

# On the Utility of Redundant Encodings in Mutation-based Evolutionary Search

Joshua D. Knowles<sup>1</sup> and Richard A. Watson<sup>2</sup>

<sup>1</sup> IRIDIA - CP 194/6, Université Libre de Bruxelles, Ave. F. D. Roosevelt, 50, 1050 Brussels, Belgium. E-mail: [jknowles@ulb.ac.be](mailto:jknowles@ulb.ac.be)  
Web: <http://iridia.ulb.ac.be/~jknowles/>

<sup>2</sup> DEMO, Volen Center for Complex Systems, MS018, Brandeis University, Waltham, MA 02454, USA. E-mail: [richardw@cs.brandeis.edu](mailto:richardw@cs.brandeis.edu)

**Abstract.** A number of recent works in the evolutionary computation field have suggested that introducing large amounts of genetic redundancy may increase the evolvability of a population in an evolutionary algorithm. These works have variously claimed that the reliability of the search, the final fitness achieved, the ability to cope with changing environments, and the robustness to high mutation rates, may all be improved by employing this strategy. In this paper we dispute some of these claims, arguing that adding *random* redundancy cannot be generally useful for optimization purposes. By way of example we report on experiments where a proposed neutral encoding scheme (based on random Boolean networks) is compared to a direct encoding in two mutation-only EAs, at various mutation rates. Our findings show that with the appropriate choice of per-bit mutation rate, the evolvability of populations using the direct encoding is no less than with the redundant one.

## 1 Introduction

The neutral theory of evolution [7] states that a large fraction of mutations are fitness neutral. At the molecular level, many genotypes (RNA) effectively map to functionally equivalent phenotypes (proteins) and can form *neutral networks* [10] of genotypes separated by only one (or a few) point mutations. At a higher level also, genetic regulatory networks [2] may exhibit neutrality between the organization of the networks and their characteristic expression patterns. The dynamics of populations evolving on such neutral networks are different to those on adaptive landscapes [6]: they are characterized by long periods of ‘fitness stasis’ (where variation acts to increase the genetic diversity of the population, causing it to diffuse over the neutral network) punctuated by shorter periods of rapid fitness increase (and loss of diversity) due to the discovery of a ‘portal’ [14] to a network of higher fitness. These dynamics, it is suggested, mean that populations may be better able to evolve to higher fitness configurations and less likely to fall prey to the usual problems of entrapment at local optima.

Some researchers in the evolutionary computation (EC) community ([1], [4], [11], [12], [13], [15], [18]) have begun to address neutral theory in their work. Some (e.g. [1],[15]) suggest that certain real-world problems, when represented naturally, will exhibit a high degree of neutrality, and thus standard models of the dynamics of populations evolving on (non-neutral) adaptive landscapes may

not be relevant. Others ([4], [11], [12], [13]) suggest that introducing artificial redundancy into the representations of optimization problems may improve the evolvability of solutions in an evolutionary algorithm. In particular, the latter have variously claimed that the reliability of the search [12], the final fitness level achieved, the ability to cope with changing environments, and the robustness to high mutation rates [4], of EAs may all be improved by genetic redundancy.

The idea that a search process, that would otherwise be trapped at a local optimum, may be able to traverse, or diffuse over, a neutral network and escape to higher-fitness regions of the search space is intuitively appealing. However, from an engineering point of view, we must be careful in embracing neutrality as a generally advantageous property of a search space. There are, after all, other ways to escape from local optima or decrease the likelihood of becoming stuck at one—for example, using different variation operators, by employing diversity maintenance techniques, and not least, by choosing another encoding scheme. At present, it is poorly understood exactly what kind of neutrality is advantageous, and under what circumstances such an advantage will be shown—it is clear that not all types of genetic redundancy will be useful [5].

In [4], Ebner *et al.* describe several valuable conceptual features of redundant encodings and the neutral networks that such encodings might afford. For example, they suggest that the frequency distribution of phenotypes, the ‘reachability’ of phenotypes, the ‘innovation rate’ (the number of new phenotypes encountered as a function of neutral walk length), and the extent of the neutral networks are important characteristics of a redundant encoding and they provide corresponding statistical measures for these features. Their discussion and examinations using various artificially redundant encodings are valuable in understanding the possible nature of redundancy and neutrality in different genotype to phenotype mappings, and in assessing the possible utility of intrinsically redundant mappings in natural and engineering domains. However, we suggest that it is still not clear when an encoding incorporating neutrality should be preferred over a non-redundant encoding in cases where there is a choice. In this paper we wish to point out that some simple suggestions of when neutrality will be advantageous are inadequate, and incidentally, to contrast this with some other possibilities that may be genuinely advantageous but have not yet been demonstrated.

