

Smooth Operator?

Understanding and Visualising Mutation Bias

Seth Bullock

Informatics Research Institute, School of Computing, University of Leeds
seth@comp.leeds.ac.uk

Abstract. The potential for mutation operators to adversely affect the behaviour of evolutionary algorithms is demonstrated for both real-valued and discrete-valued genotypes. Attention is drawn to the utility of effective visualisation techniques and explanatory concepts in identifying and understanding these biases. The *skewness* of a mutation distribution is identified as a crucial determinant of its bias. For redundant discrete genotype-phenotype mappings intended to exploit neutrality in genotype space, it is demonstrated that in addition to the mere *extent* of phenotypic connectivity achieved by these schemes, the *distribution* of phenotypic connectivity may be critical in determining whether neutral networks improve the ability of an evolutionary algorithm overall.

Mutation operators lie at the heart of evolutionary algorithms. They corrupt the reproduction of genotypes, introducing the variety that fuels natural selection. However, until recently, the process of mutation has taken a back seat to the more dramatic genetic operators. Sexual recombination (the splicing together of genetic material from multiple parents) is often regarded as the major source of an evolutionary algorithm’s ability to discover fit phenotypes (the evolutionary programming paradigm being a notable exception). While mutation is required to introduce novel genetic material, it is recombination’s role in assembling groups of well-adapted alleles that is often concentrated upon [1]. However, doubts concerning the validity of this “building block hypothesis” have recently undermined the notion that the power of evolutionary algorithms can be identified with the role of sexual recombination [2].

Moreover, recent work on neutrality in genetic encodings [3] suggests that mutation events and the character of an evolutionary algorithm’s mutation space may have a greater role to play in the dynamics of evolutionary algorithms than had heretofore been appreciated. Rather than considering evolving populations to be engaging in some kind of hill-climbing via parallel assessment of different combinations of co-adapted alleles, Kimura’s [4, 5] neutral theory of evolution proposes that a more useful image is of populations percolating through “neutral networks” of adjacent points in genotype space that each code for phenotypes with equivalent fitness. It is contended that such populations, rather than tending to converge at the peaks of local optima, will continue to diffuse through a succession of neutral networks of increasing fitness. While it has been suggested that some difficult evolutionary optimisation problems already exhibit neutral

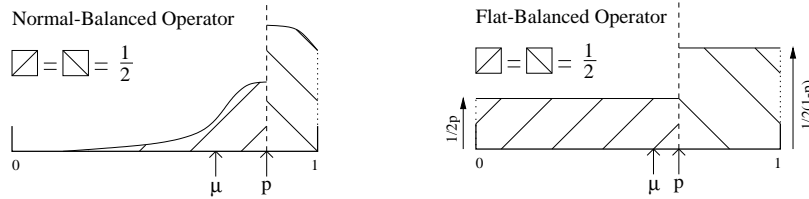


Fig. 1. Probability density functions for two mutation operators that perturb a real-valued parental gene, p , upwards or downwards with equal chance. Mean mutant destination is denoted μ . Mutants must lie within the legal range $[0, 1]$. For the *Normal-Balanced* operator (left), once direction of mutation has been determined, values are repeatedly drawn from the appropriate half of a normal distribution with mean p until a legal mutant value is achieved. For the *Flat-Balanced* operator (right), upwards mutants are drawn from a uniform distribution over all values in the range $[p, 1]$ while downwards mutants are drawn from the range $[0, p]$. Three additional operators are explored in this paper, but are not depicted. *Flat* draws mutants from a flat distribution over the legal range $[0, 1]$. *Absorb* draws mutants from a normal distribution with mean p , replacing illegal mutants by the nearest legal mutant value. *Reflect* behaves as *Absorb*, but any illegal mutant lying a distance d beyond a legal boundary is replaced by a value lying d within that boundary.

networks [6], others have sought to develop genetic encodings that encourage neutral networks in the hope that this will improve the ability of evolutionary algorithms to find optimal solutions in general [7, 8].

But how well do we understand the workings of mutation operators? Although they are often very simple to code, they come in many flavours, and previous work [9] has demonstrated that they often exhibit biases that may remain undetected, despite having appreciable effects on the course of artificial evolution — arbitrarily steering populations away from particular areas of the search space and toward others, for example.

