Complex Systems **7** (1993) 327–346

Preserving the Diversity of a Genetically Evolving Population of Nets Using the Functional Behavior of Neurons

Nachum Shamir*
David Saad†
Emanuel Marom
Faculty of Engineering, Tel Aviv University,
Ramat Aviv 69978, Israel

Abstract. Population diversity loss is a major obstacle in applying genetic algorithms to optimization problems, which often results in population degeneration and premature convergence. The diversity changes caused by three natural-selection strategies—comparing new offspring to the least-fit specimen in the population, to one of the parents, and to the most similar specimen in the population—are analyzed theoretically and demonstrated experimentally. Using Hamming distances, the changes in diversity induced by these strategies are analyzed for an evolving population of binary strings. The results of the analysis show that the strategy of comparing new offspring to the most similar specimen (selecting the fitter of the two) causes the smallest diversity loss.

To demonstrate the efficiency of the various methods we examine the population diversity of neural nets trained to perform certain tasks using a genetic algorithm. The functional behavior of neurons, represented by the internal representations of each neuron for the entire training set, is used to derive the functional similarity of every pair of neurons and to evaluate the similarity of every pair of nets in a population of neural networks. Using a measure of the functional behavior of neurons, the changes in diversity are demonstrated for evolving populations of nets trained on the Parity data set. The experimental results demonstrate the success of the third strategy in preserving population diversity throughout generations: overcoming obstacles in the course of training and preventing population degeneration, and thus providing more successful and reliable learning.

*Current address: Department of Electrical Engineering, Technion–Israel Institute of Technology, Technion City, Haifa 32000, Israel.

†Current address: Department of Physics, University of Edinburgh, J. C. Maxwell Building, Mayfield Road, Edinburgh EH9 3JZ, UK.

1. Introduction

Genetic algorithms are currently used for coping with difficult optimization problems. By imitating the principles of survival and evolution from nature, genetic algorithms process an ensemble of solution proposals (called populations), gradually enhancing the average performance while attempting to progress toward the global optimum. In the course of genetic evolution the population might undergo an undesired process of diversity loss, constantly decreasing the variety of its specimens. This loss is caused by highly fit specimens gradually overrunning the entire population, resulting in a complete population degeneration. When the problem being addressed has a single maximum point, higher fitness directly indicates a better solution proposal, and the population is likely to degenerate to the global optimum. However, most nontrivial optimization problems have a large number of local maxima (they are often called "deceptive problems"), and the genetic process degenerates in many cases to a local optimum instead of the global one [1, 2, 3]. In artificial genetic evolution, as well as in nature, more-fit specimens are given greater opportunities to reproduce. This *selective pressure* [1, 17] is counterbalanced by mutations and random crossover operations that add stochasticity to the evolution of the population. Population diversity should be preserved to prevent degeneration while maintaining the general trend of the evolution and some sort of selective pressure. The amount of diversity required is similar to the annealing procedure in simulated annealing optimization algorithms, and is likely to vary from problem to problem; the harder the optimization problem, the larger the diversity required. Examining end results in artificial genetic evolution, we obviously have no interest in specimen duplication since we are interested only in the fittest specimen. Thus there is no need for the entire population to degenerate to the global optimum, and the evolutionary process can successfully end without losing the population diversity.

Preserving population diversity is essential for locating a global optimum, but is also crucial for coping with time-varying problems and tasks in which more than a single solution is required (locating eigenvalues, resonance modes, etc.). In nature, population diversity is maintained as evident by the variety of species and specimens. Various mechanisms explain how, in spite of Darwin's "survival of the fittest" law, less-fit specimen characteristics survive and variety is maintained [4, 5]:

1. The population is not in equilibrium, and less-fit specimens have not been driven out.

2. The variation of environmental conditions in space and time causes different species to adapt to different environments. Migration of specimens between the various environments results in an equilibrium state where the various species thrive.

3. Selection is not done with respect to the entire population, but to the subpopulations. The subpopulation is characterized by its need for a common resource, for which its individuals must compete.
4. Mutations and random fluctuations applied during reproduction introduce new characteristics to the population.