The remainder of this paper is organized as follows. In the next section we discuss why, in general terms, we would not expect the performance of EAs to be improved by the introduction of arbitrary neutral networks. We also highlight a number of results reported in the literature, which in our view do not represent convincing demonstrations of the advantages of the employed redundant encodings. In section 3 we describe a number of experiments that compare a redundant versus a direct encoding, regarding their performance in otherwise identical mutation-only EAs. Three distinct types of problem, H-IFF, MAX-SAT, and *NK* landscapes are used for the comparison. The results of these experiments are reported in section 4, and we review our findings in section 5.

## 2 Against Arbitrary Neutral Networks

Why should we believe that neutrality in a problem space is advantageous from a search or optimization point of view? One simple way to understand the effect of

neutrality is that it may afford better exploration for a search process because, for any method that follows fitness increases, a redundant encoding may have better connectivity to other (perhaps fitter) phenotypes and thus fewer local optima. All else being equal, even an arbitrary increase in connectivity—neutral networks connecting arbitrary phenotypes—certainly provides a potential advantage for optimization—[4], for example, provides illustrations of this advantage. However, in engineering domains we are not necessarily required to keep ‘all else equal’. If we have a choice about whether to include redundancy in an encoding scheme, the advantages of doing so must be weighed against the potential advantages of other alternatives. For example, we could provide ‘connectivity’ from any local optimum to any point in the search space by using a simple hill-climber that, in addition to considering neighbouring points in the search space, also occasionally considers random points in the search space. Alternatively, we might suppose that it would be preferable for neutral networks to increase connectivity to nearby (but not immediately adjacent) points in the search space rather than arbitrary points—but similarly, this could be provided by a simple mutation hill-climber using a larger mutation rate.

A useful question for focusing our reasoning is the following. Why, if at all, should a search method in a problem space with neutrality be able to find points of higher fitness faster or more reliably than it would with additional random exploration on a non-neutral space? For example, the expected time for a random walk to change  $M$  variables to a specific configuration is exponential in  $M$ , as is the expected time to jump from one point to another specific point directly under  $M$ -point mutation. In some scenarios, the exploration of a population on a neutral network might not be described accurately by a random walk, but favourable analyses are thus far absent. Of course, conceivably, a particular encoding, ‘natural’ or artificial, might provide some non-random bias in the connectivity of points that has an *a priori* bias appropriate for a particular problem domain. However we do not yet have any suggestions for what kind of neutrality is useful for what class of problems. Another possibility is that, in certain scenarios, the structure of neutral pathways might be somehow reorganized adaptively as evolution progresses. These possibilities, a useful *a priori* bias and/or an adaptive bias, in the connectivity of neutral networks could provide a genuine adaptive advantage in certain classes of problems—but they have not yet been demonstrated. In the meantime, it is difficult to see why any kind of arbitrary increase in connectivity provided by random neutral networks, whether inherent in the natural encoding of a problem space or added artificially, would provide an optimization advantage over much simpler search methods on non-redundant encodings.

To provide a preliminary illustration of this reasoning we examine, by way of example, a redundant encoding proposed by [4] and [13]. One particular encoding reported in this work, based on the operation of a random Boolean network (RBN), appears to be favourable with respect to the various statistical measures mentioned above. Furthermore, some experiments report results that seem to show that this mapping aids evolutionary search in multi-modal landscapes [4], [11]. However, it seems likely that the increased connectivity provided by the RBN encoding is arbitrarily structured, and that accordingly, the optimization advantage seen in this experimental work could be provided by a

simpler method of random exploration. In general, an increased mutation rate is a fairly unbiased means to increase random exploration (though not completely, see [3]). Therefore, it seems unlikely that an arbitrary artificial redundant encoding could, in general, be better than increased mutation. One possible disadvantage of a higher mutation rate is an inability to retain good solutions should they be discovered—however, the use of some elitism (retaining the best solution found so far) easily remedies this problem. Accordingly, the use of higher mutation rates, in hill-climbers and GAs with elitism (steady-state variety), is the alternative that we examine in some detail below.

