

Genetic Redundancy in Evolving Populations of Simulated Robots

Abstract A number of authors have argued that redundancy in biological organisms contributes to their evolvability. We investigate this hypothesis via the experimental manipulation of genetic redundancy in evolving populations of simulated robots controlled by artificial neural networks. A genetic algorithm is used to simulate the evolution of robots with the ability to perform a previously studied task. Redundancy is measured using systematic lesioning. In our experiments, populations of robots with larger genotypes achieve systematically higher fitness than populations whose genotypes are smaller. It is shown that, in principle, robots with smaller genotypes have enough computational power to achieve optimal fitness. Populations with larger (redundant) genotypes appear, however, to be more evolvable and display significantly higher diversity. It is argued that this enhanced evolvability is a direct effect of genetic redundancy, which allows populations of redundant robots to explore *neutral networks* spanning large areas of genotype space. We conjecture that, where cost considerations allow, redundancy in functional or potentially functional components of the genome may make a valuable contribution to evolution in artificial and perhaps in biological systems. The methods described in the article provide a practical way of testing this hypothesis for the artificial case.

Orazio Miglino

Dipartimento di Psicologia
Seconda Università di Napoli
Via Vivaldi, 43
81100 Caserta
Italy

Richard Walker

Consiglio Nazionale
delle Ricerche
Istituto di Psicologia
Viale Marx, 15
00137 Rome
Italy

Keywords

redundancy, neural networks,
genetic algorithms

1 Introduction

In a redundant system individual components can be modified or damaged without affecting functionality. In this sense, biological organisms are redundant at many different levels. Genetic redundancy, for example, involves not only junk DNA (which is redundant by definition) but also functional or potentially functional sections of the genome; at higher levels of organization different amino acid sequences can map to similar or identical protein foldings; at a still higher level many aspects of behavior and cognition are unaffected by the details of neural micro-architecture.

In many instances redundancy in biological organisms may reflect nothing more than the inefficiency of biological evolution in finding effective solutions to complex adaptive problems. In other cases redundancy can protect organisms against the consequences of accidental damage and, in the case of the genome, against deleterious mutations [16]. It has also been argued, however, that certain kinds of redundancy can play a more constructive role, contributing actively to the adaptability of the organism and even to the evolution of new functionality. In this article we report computer simulations designed to test this hypothesis via the manipulation of genetic redundancy in evolving populations of simulated robots controlled by artificial neural networks.

2 Redundancy and Evolvability in Natural and Artificial Systems

The evolvability of a population is its ability to generate adaptive variation [1] and thus to respond to the changing requirements of its environment. Evolvability is known to depend on a broad range of different factors. These include the choice between sexual and asexual reproduction; the characteristics of the genotype-to-phenotype mapping and related constraints [17]; mutation rates; and the availability and effectiveness of mechanisms for DNA repair. A number of authors have argued that one of the factors contributing to evolvability may be *genetic redundancy*, in particular in the form of duplicate genes and redundant mappings from RNA to secondary structure.

Ohno [14] has suggested that the creation of redundancy via gene duplication allows organisms to conserve their original genetic material while releasing the duplicate copy from the constraints of natural selection. This, it is argued, allows the genome to explore new adaptive opportunities and, on rare occasions, to acquire novel functionality. Recent empirical work has suggested that the proportion of redundant material maintained may in fact be *higher* than Ohno's theory would predict [7]. While an examination of these observations and supporting theoretical models goes beyond the scope of this article, there is a strong consensus among researchers that natural selection favors the retention of redundant genetic material resulting from gene duplication and that this material can facilitate the evolution of new functionality (or the partition and specialization of existing functions). This result is supported by the work of Calabretta et al. [2], who have used ALife simulations to compare the evolutionary performance of systems allowing gene duplication with systems in which this possibility is not allowed. The results support the hypothesis that systems with redundant, duplicate genes achieve higher fitness than systems without them.

Recent theoretical and modeling work has suggested a second, independent mechanism whereby genetic redundancy may contribute to adaptive evolution. Computer modeling of the mapping of RNA sequences to secondary structure has shown that a population of homogeneous phenotypes may in fact contain many different genotypes [8]. The organisms constituting such a population are, by definition, genetically redundant. For levels of redundancy beyond a critical threshold, RNA sequences coding for the same secondary structure become linked in *neutral networks*¹ spanning large areas of sequence space. Many adaptive opportunities lie within a single point mutation of the area spanned by the network. Huynen, Stadler, and Fontana [9] have argued that the diffusion of populations across this kind of large neutral network allows them to search for these rare opportunities via a highly efficient parallel search strategy. If this is correct, redundant RNA coding of secondary structure is likely to play an important role in adaptive evolution. Extending the argument, it seems intuitively likely that other forms of redundant coding (e.g. from amino acid sequences to protein structure, or from neural structure to cognitive function) may play a similar role defining higher level neutral networks that make their own contribution to evolvability. This suggestion is supported by anecdotal evidence from experiments in artificial evolution. Cliff and Miller, for example, report investigations into coevolution of pursuit and evasion behavior in which the best results were obtained where the neural networks controlling artificial organisms contained redundant units [4]. Similarly, Thompson et al. [15], in their work on evolvable hardware, discovered that efficient evolution of electronic circuitry required the use of gate arrays that were several times larger than the circuits that finally evolved.