These mechanisms, which preserve the diversity of species in nature, have been integrated into artificial genetic algorithms in various ways. Random mutations are applied to every new solution proposal in an attempt to slow down degeneration and introduce new characteristics to the population. Since mutations must be of limited extent (so that the genetic process is not reduced into a meaningless random search process), they cannot prevent degeneration but only slow it down. Additional methods for introducing stochastic fluctuations during reproduction are inversion [6] and dynamic parameter encoding [7], which control the location and length of every encoded feature.

A different approach was demonstrated by De Jong [8], who added a *crowding factor* that affects the selection algorithm. For each new offspring a small subset of the population is selected at random and the most similar specimen in that subset is compared to the new offspring. Mauldin [9] proposed a *uniqueness* measure that forced new offspring to maintain a minimal distance from the most similar specimens, and experimentally demonstrated the superiority of crowding, uniqueness, and their combinations with respect to survival of the fittest and random-selection rules.

Coping with multimodal function optimizations requires selection rules capable of maintaining independent subpopulations in specific niches of the search space. The size of each subpopulation must be restricted to prevent it from overrunning the entire population. In nature, if a particular subpopulation grew with no limit, it would consume all available environmental resources, resulting in starvation and/or elimination of all specimens. Since competition would arise within this subpopulation, its final size would be determined by the equilibrium between the number of organisms and the resources they consume. The *sharing* mechanism proposed by Goldberg and Richardson [12] introduces artificial competition among similar specimens by subtracting a penalty term from the fitness of every individual according to the number and distance of its neighbors. This sharing mechanism has successfully preserved the diversity of the entire population, prevented the subpopulation from becoming oversized, and was extremely successful in locating the various modes of the tested multimodal functions [13, 14, 15].

The use of genetic algorithms for neural net training is done in a variety of methods where network weights and structure are encoded into artificial "chromosomes" [16, 17, 18, 19]. Applying diversity-preserving methods such as crowding, uniqueness, and sharing to those training methods is extremely difficult, and often impossible, due to the problem of permutation [11]. The problem of permutation arises from the fact that neurons that perform equivalent tasks are located at different positions in the hidden layers, and are therefore encoded at different locations in the chromosomes. Comparing pairs of chromosomes as they are arbitrarily ordered may produce meaningless random results and damage the performance of those processes responsible for maintaining the population diversity. To overcome that obstacle the use of the functional behavior measure is proposed. This measure is capable of scoring the functional and structural resemblance of nets having arbitrary hidden neuron order and different sizes.

In this work we discuss the diversity changes caused by three natural-selection strategies:

1. comparing new offspring to the least-fit specimen in the population;
2. comparing new offspring to one of the parents; and
3. comparing new offspring to the most similar specimen in the population.

These three strategies are theoretically analyzed and experimentally examined in the following sections. Using Hamming distances between the new offspring and the various specimens in the population, we discuss the effect each natural-selection rule has on the diversity of binary string populations and, for a population of neural networks, use the functional behavior of neuron measure to apply the same natural-selection rules, then demonstrate and monitor their performance.

2. Using Hamming distance for diversity analysis

For every pair of binary strings $v_i = (b_1^i, \ldots, b_n^i)$ and $v_j = (b_1^j, \ldots, b_n^j)$, where b is equal to 0 or 1, the Hamming distance is defined by

$$H(v_i, v_j) = \sum_{k=1}^{n} b_k^i \oplus b_k^j \tag{1}$$

where $\oplus$ signifies a binary "exclusive-or" operation. The normalized Hamming distance is obtained by dividing the Hamming distance by the number of bits n:

$$\tilde{H}(v_i, v_j) = \frac{H(v_i, v_j)}{n} \tag{2}$$

so that $\tilde{H}(v_i, v_j)$ is limited to the interval $[0, 1]$. Hamming distance is used for grading the difference between binary strings that have an equal number of bits. By calculating the normalized Hamming distance average, one can estimate the *diversity* of a given set.of strings

$$D = \frac{1}{m(m-1)/2} \sum_{i=1}^{m-1} \sum_{j=i+1}^{m} \tilde{H}(v_i, v_j) \tag{3}$$

where m is the defined set size and $m(m-1)/2$ is the number of possible string pairs in the set. All diversity values lie in the interval $[0, 1]$.