### 3 Experimental Method

**3.1 Encodings** In the experiments that follow, we will compare a direct encoding and a redundant encoding of binary strings. In this context, we use the term “phenotype” to refer to the binary string of  $M$  bits that the fitness function evaluates. The term “genotype” is reserved for the binary string that is under the direct influence of the variation operators of the evolutionary algorithm. In the direct encoding, the genotype is identical to the phenotype, and no ‘mapping’ step is necessary. In contrast, the genotype of the redundant encoding is of length  $L > M$  bits, and represents a particular random Boolean network (RBN), and its initial state (described below). The RBN is then iterated for a fixed number of steps, and its final state represents the phenotype for which the genotype encodes.

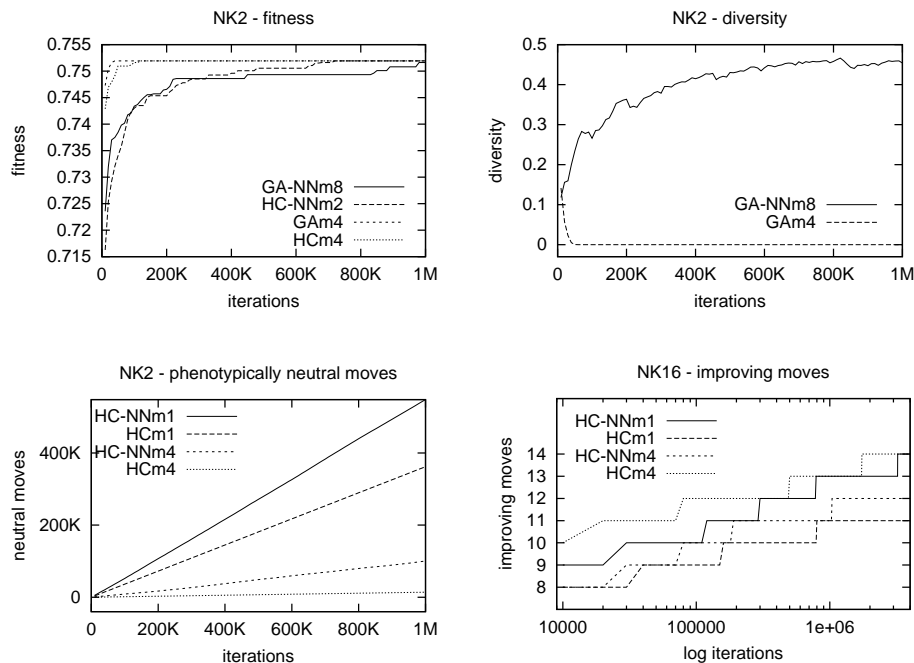
**3.2 Aims** The experiments are designed to evaluate the following aspects of performance of EAs using the random Boolean network encoding: (i) the distribution of fitness levels achieved on relatively long runs, for the size of problem; (ii) the rate of fitness increase; (iii) the robustness to the choice of mutation rate; and (iv) changes in (i) and (ii) with respect to any controllable problem features and problem size. We assess these aspects of performance by comparing identical EAs operating on a direct encoding and an RBN encoding of the binary phenotypes.

**3.3 The RBN encoding** An RBN is an iterated dynamical system described over a set of  $V$  Boolean variables. The state of each variable at time  $t$  is dependent on the state of  $W$  other variables at time  $t - 1$ . The new state is the result of a Boolean function over these variables. In our experiments (following the setup described in greater detail in [4]), the initial state of the variables, the wiring determining which variables are dependent on which others, and the Boolean state-update functions for each variable are all specified in the genotype and are subject to evolutionary adaptation. The value of  $W$  is fixed at 3. The number of variables  $V$  equals the phenotype length  $M$ , and the phenotype is determined by running the Boolean network for 20 iterations and reading off the final state of the variables. The problems we address are of length 32 and 64 bits. With the RBN encoding, the genotypes associated with these phenotypes are of length 768 and 1728 bits respectively.