¹ For readers unfamiliar with neutral networks, a brief explanation is in order. Consider a sequence coding a secondary structure, where all such structures coded by nodes on the network have the same fitness and where links between nodes represent biologically plausible mutations transforming one sequence into another. Given that the transition from one node to another has no effect on fitness, these mutations are said to be neutral, and the network is referred to as a neutral network.

3 Experimental Design

One of the reasons for the shortage of systematic investigations into the relationship between redundancy and evolvability is the practical and theoretical challenge of measuring redundancy in complex systems. To measure the redundancy of a system it is first necessary to identify which components of the system are functional and which can be removed or damaged without affecting its fitness. This is often a non-trivial task, as shown, for example, by the debate around so-called junk DNA. In natural systems, tightly focused lesioning of specific components is technically difficult and expensive, and even where the use of such techniques is feasible, the effects of lesions may not be immediately apparent; additional problems may be caused by the difficulty of untangling the benefits of redundancy from the metabolic costs.

The use of ALife techniques can help to circumvent the difficulties mentioned above. In the experiments reported in this article we use a *genetic algorithm* [12] to evolve populations of simulated robots with the ability to perform a previously studied landmark navigation task [11]. Robots are controlled by a feed-forward neural network. By design the simulations provide robots with strictly limited opportunities for the generation of adaptive variation. Selection is based on a deterministic truncation procedure; reproduction is asexual; there is no learning or development during the lifetime of the organism; the environment is static and homogeneous, providing no opportunity for niche selection [18]. As a result, genetic variation depends exclusively on random mutations in connection strengths. Robot physics is defined in such a way as to exclude any relationship between the size of the genome and the physical performance of the robot. Genetic redundancy is, in other words, cost-free.

In the artificial setting just described, we compare the performance of populations of organisms with different sized genomes. Systematic lesioning is used to measure the redundancy of individual organisms. Using the results of this analysis, we investigate the relationships among genome size, redundancy, population diversity, and fitness.

3.1 Environment and Task Definition

The environment used in our experiments (see Figure 1) consisted of a 21×21 grid of cells with a central 5×5 *target area* and two landmarks placed in diagonally opposite corners. The genetic algorithm used in the simulations evolves populations of robots with the ability to start from any point on the edge of the grid, to move as rapidly as possible to the target area, and to spend as much time there as possible. By design the only source of information available to the robot is information from the input neurons (see below), which changes continually as the robot moves in the environment. To achieve optimal performance the robot has to use this information to identify the optimal route to the target area and to detect when it has arrived there. Theoretical analysis suggests that successful accomplishment of the task requires the computation of trigonometric functions. This makes it likely that there exists no linear mapping of input to output capable of implementing an effective strategy. This conclusion is supported by experimental work with so-called *growing networks*, reported elsewhere [11], which shows that networks, when allowed to evolve an optimal architecture for this task, consistently select architectures with a significant number of hidden units. The selection of this kind of architecture is strong evidence that the task is linearly non-separable. Taken together, the theoretical and empirical evidence suggests that the task is harder, and the fitness landscape more rugged, than might at first appear.

3.2 Fitness Function

The fitness of individual robots was evaluated by placing the robot on a cell at the edge of the grid, facing in a direction perpendicular to the edge. It was then allowed

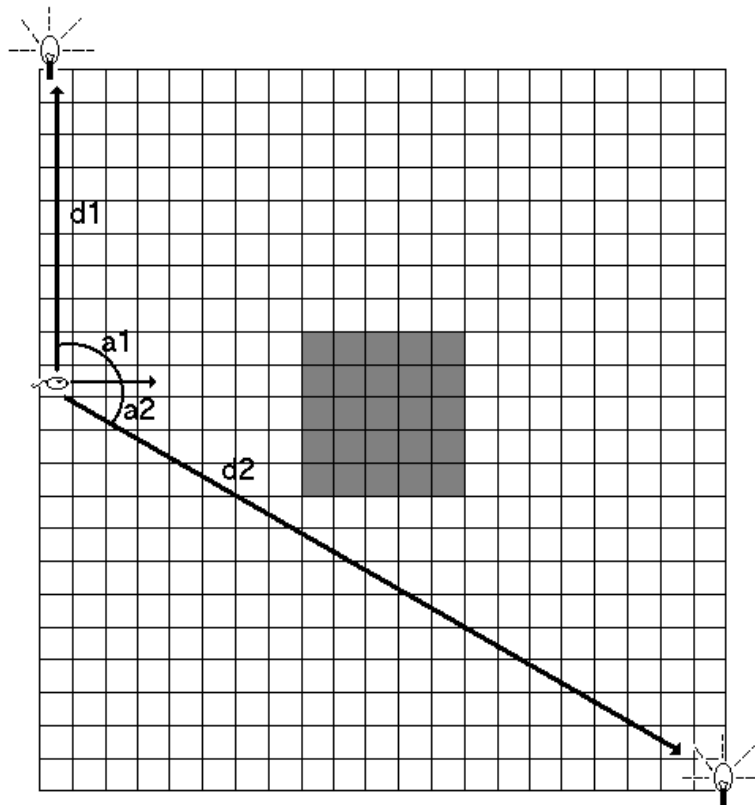


Figure 1. The test environment. D_1 and A_1 represent the robot's distance and angle with respect to landmark 1; D_2 and A_2 , those with respect to landmark 2.