This paper will begin by exploring a very simple pair of mutation operators (see Fig. 1). Since these operators, or ones like them, are used widely whenever genotypes feature real values which are constrained to lie within some legal range, analysing them serves a direct purpose in increasing our understanding of the component parts of evolutionary algorithms. But furthermore, in demonstrating their counter-intuitive behaviour and the biases that are inherent in even these simple operators, we may be able to refine our intuitions and develop useful explanatory concepts that allow us to gain insights into the more complex genetic encodings explored in the latter half of the paper.

1 Perturbing Bounded Continuous-Valued Traits

Although the canonical genetic algorithm (GA) employs binary genotypes, many practitioners find it useful to encode members of the evolving population as vectors of real values. Often these values must lie within some legal range. For

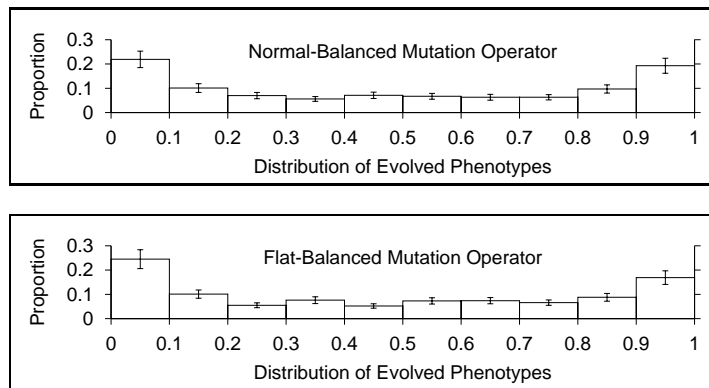


Fig. 2. Each histogram depicts the distribution of trait values for 20 populations of 1000 single-trait organisms after 5000 generations of evolution on a flat fitness landscape. For the Normal-Balanced operator, perturbations were drawn from a normal distribution with zero mean and standard deviation 0.2. An unbiased mutation operator is expected to exhibit a flat distribution, since no value is more likely to evolve than any other. Both of the mutation operators simulated here favour extreme-valued traits.

instance, consider using a GA to evolve a population of images, each encoded as a vector of real-valued RGB triples, e.g., $\{\{0.1, 0.9, 0.3\}, \{0.4, 0.6, 0.9\}, \dots\}$. Point mutations that *perturb* these RGB components by some small random amount must generate mutant values that remain within the legal range $[0, 1]$. In previous work [9], it was demonstrated that in dealing with illegal values, mutation operators can often introduce a *mutation bias* that either favours trait values close to their legal limits, or values that are far from these limits. In either case, such biases may adversely affect the course of evolution (tending to favour bold or dull images in the example being considered here) perhaps retarding evolutionary optimisation or introducing artefacts into evolutionary simulation models.

One mutation operator, here dubbed *Normal-Balanced* (which was not considered in [9], but was suggested in response to this work) is depicted in Fig. 1 (left), alongside a similar but simpler *Flat-Balanced* mutation operator which proves more amenable to formal analysis. Proponents of the former scheme were of the opinion that it would be able to deal with the possibility of illegal mutants without biasing a GA’s evolutionary dynamics.

However, to some extent, intuitions regarding the behaviour of these mutation operators conflict. The fact that mutation rate remains constant across the parental range, and that the likelihood of upwards and downwards mutations are always equal might be taken to indicate that the operators will *not* bias evolving populations. However, the fact that, for any parental value, the expected mutant value (denoted μ in Fig. 1) lies towards the centre of the legal range might suggest that the operators may bias populations *away* from extreme-valued traits.

In order to discover the behaviour of these mutation operators, we can evolve populations on a flat fitness surface, and check whether there exist particular trait values that are more likely to evolve than others (see Fig. 2). In addition, since the Flat-Balanced operator is relatively simple, it is straightforward to *derive* the expected probability density function for the trait values of a population after a period of mutation. Consider the mutant phenotype x of a parental phenotype p (Fig. 1, right). If $p < x$, the probability density function $f(x)$ takes the value $\frac{1}{2(1-p)}$, while if $p > x$, $f(x) = \frac{1}{2p}$. After some period of mutation. . .

$$f(x) = \int_0^x f(p) \frac{1}{2(1-p)} dp + \int_x^1 f(p) \frac{1}{2p} dp$$

That is, the value of the probability density function at x is equal to the contribution from every parental value lower than x added to the contribution from every parental value higher than x . What shape is this function?