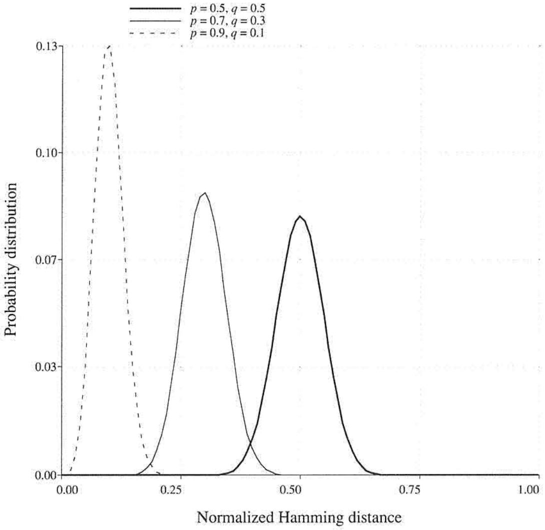


Figure 1: Normalized Hamming distance distribution for corresponding bit probabilities of $p = 0.5$, $p = 0.7$, and $p = 0.9$. As the probability of corresponding bits having equal value is increased, the mean value decreases and the distribution becomes narrower, indicating a reduced standard deviation and thus lower diversity.

The Hamming distance distribution function P is easily derived (see [20]):

$$P(H = k) = \binom{n}{k} q^k p^{n-k} \qquad k = 0, \ldots, n \tag{4}$$

where p is the probability that corresponding bits of different strings have equal value, $q = 1 - p$ is the probability of those bits being different, and $\binom{n}{k}$ is the number of different k-bit groups that could be selected from a given n-bit string. The distribution in equation (4) is a binomial distribution having expectancy $\langle H \rangle = nq$ and variance $\text{Var}(H) = npq$, and therefore the normalized Hamming distance expectancy is $\langle \tilde{H} \rangle = q$ and its variance is $\text{Var}(\tilde{H}) = pq$. The effect of bit probability on the distribution of normalized Hamming distance is illustrated in Figure 1 for probability values of $p = 0.5$, $p = 0.7$, and $p = 0.9$. As the odds of corresponding bits having equal value increase, the mean value of the distribution is reduced and the distribution becomes narrower, indicating a drop in the standard deviation and thus lower diversity.

The effect of genetic reproduction and the natural-selection rule can now be discussed using statistical tools. In every reproduction a pair of strings is selected, statistically preferring those with higher scores (*selective pressure* [1, 17]), and a new offspring is created using random crossover. Let parent strings be $v_{p_1} = (b_1^{p_1}, \ldots, b_n^{p_1})$ and $v_{p_2} = (b_1^{p_2}, \ldots, b_n^{p_2})$, and the offspring string be $v_{\text{offs}} = (b_1^{p_1}, \ldots, b_r^{p_1}, b_{r+1}^{p_2}, \ldots, b_n^{p_2})$, where r is an arbitrary point of crossover; mutations are neglected. Since r bits in v_{offs} are identical to the corresponding bits of v_{p_1}, and $n-r$ bits are identical to the corresponding bits of v_{p_2}, the resulting normalized Hamming distances expectancies are

$$\begin{aligned} \left\langle \tilde{H}(v_{p_1}, v_{\text{offs}}) \right\rangle &\approx \frac{(n-r)q}{n} \\ \left\langle \tilde{H}(v_{p_2}, v_{\text{offs}}) \right\rangle &\approx \frac{rq}{n} \end{aligned} \qquad (5)$$

If the new offspring is found to be more fit than an existing string in the set, that string will be replaced by the new offspring, usually causing a diversity drop for the entire set. The initial diversity estimation before the string exchange, D_q, is computed according to equation (3). When the new offspring v_{offs} replaces a selected string v_e, the new diversity is D_{q+1}, so the change in diversity is

$$\Delta D = D_{q+1} - D_q = \frac{1}{m(m-1)/2} \sum_{i=1, i\neq e}^{m} \left[\tilde{H}(v_i, v_{\text{offs}}) - \tilde{H}(v_i, v_e) \right] \qquad (6)$$

since all Hamming distances not involving v_e and v_{offs} remain unchanged before and after the exchange. The diversity change expectancy is thus

$$\langle \Delta D \rangle = \frac{1}{m(m-1)/2} \sum_{i=1, i\neq e}^{m} \left[\left\langle \tilde{H}(v_i, v_{\text{offs}}) \right\rangle - \left\langle \tilde{H}(v_i, v_e) \right\rangle \right] \qquad (7)$$

It is assumed that population size is significantly larger than the string length ($m \gg n$), and that there is no statistical dependence among strings in the population, except between the offspring and its parents. Three cases for the calculation of the diversity change expectancy are now discussed.