to move for 20 cycles of computation, and the number of cycles spent in the target area was counted. This operation was repeated starting from each of the 80 edge cells on the board. Corner cells were tested twice, with the organism oriented once in a vertical and once in a horizontal direction. Fitness was defined as the proportion of total cycles spent in the target area. The data reported in the paper are normalized to a scale of 0 to 1, with a score of 1 representing optimal performance.

3.3 The Robots

Each of the simulations involved a population of 100 robots simulated in software. Each robot was controlled by a fully connected, two layer perceptron with four input neurons, two output cells, and a variable number of hidden units (see Figure 2). The input neurons signaled, for each landmark, the angle of the robot to the landmark and its distance from it. The binary output cells coded four possible motor actions: turn 90° left, move forward, turn 90° right, stay still. Note that this coding does not allow diagonal movement. Initial connection strengths were uniformly distributed random numbers in the range -1 to 1 . A set of chromosomes (the robot's genotype) coded the strengths of connections to and from specific hidden units (inter-neurons connecting sensor input to motor output). The number of chromosomes directly determined the number of hidden units. There was thus a direct relationship between the size of the genotype and that of the neural network controlling the robot. By design there was no

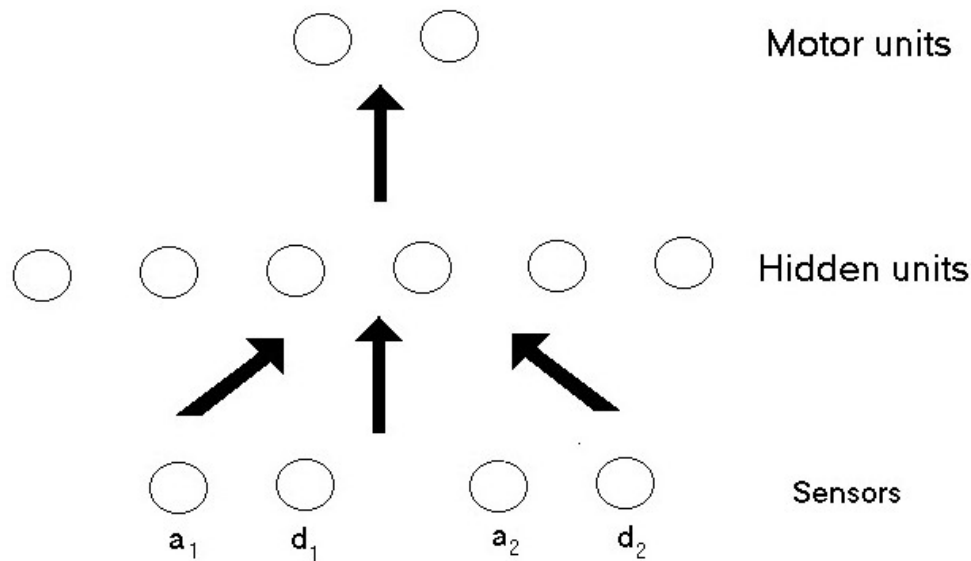


Figure 2. Neural architecture of simulated robots.

relationship between the size of the neural network and the physical performance of the robot: larger control networks incurred no extra cost in performance or fitness.

3.4 Genetic Algorithm

The evolutionary process simulated in our experiments was based on a genetic algorithm that was iterated for 600 generations. At the beginning of every generation every robot in the population was tested for fitness using the procedure described above. Reproduction was asexual. Each of the fittest 20 robots was made to produce five clones; other robots produced no offspring. During reproduction 20% of neural connections were mutated by summing the previous connection strength and a random number, uniformly distributed between -1 and 1 . It is worthy of note that the rate of mutation used in the experiments was a high one. In informal tuning experiments, conducted prior to the work reported in this paper, it was found that that simulations using this high rate of mutation evolved optimal behavior correctly, and that the time required to achieve optimality was significantly shorter than for simulations with lower mutation rates.

3.5 Measuring the Effects of Genotype Size

In order to investigate the effects of genome size, we compared the results of simulations in a population of robots with six chromosomes coding for six hidden units (6H) with results for a second population whose organisms had 10 chromosomes coding for 10 hidden units (10H). To ensure statistical robustness, each simulation was repeated 10 times using a different random number seed for each trial.

