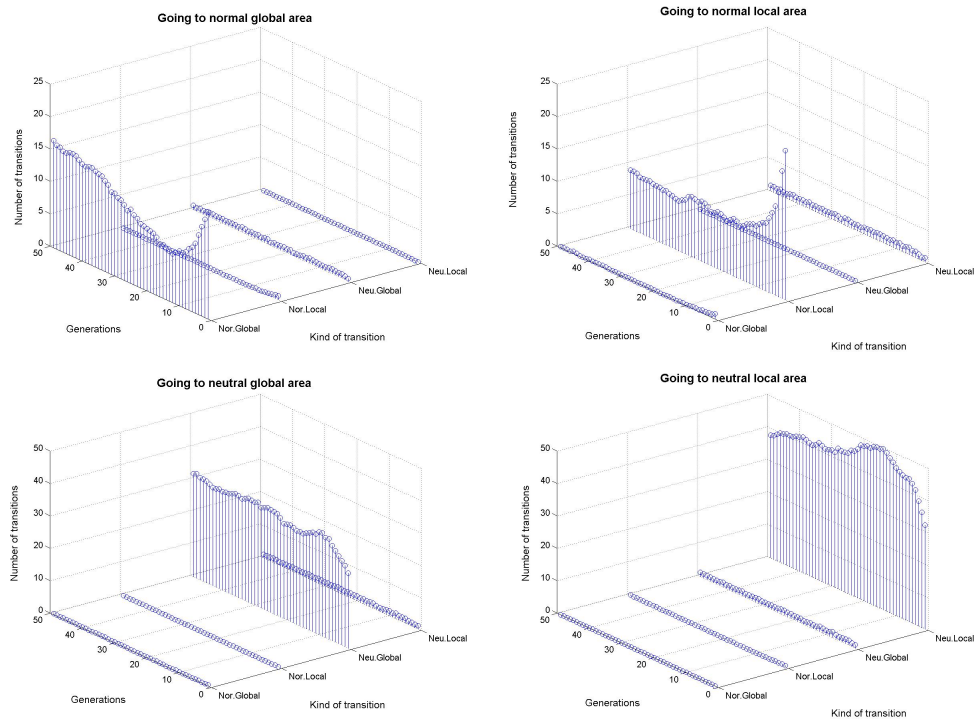

Figure 5: Number of transitions to the normal global area (top left), normal local area (top right), neutral global area (bottom left) and neutral local area (bottom right), when the fitness of the neutral layer is **38**. Random initialisation and population size 100.