- **The offspring replaces a randomly selected string from the population.** Since there is no statistical dependence between v_e and the rest of the population,

$$\sum_{i=1, i\neq e}^{m} \left\langle \tilde{H}(v_i, v_e) \right\rangle \approx \sum_{i=1, i\neq e}^{m} q = (m-1)q \qquad (8)$$

For v_{offs}, a slightly different situation exists since there is a statistical dependence with the two parents strings:

$$\begin{aligned} \sum_{i=1, i\neq e}^{m} \left\langle \tilde{H}(v_i, v_{\text{offs}}) \right\rangle &= \left\langle \tilde{H}(v_{p_1}, v_{\text{offs}}) \right\rangle + \left\langle \tilde{H}(v_{p_2}, v_{\text{offs}}) \right\rangle \\ &\quad + \sum_{i=1, i\neq e, p_1, p_2}^{m} \left\langle \tilde{H}(v_i, v_{\text{offs}}) \right\rangle \\ &\approx \frac{(n-r)q}{n} + \frac{rq}{n} + \sum_{i=1, i\neq e, p_1, p_2}^{m} q \\ &= \frac{(n-r)q}{n} + \frac{rq}{n} + (m-3)q = (m-2)q \end{aligned} \qquad (9)$$

From equations (7), (8), and (9), the diversity change expectancy is derived:

$$\langle \Delta D \rangle \approx \frac{1}{m(m-1)/2}\left[(m-2)q-(m-1)q\right]$$
$$= -\frac{2q}{m(m-1)} \qquad (10)$$

where the negative sign indicates diversity loss.

- **The offspring replaces one of its parents.** Assuming that both parents have equal chances of being replaced, the Hamming expectancy of the surviving parent with the new offspring is the average of the two expectancies in equation (5):

$$\left\langle \tilde{H}(v_p, v_{\text{offs}}) \right\rangle = \frac{\left\langle \tilde{H}(v_{p_1}, v_{\text{offs}}) \right\rangle + \left\langle \tilde{H}(v_{p_2}, v_{\text{offs}}) \right\rangle}{2}$$
$$\approx \frac{\frac{(n-r)q}{n} + \frac{rq}{n}}{2} = \frac{q}{2} \qquad (11)$$

$\sum_{i=1,i\neq e}^{m} \left\langle \tilde{H}(v_i, v_e) \right\rangle$ remains the same as in equation (8), but v_{offs} statistically depends on only one parent so that

$$\sum_{i=1,i\neq e}^{m} \left\langle \tilde{H}(v_i, v_{\text{offs}}) \right\rangle = \left\langle \tilde{H}(v_p, v_{\text{offs}}) \right\rangle + \sum_{i=1,i\neq e,p}^{m} \left\langle \tilde{H}(v_i, v_{\text{offs}}) \right\rangle$$
$$\approx \frac{q}{2} + \sum_{i=1,i\neq e,p}^{m} q = \frac{q}{2} + (m-2)\,q \qquad (12)$$
$$= \left(m - \frac{3}{2}\right) q$$

The diversity change expectancy is

$$\langle \Delta D \rangle \approx \frac{1}{m\,(m-1)\,/2}\left[\left(m-\frac{3}{2}\right) q - (m-1)\,q\right]$$
$$= -\frac{q}{m(m-1)} \qquad (13)$$

where the negative sign again indicates diversity loss.