3.6 Measurement of Redundancy

The redundancy of individual robots was measured by systematically deleting all possible combinations of chromosomes, reassessing the fitness of the organism after each lesion, and recording the size of the smallest subset of functional chromosomes (SSFC) capable of achieving the same fitness as the intact version. In view of the computa-

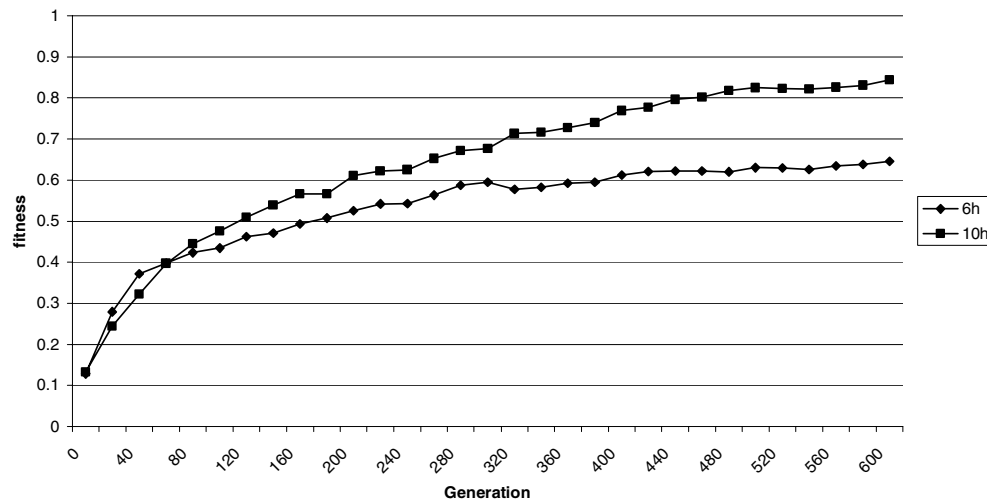


Figure 3. Evolution of mean fitness in populations 6H and 10H.

tional expense of this procedure, the redundancy was computed exclusively for robots selected for reproduction, sampled at 20-generation intervals.

3.7 Measurement of Population Diversity

In addition to identifying the size of the SSFC, we computed, for each genotype present in the population, the number of *neutral* lesions, that is the number of lesions having no effect on fitness. This number was used as the robot's *signature*. Robots with different signatures were said to belong to different *varieties*. Population diversity was measured by counting the number of varieties present in the population.

4 Results

4.1 Evolution of Fitness

In both 6H and 10H, robots improved their performance asymptotically during the evolutionary process. As can be seen from Figure 3, the mean fitness of the two populations was about the same during the early stages of evolution (from generation 0 to generation 60). From this stage on, robots in 10H consistently outperformed those in 6H ($F = 12,072.1$, d.f. 1/12,398, $p < 0.00001$). By generation 600, at the end of the evolutionary process, the two populations showed significantly different fitness distributions (see Figure 4). Although individual robots belonging to 6H occasionally achieved high levels of fitness, their genotypes never reached fixation. Viewed as a whole, the population generated an excess of organisms with fitness <0.7 and a deficit in organisms with fitness >0.7 . In 10H, on the other hand, optimally performing robots were predominant. At the end of the evolutionary process 10H displayed a deficit in organisms with fitness <0.7 and an excess of organisms with fitness higher than this value.

4.2 Behavioral Strategies

Both in 6H and in 10H, robot behavior evolved in four distinct stages (see Figure 5):

1. The robot moved forward in a straight line from its starting position and did not stop until the end of the experiment; this implies that from most starting positions the robot missed the target area altogether.

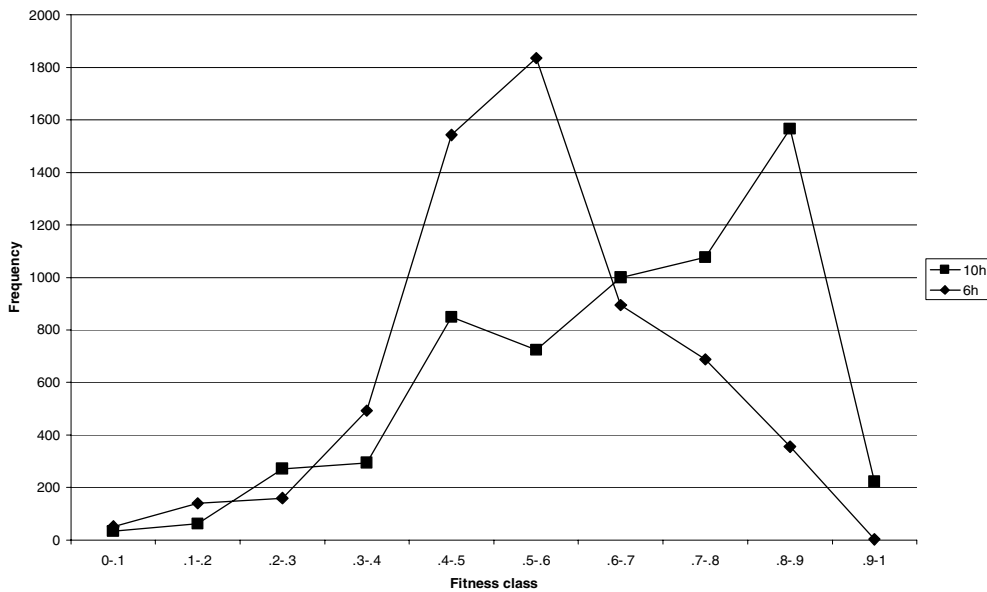


Figure 4. Distribution of fitness in populations 6H and 10H.