$$\begin{aligned} f'(x) &= f(x) \frac{1}{2(1-x)} - f(x) \frac{1}{2x} \\ f'(x) &= -\frac{1}{2} f(x) \left(\frac{1}{x} + \frac{1}{(x-1)} \right) \\ f(x) &= C e^{-\frac{1}{2} \int_0^x \frac{1}{y} + \frac{1}{(y-1)} dy} \\ f(x) &= C e^{-\frac{1}{2} (\ln |x| + \ln |1-x|)} \\ f(x) &= \frac{C}{\sqrt{(x(1-x))}} \end{aligned}$$

Since it must be the case that $f(x) \geq 0 \forall x \in [0, 1]$, it follows that $C > 0$, and thus that $f(x)$ is a convex parabola with a minimum at $x = 0.5$. Hence we would predict from this analysis that the mutation operator would tend to push populations towards the extremes of the legal range.

The same general conclusion could be reached (via a more involved analysis) for the Normal-Balanced mutation operator.

Why do these mutation operators bias populations towards extreme-valued traits? Neither simulating their performance, nor deriving their character formally, gives us much insight into the relationship between the form of the operators and their behaviour. The frequency distributions in Fig. 3 begin to point us in the right direction by clarifying the nature of the over-representation of extreme-valued mutants, and highlighting the relative lack of symmetry in the distribution of mutants generated by the biased operators compared to that of an unbiased operator. Fig. 4 further demonstrates that, unlike considerations of mutation *drift* (calculated as the expected mutant destination) attention to the manner in which *skewness* (calculated as the difference between the mean and median values of mutant distributions) varies across the range of possible parental trait values allows us to distinguish biased from unbiased operators.

When a distribution of mutant values is skewed *away* from the extremities of a range (i.e., the median mutant is further from the centre of the legal range than the mean mutant), mutants falling towards the boundary will also fall closer together than mutants falling further from the boundary. For mutation operators such as Absorb, and the two Balanced operators, the increased *density* of the more extreme mutants creates a “hot-spot”, with positive feedback ensuring that more and more mutants fall closer and closer to the boundary *and* to each other. Although, as this is happening, the *mean* mutant may remain far from the legal