- **The offspring replaces the most similar string in the population.** The population is scanned and the string v_e having the lowest Hamming distance to the new offspring v_{offs} is found. If the new offspring is better, it will replace v_e. Let s be the number of matching bits in v_{offs} and v_e. It is important to note that $s \geq n/2$ since v_e could be one of the parents if a more similar string cannot be found. In calculating the diversity loss one must take into account the Hamming distance expectancy of string v_e with the two parent strings, since it is no longer possible to assume that they are statistically independent. v_{offs} has r bits that are identical to v_{p_1}; since s bits in v_{offs} are identical to corresponding bits in v_e, the average number of identical bits between v_e and v_{p_1} is $(s/n)r$. For the same reasons the average number of identical bits between v_e and v_{p_2} is $(s/n)(n-r)$, and the normalized Hamming expectancies are

$$\begin{aligned}\left\langle \tilde{H}(v_{p_1}, v_e)\right\rangle &\approx \frac{n-(s/n)r}{n}q\\ \left\langle \tilde{H}(v_{p_2}, v_e)\right\rangle &\approx \frac{n-(s/n)(n-r)}{n}q\end{aligned} \tag{14}$$

$\sum_{i=1,i\neq e}^{m}\left\langle \tilde{H}(v_i, v_{\text{offs}})\right\rangle$ remains the same as in equation (9), and the sum of terms involving v_e is

$$\begin{aligned}\sum_{i=1,i\neq e}^{m}\left\langle \tilde{H}(v_i, v_e)\right\rangle &= \left\langle \tilde{H}(v_{p_1}, v_e)\right\rangle + \left\langle \tilde{H}(v_{p_2}, v_e)\right\rangle + \sum_{i=1,i\neq e,p_1,p_2}^{m}\left\langle \tilde{H}(v_i, v_e)\right\rangle\\ &\approx \frac{n-(s/n)r}{n}q + \frac{n-(s/n)(n-r)}{n}q + \sum_{i=1,i\neq e,p_1,p_2}^{m} q\\ &= \frac{n-(s/n)r}{n}q + \frac{n-(s/n)(n-r)}{q} + (m-3)q\\ &= \frac{n-s}{n}q + (m-2)q\end{aligned} \tag{15}$$

The diversity change expectancy is

$$\begin{aligned}\langle \Delta D\rangle &\approx \frac{1}{m(m-1)/2}\left[(m-2)q - (m-2)q - \frac{n-s}{n}q\right]\\ &= -\frac{n-s}{n}\frac{2q}{m(m-1)}\end{aligned} \tag{16}$$

Note that s has the range $n/2 \leq s \leq n$. When $s = n/2$ the result is identical to the case when one of the parents is replaced, whereas for $s = n$ no diversity change takes place since a string in the population is replaced by an identical one.

The lowest diversity change is obtained for this last case (equation (16)), thus indicating the superiority of the method that compares every new offspring to the most similar string in the population and thus prevents string duplications. In this discussion, statistical independence of all strings was assumed. When diversity is not preserved, each generation produces more dependent strings. Given a high rate of diversity loss, finite population size, and a long evolutionary cycle, one can no longer neglect the effects of the accumulation of similar strings on the population diversity. In this case one cannot assume statistical independence, so diversity change approximations are no longer valid and diversity collapses. The diversity loss predicted by our analysis (equations (10), (13), and (16)) is therefore replaced by a nonlinearly increasing diversity loss that deteriorates as the population continues to evolve.

Applying the proposed diversity-maintaining method to genetic algorithms used for neural net training requires a new measure for scoring the resemblance of nets and neurons with continuous output response. The Hamming distances cannot be used in this case since continuous net parameters are directly encoded into the population. The measure presented in this work is based on the functional behavior operator [10, 11], described briefly in the next section.

3. Evaluating net similarity using the functional behavior of neurons

The "functional behavior" of a neuron describes how that neuron and its corresponding sub-net respond when various input vectors are presented to the net. The net is assumed to have no feedback connections. Each sub-net is a part of the net starting at the input layer and ending at a single hidden or output neuron that is the output neuron of the sub-net. The sub-net contains all relevant neurons from previous layers and all interconnections that lead to those neurons. Every sub-net realizes, therefore, a function $f : \Re^n \rightarrow \Re$ on the input vectors, where n is the number of input neurons. This function is defined as the sub-net response function. The output of the sub-net ending at neuron i is represented by

$$s^i = f^i(v_1, \ldots, v_n) \tag{17}$$

where $(v_1, \ldots, v_n)$ represents the input vector.