2. The robot displayed diversified behavior according to its starting position. From some starting positions the robot moved forward across half the grid and then stopped; in others it failed to stop; when the robot stopped within the target area or crossed the area, it received a fitness premium; in other cases the behavior made no contribution to its fitness.
3. The robot again displayed diversified behavior. The robot moved forward across half the grid. When it was on one side of the target area, it stopped; when it was on the other side, it rotated 90° in the direction of the target area, moved forward, and came to rest within the area; when the robot succeeded in rotating, it improved its fitness with respect to robots in stage 2.
4. (Optimal behavior.) The robot moved forward across half the grid and then, according to its position, either stopped (if it was within the target area) or rotated 90° in the direction of the target area, moved forward, and came to rest within the area.

4.3 Fitness, Size of SSFC, and Population Diversity

Both in 6H and in 10H, fitness was positively correlated with size of SSFC ($r_{6H} = 0.733$; $r_{10H} = 0.711$). In the early stages of the evolutionary process robots had small SSFC and poor performance. As performance improved, the size of the SSFC grew. If size of SSFC is taken as an index of complexity, it can be said that evolution led to an increase in complexity, which in turn allowed improved performance on the assigned task. The observed increase in computational complexity corresponds to the increasingly complex behavioral strategies described earlier.

The positive correlation between the number of chromosomes and the mean population fitness at the end of the evolutionary process lends itself to a number of possible interpretations. The simplest of these is that the algorithmic complexity [3] of the experimental task is such as to require a larger number of hidden units than were available

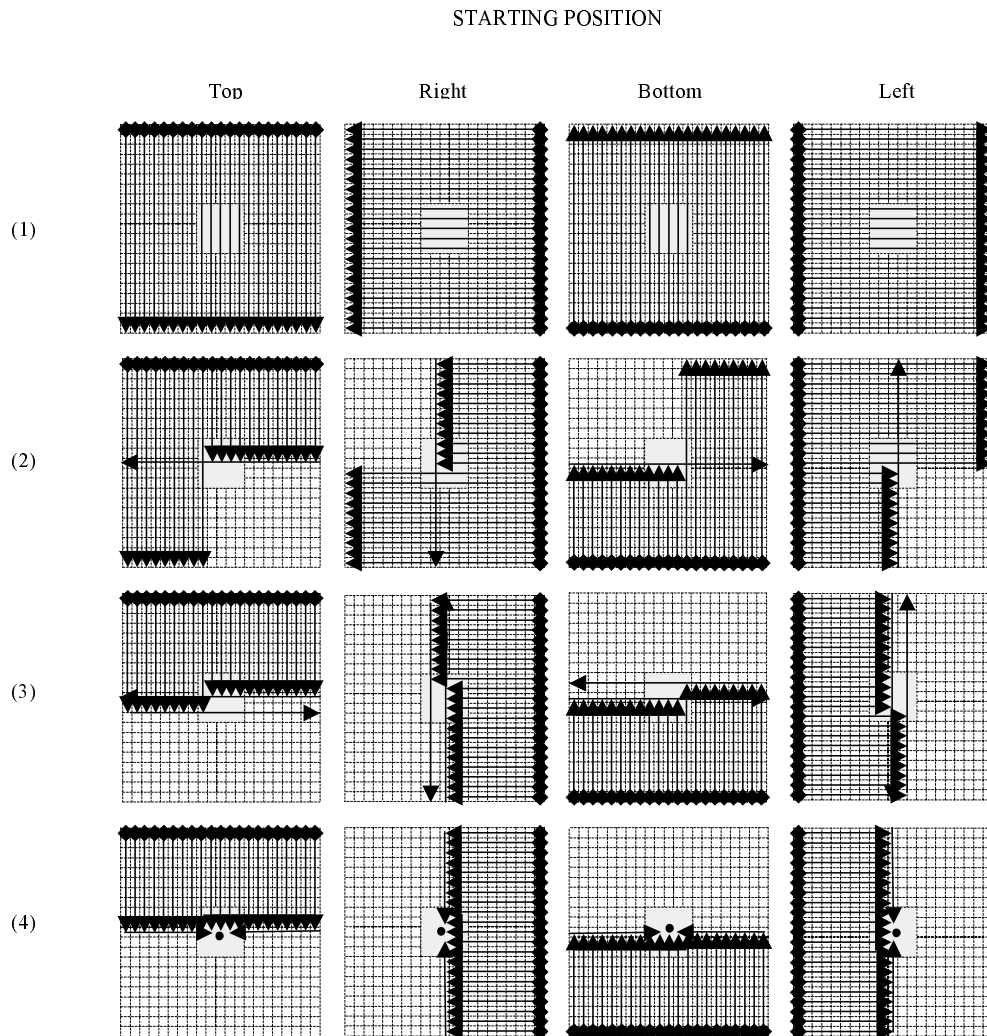


Figure 5. Behavioral strategies adopted by robots at different stages in the evolutionary process.

to robots with smaller genomes. In order to exclude this hypothesis, we divided our complete population of robots into 10 distinct fitness classes, identifying for each class the smallest SSFC in the dataset. In this way we found an upper bound for the size of the smallest network capable of achieving this level of fitness. As can be seen from Table 1, even in the highest fitness class the size of the smallest SSFC (five hidden units) was smaller than the size of the smallest organisms used in the simulations (six hidden units). This implies that a robot controlled by a neural network with six hidden units has sufficient computational resources to achieve optimal performance on the assigned task and that any chromosomes additional to those required to code for this number of hidden units are redundant with respect to the computational requirements of the task.