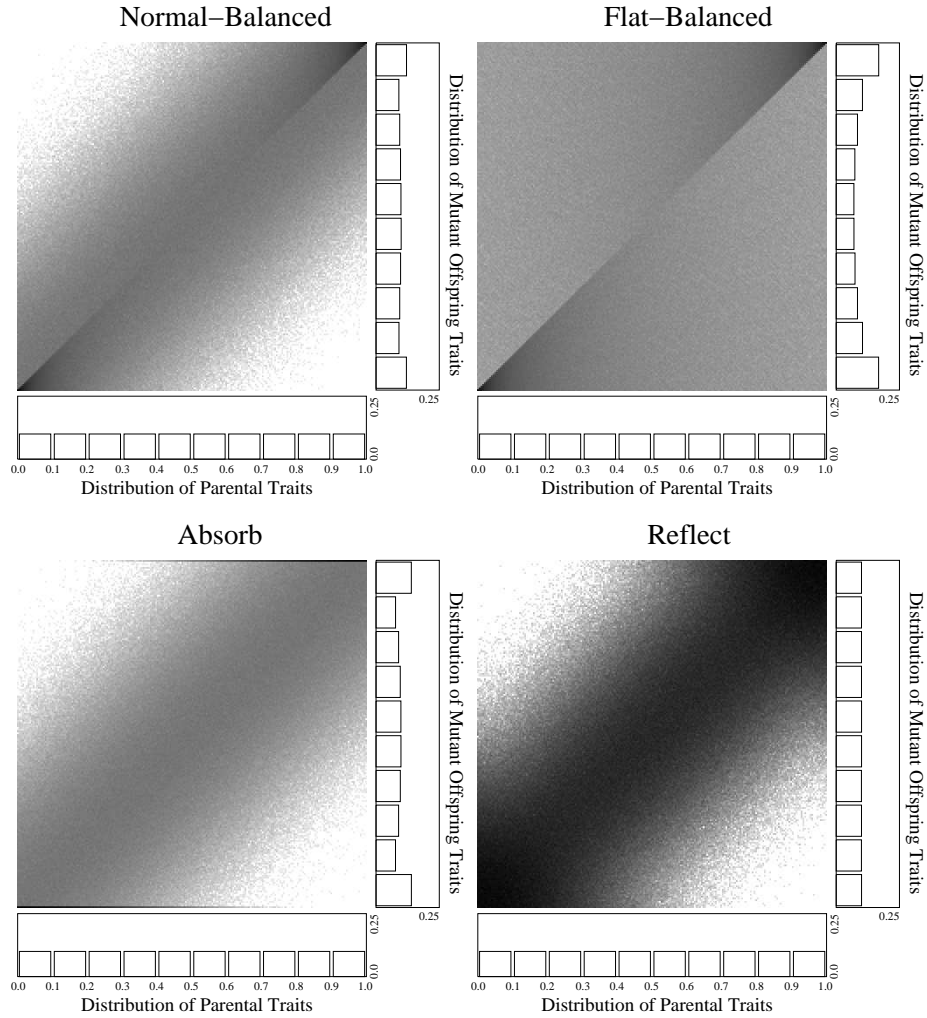


Fig. 3. For each mutation operator, the manner in which the distribution of mutants varies with parental trait value is plotted. The grey-level of each cell (x, y) in a 200×200 array represents the number of mutants in bin y after 10,000 randomly generated parents from bin x were each mutated once. The resulting aggregate distribution of mutant trait values is depicted as a histogram to the right of each array (parental values being depicted by the uniform histogram beneath each array). Deviation from uniformity in this right-hand histogram is indicative of mutation bias. Notice the dense areas in the corners of the Normal-Balanced, Flat-Balanced, and Absorb plots, indicating an accumulation of extreme-valued mutants, and that the Reflect operator has a plot that is symmetrical about $y = x$, and darker overall, indicating a more uniform distribution of mutants. (The values in each array were log-scaled to better reveal the structure in the distribution).

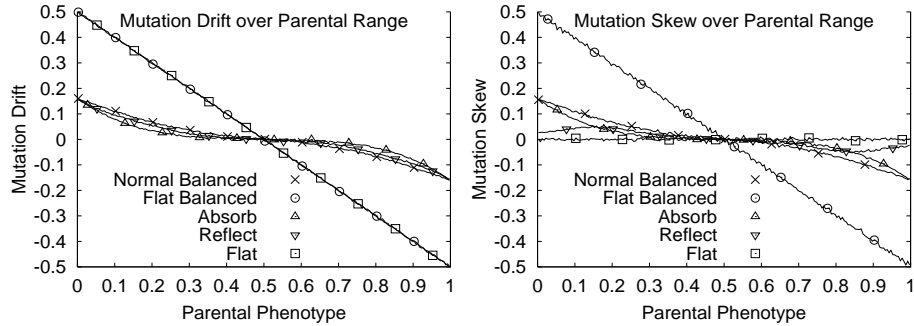


Fig. 4. Depicting the manner in which the strength and direction of mutation *drift* (left) and mutation *skew* (right) vary with parental value. Considerations of mutation drift cannot distinguish those operators biased towards extreme values (Flat-Balanced, Normal-Balanced, and Absorb) from those that are unbiased (Flat and Reflect). However, mutation skew varies inversely with parental value only for those operators favouring extreme values, tending to trap populations near trait boundaries.

boundary, the asymmetrical, skewed distribution of mutants ensures that this has little bearing on the location of the most densely populated mutant destination. In contrast, operators such as Reflect counter the tendency for populations to aggregate at the extreme values of a legal range, by decreasing the skewness of mutant distributions as parents approach trait boundaries.

From this discussion, it is tempting to conclude that certain mutation operators (perhaps Reflect) should be employed, rather than others (Absorb, or either of the Balanced operators). While there may be some truth in this, since certain mutation operators may be less biased than others, previous work [9] strongly suggests that mutation bias in one form or another is part and parcel of the evolutionary process. Rather than attempt to eliminate these biases from evolutionary algorithms, their presence should be anticipated and controlled for.

The analysis and visualisation presented in this section has revealed that our intuitions concerning the behaviour of mutation operators, and importantly the sources of their biases, are often not to be trusted. Measures of behaviour that at first sight appear sensible and instructive (e.g., mean mutant destination as a measure of mutation drift) can be misleading. The skewness of mutation distributions has been identified as a crucial aspect of a mutation operator's behaviour, and has been used to explain the biases of mutation operators that favour extreme-valued traits.