**3.4 Mutation rates** In our experiments we aim to evaluate the performance of the RBN encoding in comparison to the use of a direct encoding, *given an appropriate choice of mutation rate*. The question whether the RBN encoding facilitates greater exploration (and thus evolvability) when mutation is restricted to a single point per-genotype mutation is regarded here as irrelevant, since this kind of mutation would never be used in practical EAs designed for use on multi-modal fitness landscapes. Thus, we choose to compare the performance of the two encodings in EAs using *per-bit mutation*, and we attempt to find an appropriate rate of mutation for both encodings. All of our experiments are carried out using per-bit (flip) mutation rates of  $1/L$ ,  $2/L$ ,  $4/L$  and  $8/L$ .

**3.5 Problems** The problems that we use are all unconstrained binary string problems, allowing straightforward use of the RBN encoding. None of the problems intrinsically possesses substantial neutral networks. Our interest here is in evaluating if the *introduction* of neutrality through a redundant genotype-phenotype mapping helps perform search in general, and not with the question of how to perform search on landscapes that have intrinsic neutrality, such as Royal stairways or  $NKp$  landscapes. Hence, for our problems we use  $NK$  landscapes [6], H-IFF, initially proposed in [17] and described more fully in [16], and MAX-SAT [8]. The  $NK$  landscapes allow us to observe if relative performance is affected by varying degrees of ruggedness or correlation. We use an  $NK$  landscape generator, [9], available as freeware, to generate landscapes of size  $N = 32$ . We generate one problem each at levels of  $K = 2, 4, 8, 16$ , and these are labelled NK2, NK4, NK8, and NK16, respectively, in the results section. The H-IFF problem is hill-climber-hard, in that the two global optima (the all 0s string and the all 1s string) are difficult to find by mutation, starting from a random string, because contiguous blocks of both 0s and 1s of size two or greater are rewarded by the hierarchical fitness function. Thus, strings tend to evolve toward local optima (strings comprising sizable blocks of both 0s and 1s) with a very large Hamming distance from the two global optima. This problem is very rugged and it has been shown that mutation cannot be guaranteed to succeed in time less than exponential in the problem size [16]. We use H-IFF instances of length 32 and 64 bits, labelled H-IFF32 and H-IFF64. MAX-SAT is a well-known NP-hard combinatorial optimization problem, and its real-world applications are many. Our two instances are un-weighted 3-SAT problems generated using an algorithm described in [8], for which freeware is available. The first has 32 variables and 200 clauses (SAT32-200) and the second has 64 variables and 280 clauses (SAT64-280), and both are satisfiable. We treat these 3-SAT instances as MAX-SAT optimization problems, with fitness calculated as the proportion of satisfied clauses.

**3.6 The EAs: Hill Climber and Genetic Algorithm** We evaluate the performance of the RBN in (i) a random mutation hill-climber (HC) and (ii) a steady-state mutation-only genetic algorithm (GA). HC uses a (1+1) population update: at each iteration, one random mutant of the current solution is generated (independently of previous steps) and the mutant replaces the current solution iff it is at least as fit. GA uses a population size of 100 and uses a steady-