The frequency distribution for SSFC (see Figure 6) shows that only 4.2% of organisms in 6H developed an SSFC of size 6 (the maximum theoretical value) and only 0.02% of organisms in 10H developed an SSFC of size 10. This is strong evidence of

Table 1. Minimum size of SSFC in organisms belonging to different fitness classes (aggregate results for 6H and 10H).

Fitness class	Min. SSFC
0.00–0.09	1
0.10–0.19	1
0.20–0.29	2
0.30–0.39	3
0.40–0.49	3
0.50–0.59	3
0.60–0.69	4
0.70–0.79	4
0.80–0.89	5
0.90–1.00	5

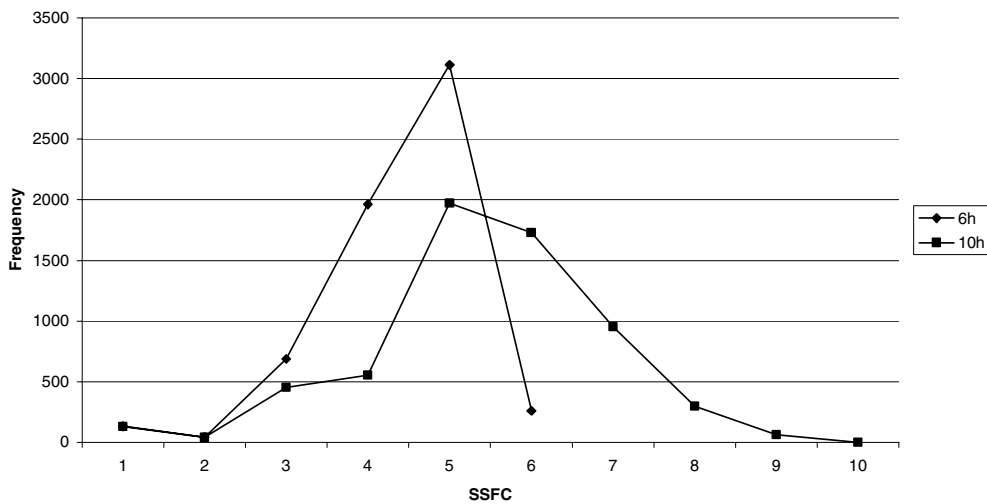


Figure 6. Distribution of SSFC size in populations 6H and 10H.

redundancy. In 10H, 78.7% of all robots in the population developed SSFC with six or fewer chromosomes. Comparing the performance of these robots with organisms in 6H—whose SSFCs were necessarily of size ≤ 6 —it was found that the robots belonging to 10H achieved systematically higher performance ($F = 11,707$, d.f.=1/11,083, $p < 0.00001$). The individual organism that recorded the highest performance, during the simulations, was a member of 10H, with an SSFC size of 6. This again supports the idea that successful organisms were, in general, highly redundant.

Analysis of the number of *varieties* within the population (see Figure 7) showed that, in every generation, 10H produced higher population diversity than 6H (ANOVA, d.f. 1/198, $p < 0.00001$ in all cases). In the first 100 generations 10H always contained at least eight different varieties; in 6H, on the other hand, the average number fell rapidly from 12.3 in generation 0 to 2.7 in generation 100. By generation 600 6H had almost completely lost its diversity (mean number of varieties = 1.1), whereas 10H maintained a mean diversity of 3.0. In other words, 10H could rely on a greater pool of diversity at the end of the evolutionary process than was available to 6H at generation 100.