A neuron's functional behavior is defined as the vector of output values generated for the entire set of input vectors:

$$B^i = (s_1^i, \ldots, s_p^i) \tag{18}$$

where i is the neuron index, s_j^i is the output value of neuron i when the input vector indexed j is fed to the net, and p is the number of input vectors.

In order to compare different neurons and different functional behaviors, the measure is normalized with respect to its overall norm $E^i = \sum_{j=1}^{p} {s_j^i}^2$:

$$\tilde{B}^i = \left(\frac{s_1^i}{\sqrt{E^i}}, \ldots, \frac{s_p^i}{\sqrt{E^i}} \right) \tag{19}$$

The degree of matching between a pair of neurons i_1 and i_2 is defined by the correlation of their corresponding normalized functional behaviors:

$$\text{match}(i_1, i_2) = \tilde{B}^{i_1} \cdot \tilde{B}^{i_2} = \frac{1}{\sqrt{E^{i_1} E^{i_2}}} \sum_{j=1}^{p} s_j^{i_1} s_j^{i_2} \tag{20}$$

This normalized matching factor lies in the interval $[-1, 1]$. Its amplitude determines how close is the behavior of the corresponding sub-nets, where a negative sign stands for opposite response. It is important to note that for linearly dependent functional behavior vectors the matching factor is either 1 or -1:

$$\begin{aligned}\mathrm{match}(i_1, i_2)_{B^{i_1}=\alpha B^{i_2}} &= \tilde{B}^{i_1} \cdot \tilde{B}^{i_2} = \frac{1}{\sqrt{\alpha^2 E^{i_2} E^{i_2}}} \sum_{j=1}^{p} \alpha s_j^{i_2} s_j^{i_2} \\ &= \mathrm{sign}(\alpha)\end{aligned} \tag{21}$$

where α denotes the linear dependence.

Net similarity is evaluated by the average neuron matching according to equation (20). Both nets are processed layer by layer starting with the first hidden layer and ending with the output layer. For each neuron in the first net, the most similar neuron in the other net is found and vice versa:

$$\begin{aligned} S_l^1 &= \sum_{j=1}^{N_l^1} \max_{k=1,\dots,N_l^2} \left|\mathrm{match}(n_{l,j}^1, n_{l,k}^2)\right| \\ S_l^2 &= \sum_{j=1}^{N_l^2} \max_{k=1,\dots,N_l^1} \left|\mathrm{match}(n_{l,j}^2, n_{l,k}^1)\right| \\ \mathrm{match}(\mathrm{net}_1, \mathrm{net}_2) &= \frac{\sum_{l=1}^{L} (S_l^1 + S_l^2)}{\sum_{l=1}^{L} (N_l^1 + N_l^2)} \end{aligned} \tag{22}$$

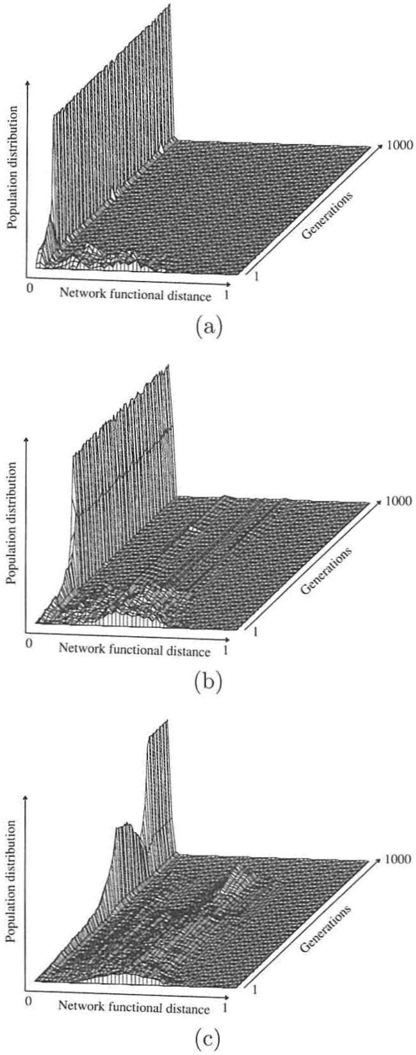
where L is the number of hidden (and output) layers, N_l^i the number of neurons in layer l of net i, and $n_{l,j}^i$ is neuron j in layer l of net i. The computational complexity of this calculation is approximately $O(N^2 p)$, where N is the average number of neurons in the hidden and output layers and p is the number of training vectors.