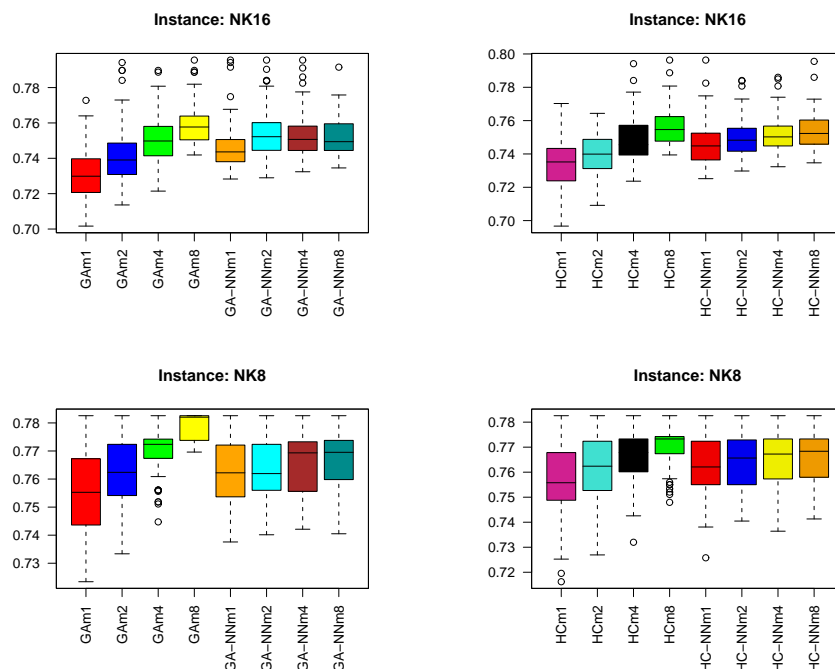


**Fig. 1.** Plots showing the evolution of (top left) fitness and (top right) diversity on NK2, and the cumulative number of (bottom left) phenotypically neutral moves. For NK16 the cumulative number of fitness-improving moves is shown (bottom right) against log iterations. All values are the mean of 30 independent runs. See text for discussion of these plots.

state population update. At each iteration, a 2-tournament is used for selection and replacement. In this, two members of the population are chosen at random without replacement. The fitter of the two becomes the parent. The parent is then replaced in the population, copied and a mutant of the copy is generated and evaluated. The mutant then replaces the less fit of the two chromosomes from the tournament, iff it is at least as fit. The two EAs using the random Boolean network encoding are identical to HC and GA except for the encoding, and are labelled HC-NN and GA-NN, respectively. We further use the shorthand GA-NNm1 to denote GA-NN used with a mutation rate of  $1/L$ , and similarly for the other EAs and mutation rates. With all the algorithms we run until the global optimum has been found (if known) or 4 million iterations have been performed.

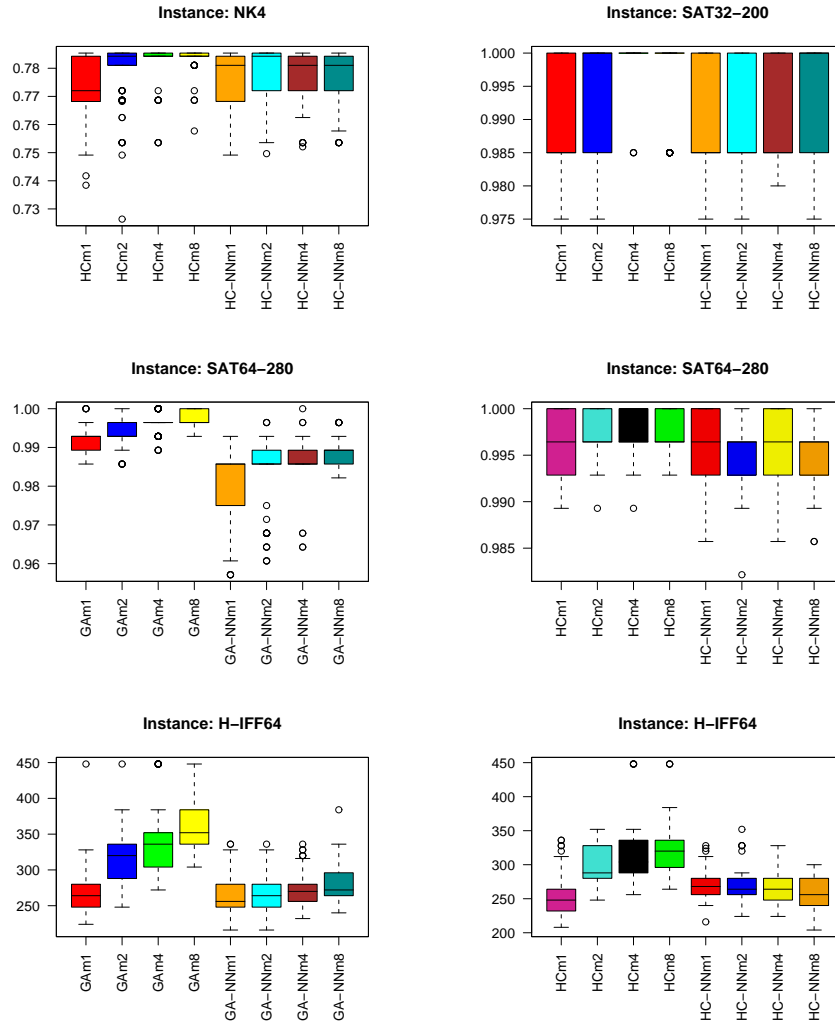
## 4 Simulation Results

Figure 1 presents four plots which give an insight into the difference between algorithm runs with and without the RBN encoding. On NK2, all algorithms reached the global optimum within the allotted number of iterations so that