2 Understanding Mutation in Neutral Networks

Discrete-valued genotypes require mutation operators that are perhaps simpler in structure than the less orthodox operators analysed for real-valued genotypes in the previous section. However, like the operators discussed above, discrete (typically binary) genetic encodings also impose a structure upon the mutation

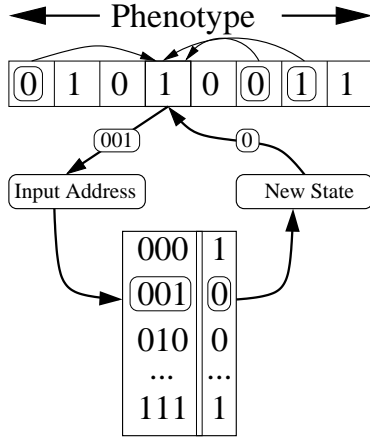


Fig. 5. Schematic representation of a *random Boolean network* as a developmental mapping. Each of the eight bits representing the organism's phenotype is associated with three neighbours, and a look-up table with eight entries. The state of each bit is updated by consulting the entry in its look-up table (only one look-up table is depicted) determined by the value of its three specified neighbours. All eight bits are updated once per cycle. After 20 cycles, the value of the eight phenotypic bits is taken as the organism's phenotype. A 144-bit genotype represents the initial state of the phenotype (8 bits), the entries in each phenotypic bit's look-up table (8×8), and each phenotypic bit's three neighbours ($8 \times 3 \times 3$). Adapted from [7].

space, allowing some transitions but preventing others, and as a result have just as much potential to bias evolutionary dynamics [9]. In this section we will employ some of the techniques introduced in section 1 to explore the biases inherent in a recently proposed massively redundant genetic encoding scheme [7, 8].

Recently, encouraged by theoretical [3] and empirical [6] work suggesting that neutrality in fitness landscapes may improve the ability of evolutionary algorithms to discover fit phenotypes, researchers have begun to explore the introduction of *redundancy* into discrete genetic encodings [7, 8]. Although redundancy (many genotypes mapping onto the same phenotype) increases the size of the search space, the correct kind of redundancy may populate this search space with neutral networks that increase phenotypic connectivity — enabling evolving populations to more easily explore a wider range of mutant phenotypes than they would be able to under a regular non-redundant encoding scheme. Potentially, neutral networks might increase this connectivity to such an extent that the overall performance of evolutionary algorithms is much improved.

However, is the mere *extent* of phenotypic connectivity the only important consideration here? In this section we will explore how the *distribution* of phenotypic connectivity may impact on the evolutionary dynamics of algorithms employing redundancy in this way.

One manner in which a massively redundant genotype-phenotype mapping has been implemented is through the use of a random Boolean network (RBN) [10], specified using 144 bits of genetic information, to generate an 8-bit phenotype (see Fig. 5). Under this scheme, it has been demonstrated [7, 8] that neutral networks connect each of the 256 possible phenotypes to the majority of the 255 possible mutant phenotypes (see Fig. 6 left). However, employing the visualisation technique introduced in the previous section, we can show that the frequency distribution over these neighbours is radically non-uniform (see Fig. 6 right). In fact, under the proposed encoding scheme, a large proportion ($\approx 32\%$)