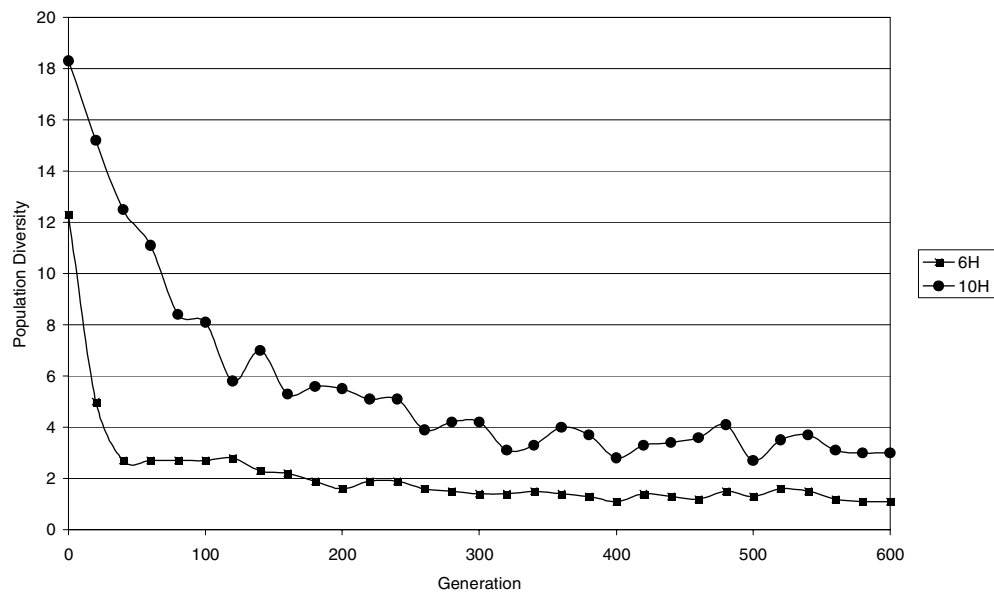


Figure 7. Mean population diversity (number of varieties) in populations 6H and 10H.

5 Discussion

In our experiments robots with larger genomes (in 10H) systematically outperform those with smaller genomes (in 6H). This is not in itself a particularly interesting result. In our model the number of chromosomes in the genome directly determines the number of hidden units in the neural network the genome expresses. The control networks of robots with larger genomes thus include a larger number of hidden units—and are computationally more powerful—than those of robots with smaller genomes. It seems trivially obvious that computationally stronger robots will outperform feebler ones.

The significance of the results reported in this paper derives from the rigorous demonstration that in the majority of robots with larger genomes a significant number of chromosomes (and hidden units) are *functionally redundant*: deleting them has no measurable effect on behavior. This shows that the obvious explanation for the superior performance of robots with larger genomes is false: the superior performance of the robot with large genomes is independent of their extra computational power. This result is strengthened by the comparison between the performance of robots in 10H and 6H whose SSFC (expressing the *functional core* of the control network) has the same size. Here too, robots belonging to 10H outperformed robots in 6H. If two populations of robots have the same computational potential but one outperforms the other, this implies that the reasons for the difference in performance lie not in the *computational* but in the *evolutionary* process. We suggest, therefore, that robots in 10H outperform robots in 6H because their genotypes are more *evolvable*. This superior evolvability, we will argue, is a consequence of differing levels of redundancy in the two populations.

Systematic lesioning has shown that the majority of robots belonging to 10H have a functional core of six or fewer chromosomes and thus include at least four redundant chromosomes; the SSFCs of robots in 6H are only slightly smaller; the number of redundant chromosomes in these robots is usually not larger than two. It follows that robots in 10H have a higher level of redundancy than those in 6H.

These results are compatible with recent results by Wilke et al. [19] showing that, in the presence of high rates of mutation, natural selection will favor redundant, mutationally robust organisms.² If selection ensures that the genome contains a functional core of a certain minimum size and maximizes redundancy under this constraint, larger genotypes will display higher redundancy than smaller ones. This is consistent with what we observe in our experiments. The results obtained by Wilke et al. do not explain, however, why populations with higher levels of redundancy should achieve higher fitness than those with lower redundancy.

The ability to evolve successfully depends critically on a population's ability to maintain a pool of adaptive variation. The data on population diversity, shown in Figure 7, show that in the early stages of the evolutionary process 6H and 10H both maintained a pool of diversity; as the process progressed, however, 6H came to be dominated by a single variety of robot, while 10H maintained its status as a quasi-species [6, 13] in which different genotypes coexisted stably. It seems likely that the existence of a range of different genotypes enabled 10H to explore a larger area of genotype space than was possible for 6H. This, we would argue, is the immediate explanation for the difference in performance between the two populations. The different levels of diversity in the two populations can best be understood, however, as a consequence of their differing levels of redundancy.

For any redundant genotype there exists, by definition, at least one mutation that connects the genotype to another genotype with the same fitness. In this way the connected genotypes form an island in genotype space. As redundancy increases, the number of connections between genotypes increases; beyond a critical threshold the islands merge, forming a neutral network spanning large regions of space [8]. This is an inevitable consequence of high redundancy. In populations with highly redundant genotypes, evolutionary search can explore the whole neutral network and its immediate mutational neighborhood; populations with lower redundancy have no such network to explore.

Applying this concept to our experiments, we suggest that in the early stages of the evolutionary process robots used small SSFCs which could only perform very simple tasks. Redundancy was high in both populations. This enabled them to explore large neutral networks until they acquired the ability to perform more complex tasks. These, as a rule, involved larger SSFCs and lower levels of redundancy. If this level was still sufficiently high, the population could then begin to explore a new neutral network, lying at a higher altitude on the landscape. This step by step process—which could be seen as an automated version of the *shaping* sometimes used to train artificial neural networks [5]—enabled robots to perform tasks of gradually increasing complexity. The process continued until redundancy fell below a critical level and the neutral network broke up or until evolutionary search discovered a globally optimum strategy.