**Fig. 2.** Results from 100 independent runs on the two more rugged  $NK$  landscapes.

there is no statistical difference in the final fitness values achieved. However, as can be seen from the plot top left, maximum fitness is achieved much faster using the direct encoding than with redundancy, in both GA and HC algorithms (the best mutation rates for each from those rates tested were selected). The plot at the top right shows the evolution of genetic diversity, defined as the sample average probability that two randomly selected individuals differ at a randomly chosen locus, as in [4]. In GA-NN's population, diversity grows over time until it stabilises near the maximum value of 0.5, indicating diffusion through the neutral network of the global optimum. In contrast, GA has lost all genetic diversity. However, the greater diversity in GA-NN's population did not increase its ability to find 'shortcuts' towards the optimum; it got there *more slowly*. In the plot bottom left, the number of phenotypically neutral mutations are plotted against iterations. The massive redundancy in the RBN encoding means the number of neutral mutations is higher than in the direct encoding, at each mutation rate. However, in the plot bottom right (for the NK16 problem), where we can see the cumulative number of mutations that lead to increases in fitness, the characteristics of the HCs employing the RBN encoding do not appear to benefit from the extra neutrality. In fact, we observe that the dynamics look very similar in the four algorithms: a similar number of fitness improvements is seen and the waiting times between them increase exponentially. The plots on these problems are typical for all of the problems considered here.



**Fig. 3.** Results from 100 independent runs each on selected problems.

The final fitness values that were recorded seem to indicate that the random Boolean network encoding does not significantly benefit either GA or HC on any problem, when an appropriate mutation rate is selected. Figures 2 and 3 present these results in the form of boxplots<sup>1</sup>. On NK16 (Figure 2), it appears that GAM4 achieves the highest median value and, together with GAM2, it has

<sup>1</sup> Box plots graphically represent the distribution of a random variable. The centre line represents the median, the length of the shaded box indicates the interquartile range (IQR) and the whiskers extend out to the largest and smallest observations within 1.5 times the IQR. Outliers are shown as small circles.

the best distribution of final values. With the hill-climbers, there appears to be a marginal advantage to the redundant encoding. We also observe that HC-NN is more robust with respect to different mutation rates than is HC on this problem. On NK8, GAm8 performs best. Once again there is more robustness to mutation-rate choice in GA-NN, but the spread in final values is also greater. There is little to choose between the performance of the best HC and HC-NN on this problem although one might argue that the standard hill-climber with the highest mutation rate (HCm8) seems to be best. On NK4, (Figure 3, top left), HC appears to perform better than HC-NN. On the MAX-SAT problems (also Figure 3) there is a clear difference in favour of the non-redundant encoding on all plots shown<sup>2</sup>. The results on the H-IFF problem shown in Figure 3 are for experiments with 64 bit H-IFF as opposed to the 16 bits used in [4]. Our results demonstrate that on this size of problem the redundant encoding performs worse than the non-redundant encoding, possibly indicating that performance does not scale with the use of redundancy. Our results on H-IFF32 (not shown<sup>2</sup>), are also consistent with this. Generalizing from all the plots, there does not seem to be a difference in the spread of final fitness values achieved, using the standard and redundant encodings, indicating that the RBN does not appear to increase the reliability of the search.

## 5 Conclusion

In this paper we have argued against the theory that adding *random* genetic redundancy to the encoding of a problem will generally increase the search performance of a correctly configured EA. The evidence in the literature given to support the use of redundant encodings, we argued, only shows the increased connectivity of the redundant space, and is not a sound demonstration of the utility of the technique. In our experiments we found that when HCs and GAs are provided with massively redundant genotypes based on a random Boolean network, their performance does not improve, in terms of the final fitness achieved on relatively long runs. In addition, the RBN encoding does not appear to affect the number of improving moves to reach a particular fitness value, or the distribution in waiting times between them. Neither was the spread of final fitness values achieved tighter for the RBN encoding (i.e. increased reliability was not observed). On the positive side, the performance of the RBN-encoded algorithms did seem to be more robust to the choice of per-bit mutation rate on most problems. The findings reported here are of course very limited, and other experiments may give different results. However, the consistent finding across the problems tackled was that with an appropriate per-bit mutation rate, escape from local optima and the improvement of fitness occur just as well in a direct encoding as with the redundancy provided by the RBN encoding.