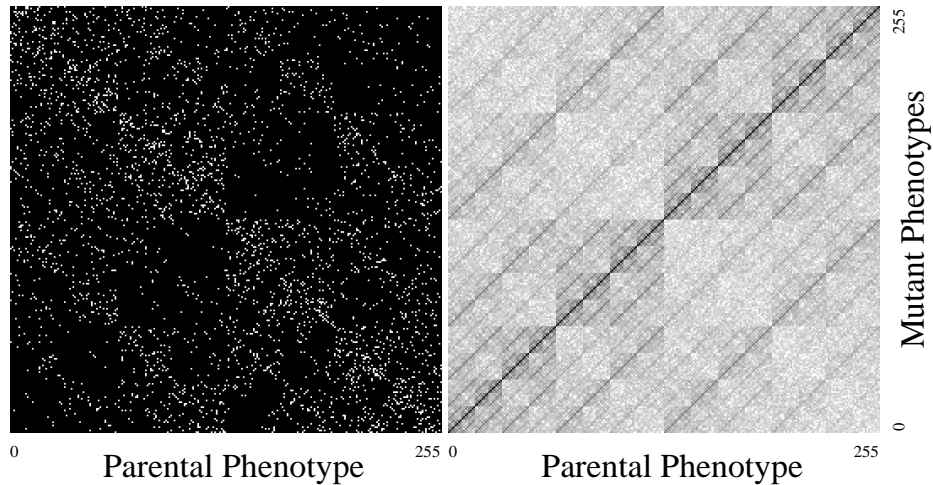


Fig. 6. Depicting the accessibility of mutant phenotypes from parental phenotypes for the RBN encoding scheme. A random 144-bit parental genotype is generated and its corresponding phenotype, x , determined. At random, one of the 144 bits is flipped and the phenotype, y , of the resulting mutant is determined. The cell (x, y) is incremented by one. This process is repeated for 10^6 random parents. *Left:* non-empty cells are depicted in black, empty cells in white. This depicts the *absolute* accessibility of mutant phenotypes from parental phenotypes. *Right:* empty cells are depicted in white, non-empty cells are depicted in grey levels with high-frequency cells shaded heavily. This depicts the *relative* accessibility of mutant phenotypes from parental phenotypes. Note the dark leading diagonal indicating the prevalence of neutral mutations. The uniformity of the left-hand plot indicates that almost all transitions are *possible*, the right-hand plot shows how misleading this conclusion is, since only a limited number of idiosyncratic transitions are at all *probable*. (The values in the right-hand image have been log-scaled to better reveal the structure in the frequency distribution.)

of non-neutral mutations are equivalent to flipping a single bit of the 8-bit phenotype, i.e., identical to those transitions that would be achieved under a regular non-redundant binary encoding (see Fig. 7). The remaining non-neutral mutations are distributed (non-uniformly) across the remaining 247 possible mutants. Whilst it is possible that the RBN scheme may enjoy advantages over a regular binary encoding scheme (perhaps it responds well to crossover, for instance), these results suggest that the cost of moving from a regular 8-bit representation to this redundant 144-bit scheme (an increase of search space size by a factor of 2^{136}) is perhaps not compensated for by the increased phenotypic connectivity that is achieved.

In order that neutral networks improve the performance of evolutionary algorithms, it must be the case that in addition to connecting each phenotype to many others they connect these phenotypes roughly equally, rather than favouring some transitions much more than others. What is important is not just

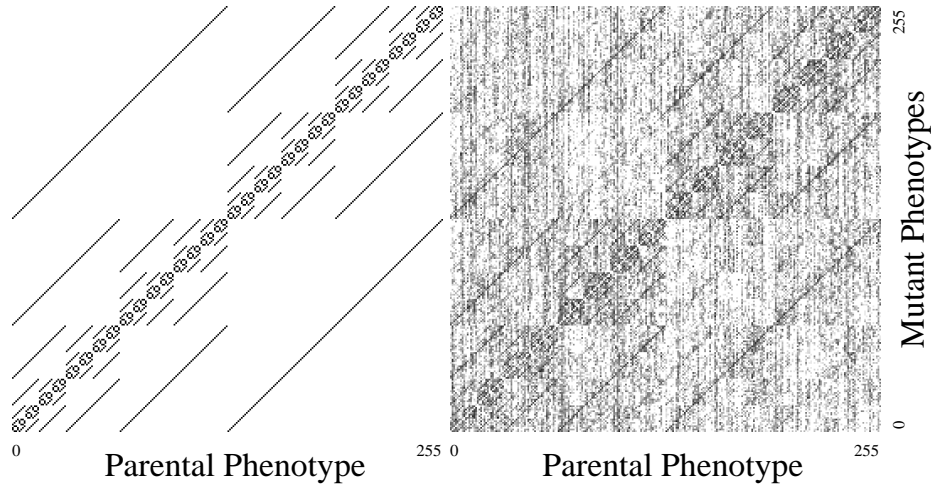


Fig. 7. *Left:* the distribution of the eight one-bit mutant neighbours of each of the 256 possible eight-bit phenotypes under a regular non-redundant *binary encoding* scheme. *Right:* the distribution of non-neutral neighbours collated for 256 *neutral networks* explored under the RBN encoding scheme. For each phenotypic value a random genotype encoding that value was generated. Each of the 144 one-bit mutants of this genotype were generated and their corresponding phenotypes determined. A neutral mutant was chosen at random and its 144 neighbours assessed, and so on. In this way 100 adjacent members of each neutral network were explored. Although each neutral network neighbours many phenotypic values, the similarity between the left- and right-hand plots indicates that a large proportion (32% in this case) of non-neutral mutation events are equivalent to flipping a single bit of the 8-bit phenotype. (The values in the right-hand image have been log-scaled to better reveal the structure in the frequency distribution.)

the *number* of phenotypes that neighbour a neutral network, but the *frequency distribution* over this neighbourhood, and whether it is significantly biased.