Experimental results show that the smallest SSFC capable of achieving optimal performance on the experimental task contained five chromosomes and that in practice many well-performing robots had SSFCs of 6. As the SSFCs of robots in 6H approached this size, they lost their redundancy. This implies that instead of exploring a large neutral network, as they had done in the early stages of their evolution, they were now confined to ever smaller regions of genotype space. Robots in 10H that had reached this same level of fitness maintained, on the other hand, a significant number of redundant chromosomes. This redundancy allowed them to explore a neutral network that for robots in 6H no longer existed, and thus to discover adaptive opportunities that 6H was unable to reach. This explains their superior performance.

² Wilke et al. have also found that lower rates of mutation favor genotypes with lower levels of redundancy. While we have not investigated this hypothesis systematically, informal investigations during the tuning phase support Wilke's finding.

To sum up, 10H outperformed 6H because it was able to maintain redundancy at all stages of the evolutionary process; redundancy created a neutral network allowing the stable coexistence of a range of different genotypes; this population diversity allowed 10H to adopt a more efficient evolutionary search strategy than was possible for 6H.

In interpreting this result it should be noted that in our simulation model redundancy was, by design, cost-free. This is of course, unlikely to be true in natural systems, where the replication of unnecessary DNA or the maintenance of unneeded neurons will necessarily exact a metabolic cost. In these systems the effective contribution of redundancy to evolvability will involve a weighing of costs and of benefits. The results of our experiments nonetheless suggest that where costs are sufficiently low, redundancy in functional or potentially functional components of the genome can make a valuable contribution to evolvability. The methods described in this article provide a practical way of testing this hypothesis for the artificial case and provide support for the idea that redundancy may be a defining feature of a broad range of complex adaptive systems [10].

References

1. Burch, C. L., & Chao, L. (2000). Evolvability of an RNA virus is determined by its mutational neighborhood. *Nature*, *406*, 625–628.
2. Calabretta, R., Nolfi, S., Parisi, D., & Wagner, G. P. (2000). Duplication of modules facilitates the evolution of functional specialization. *Artificial Life*, *6*, 69–84.
3. Chaitin, G. J. (1975). Randomness and mathematical proof. *Scientific American*, *232*(5), 47–52.
4. Cliff, D., & Miller, G. F. (1996). Co-evolution of pursuit and evasion 2: Simulation methods and results. In P. Maes et al. (Eds.), *From animals to animats IV* (pp. 506–515). Cambridge, MA: MIT Press.
5. Dorigo, M., & Colombetti, M. (1994). Robot shaping: Developing autonomous agents through learning. *Artificial Intelligence*, *71*(2), 321–370.
6. Eigen, M., & Schuster, P. (1978). The hypercycle part B: “The abstract hypercycle.” *Naturwissenschaften*, *65*, 7–41.
7. Force, A., Lynch, M., Pickett, F. B., Amores, A., Yan, Y. L., & Postlethwait, J. (1999). Preservation of duplicate genes by complementary, degenerative mutations. *Genetics*, *151*, 1531–1545.
8. Huynen, M. A. (1996). Exploring phenotype space through neutral evolution. *Journal of Molecular Evolution*, *43*, 165–169.
9. Huynen, M. A., Stadler, P. F., & Fontana, W. (1996). Smoothness within ruggedness: The role of neutrality in adaptation. *Proceedings of the National Academy of Sciences*, *93*, 397–401.
10. Kauffman, S. A. (2000). *Investigations*. Oxford, UK: Oxford University Press.
11. Miglino, O., Nolfi, S., & Parisi, D. (1996). Discontinuity in evolution, how different levels of organization imply pre-adaptation. In R. Belew & M. Mitchell (Eds.), *Adaptive individuals in evolving populations* (pp. 291–294). Addison-Wesley.
12. Mitchell, M. (1998). *An introduction to genetic algorithms*. Cambridge, MA: MIT Press.
13. Nowak, M. A. (1992). What is a quasispecies? *Trends in Ecology and Evolution*, *7*, 118–121.
14. Ohno, S. (1970). *Evolution by Gene Duplication*. New York: Springer-Verlag.
15. Thompson, A., Harvey, I., & Husbands, P. (1996). Unconstrained evolution and hard consequences. In E. Sanchez and M. Tomassini (Eds.), *Towards Evolvable Hardware* (pp. 136–165). Lecture Notes in Computer Science 1062, Heidelberg: Springer-Verlag.
16. Van Nimwegen, E., Crutchfield, J. P., & Huynen, M. (1999). Neutral evolution of mutational robustness. *Proceedings of the National Academy of Sciences*, *96*, 9716–9720.

17. Wagner, G. P., & Altenberg, L. (1996). Complex adaptations and the evolution of evolvability. *Evolution*, *50*, 967–976.
18. Walker, R. (1999). Niche selection and the evolution of complex behavior in a changing environment. *Artificial Life*, *5*(3), 271–289.
19. Wilke, C. O., Wang, J. L., Ofria, C., Lenski, R. E., & Adami, C. (2001). Evolution of digital organisms at high mutation rates leads to survival of the flattest. *Nature*, *412*, 331–333.