**Acknowledgments** JK gratefully acknowledges the support of a European Commission Marie Curie Fellowship, contract number: HPMF-CT-2000-00992.

<sup>2</sup> GA results for SAT32-200 and NK4 are not shown because the global optimum is found reliably by both GA and GA-NN at some of the mutation rates. These plots, and others, can be obtained from the first author on request.

## References

1. L. Barnett. Netcrawling—optimal evolutionary search with neutral networks. In *Proceedings of the 2001 Congress on Evolutionary Computation (CEC2001)*. IEEE Service Centre, Piscataway NJ, 2001.
2. S. Bornholdt and K. Sneppen. Robustness as an evolutionary principle. *Proceedings of the Royal Society of London Series B-Biological Sciences*, 267:2281–2286, 2000.
3. S. Bullock. Smooth operator? Understanding and visualising mutation bias. In J. Kelemen and P. Sosik, editors, *Sixth European Conference on Artificial Life*, pages 602–612, 2001.
4. M. Ebner, M. Shackleton, and R. Shipman. How neutral networks influence evolvability. *Complexity*, 2002. (In press).
5. I. Harvey and A. Thompson. Through the labyrinth evolution finds a way: A silicon ridge. In *Proceedings of ICES96*, pages 406–422. Springer-Verlag, 1996.
6. S. A. Kauffman. Adaptation on rugged fitness landscapes. In D. Stein, editor, *Lectures in the Sciences of Complexity*, pages 527–618. Redwood City: Addison-Wesley, 1989. SFI Studies in the Sciences of Complexity, Lecture Volume I.
7. M. Kimura. *The Neutral Theory of Molecular Evolution*. Cambridge University Press, 1983.
8. M. Motoki and R. Uehara. Unique solution instance generation for the 3-satisfiability (3-SAT) problem. Technical Report C-129, Dept. of Math and Comp. Sciences, Tokyo Institute of Technology, 1999.
9. M. A. Potter. An *NK*-landscape generator, 1997. Presented at the Workshop on Test Problem Generators for Evolutionary Algorithms, at the 7th ICGA.
10. C. M. Reidys and P. F. Stadler. Neutrality in fitness landscapes. *Appl. Math. Comput.*, 117:321–350, 1998.
11. M. Shackleton, R. Shipman, and M. Ebner. An investigation of redundant genotype-phenotype mappings and their role in evolutionary search. In *Proceedings of the 2000 Congress on Evolutionary Computation*, pages 493–500, Piscataway, NJ, 2000. IEEE Service Center.
12. R. Shipman. Genetic redundancy: Desirable or problematic for evolutionary adaptation? In *Proceedings of the 4th International Conference on Artificial Neural Networks and Genetic Algorithms (ICANNGA)*. Springer-Verlag, 1999.
13. R. Shipman, M. Shackleton, M. Ebner, and R. Watson. Neutral search spaces for artificial evolution: a lesson from life. In *Artificial Life VII: Proceedings of the Seventh International Conference*. MIT Press, 2000.
14. E. van Nimwegen and J. P. Crutchfield. Metastable evolutionary dynamics: Crossing fitness barriers or escaping via neutral paths? *Bulletin of Mathematical Biology*, 62(5):799–848, Sep 2000.
15. V. K. Vassilev and J. F. Miller. The advantages of landscape neutrality in digital circuit evolution. In *ICES*, pages 252–263, 2000.
16. R. A. Watson. Analysis of recombinative algorithms on a non-separable building-block problem. In *Foundations of Genetic Algorithms VI*, pages 69–89. Morgan Kaufmann, 2001.
17. R. A. Watson, G. S. Hornby, and J. B. Pollack. Modeling building-block interdependency. In *Parallel Problem Solving from Nature - PPSN V*, pages 97–106. Springer-Verlag, 1998.
18. T. Yu and J. Miller. Neutrality and the evolvability of Boolean function landscape. In *Proceedings of the 4th European Conference on Genetic Programming (EuroGP)*, volume LNCS 2038, page 204 ff. Springer-Verlag, 2001.