In this section we have discovered that an encoding scheme which promised a massive increase in phenotypic connectivity brings with it a hidden cost in terms of a radically biased distribution over this connectivity. Although many phenotypes are accessible from each neutral network, a population percolating through such a neutral network will tend to repeatedly explore only a few of the possible transitions, with many transitions occurring infrequently if at all.

How can we improve upon this situation? To achieve a more uniform distribution of phenotypic accessibility, we might explore increasing the connectivity of the random Boolean network which underlies the genotype-phenotype mapping, encouraging it to manifest chaotic behaviour which may be more sensitive to single genotypic bit flips. Moreover, if the RBN's behaviour is chaotic, running it for more clock cycles might also allow single genotypic bit flips to exert more influence on the end state of the RBN. However, this would result in a much longer genotype, and much larger neutral networks. Determining whether the

trade-off between increased phenotypic connectivity and increased search space favours redundant genotype-phenotype mappings of the kind explored here will require further theoretical and empirical enquiry.

3 Conclusions

Mutation bias is a little-explored and little-understood aspect of evolutionary algorithms. As the role of neutral mutation becomes increasingly significant in our theories of evolution and evolutionary computing, our ability to identify, visualise and understand these biases will itself become increasingly important. This paper has demonstrated that our intuitions regarding mutation bias and genotype-phenotype mappings are currently under-developed and has contributed analytic and visualisation techniques which may help us improve upon this state of affairs.

Acknowledgements

The paper benefited from discussion with Chris Goodyer, Chris Needham, Jason Noble, and Richard Watson, the comments of three anonymous referees, and the 40% advice of Robin Michaels, Paul Rolles, and Jack Rudd.

References

1. Forrest, S., Mitchell, M.: Towards a stronger building-block hypothesis: Effects of relative building-block fitness on GA performance. In Whitley, D., ed.: *Foundations of Genetic Algorithms 2*, Morgan Kaufmann, San Mateo, CA (1993) 109–126
2. Thornton, C.: The building block fallacy. *Complexity International* **4** (1997) URL: <http://www.csu.edu.au/ci/vol104/thornton/building.htm>.
3. Barnett, L.: Ruggedness and neutrality — the NKp family of fitness landscapes. In Adami, C., Belew, R., Kitano, H., Taylor, C., eds.: *Artificial Life VI*, MIT Press, Cambridge, MA (1998) 18–27
4. Kimura, M.: *The neutral theory of molecular evolution*. Cambridge University Press, Cambridge (1983)
5. Kimura, M.: *Population genetics, molecular evolution, and the neutral theory: Selected papers*. University of Chicago Press, Chicago (1994)
6. Harvey, I., Thompson, A.: Through the labyrinth evolution finds a way: A silicon ridge. In Higuchi, T., Iwata, M., Weixin, L., eds.: *First Int. Conf. Evolvable Systems*, Springer-Verlag (1997) 406–422
7. Shipman, R., Shackleton, M., Ebner, M., Watson, R.: Neutral search spaces for artificial evolution: a lesson from life. In Bedau, M.A., McCaskill, J.S., Packard, N.H., Rasmussen, S., eds.: *Artificial Life VII*, MIT Press, Cambridge, MA (2000) 162–169
8. Shipman, R., Shackleton, M., Harvey, I.: The use of neutral genotype-phenotype mappings for improved evolutionary search. *BT Technology Journal* **18** (2000) 103–111
9. Bullock, S.: Are artificial mutation biases unnatural? In Floreano, D., Nicoud, J.D., Mondada, F., eds.: *Fifth European Conference on Artificial Life*, Springer, Berlin (1999) 64–73
10. Kauffman, S.A.: *The origins of order: Self-organisation and selection in evolution*. Oxford University Press, New York (1993)