This definition breaks down each net into its basic functional elements and uses those elements to compare the nets' functional structure, overcoming the problem of hidden neuron location permutation. It is important to note that nets may also have a different number of neurons in each layer, but must have an equal number of layers. The choice of neurons that have maximal matching factors permits multiple use of a single neuron. This does not affect the integrity of the net comparison method since, when proper compensation measures are taken, neurons may be duplicated (with their entire set of input connections) without changing net performance (see [11]).

Following the diversity definition of equation (3), the diversity of a population having m nets is defined by their functional distances $\mathrm{dist}(\mathrm{net}_i, \mathrm{net}_j) = 1 - \mathrm{match}(\mathrm{net}_i, \mathrm{net}_j)$ to be:

$$D = \frac{1}{m(m-1)/2} \sum_{i=1}^{m-1} \sum_{j=i+1}^{m} \mathrm{dist}(\mathrm{net}_i, \mathrm{net}_j) \tag{23}$$

Diversity value lies in the interval $[0, 1]$.



(a)

(b)

(c)

Figure 2: Functional distance distribution curves for genetic back propagation training: using the natural-selection strategy of comparing new offspring to the least-fit specimen. The rapid loss of diversity and population degeneration is evident, being caused by a single highly fit net that overruns the entire population. (a) Population size 20. Rapid degeneration resulted with a functionally identical population after only 120 generations. (b) Population size 40. After 290 generations most population specimens have become functionally identical. (c) Population size 80. After 580 generations a highly fit specimen began overrunning the population. However, before population had degenerated, a different, more "powerful" specimen evolved, first slightly restoring diversity but then overrunning the entire population, causing final degeneration at generation 850.

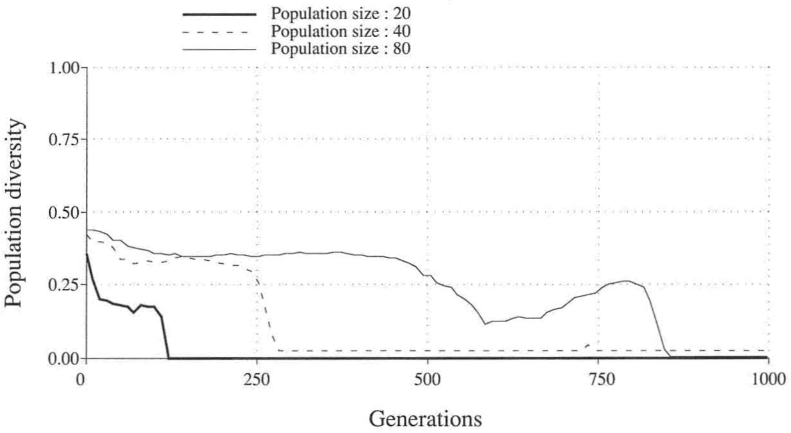


Figure 3: Diversity curves for population sizes of 20, 40, and 80, where the natural-selection strategy of comparing new offspring to the least-fit specimen has been used. In all experiments the population underwent a complete degeneration after 120, 290, and 850 generations, respectively.

4. Experimental results

The performance of the three natural-selection strategies—comparing new offspring to the least-fit specimen in the population, to one of the parents, and to the most similar specimen in the population—is tested using the Parity-7 data set. In this data set the net is trained to produce the parity bit for a 7-bit binary string, that is, a +1 output whenever an odd number of +1 bits are presented to the net and −1 otherwise. The data set contains $2^7 = 128$ vectors, each representing 7 input bits and a single output bit. This problem was chosen due to its multiple minima structure.

Using a hybrid genetic algorithm–back propagation training system (see [11]), populations of 20, 40, and 80 nets were trained exhaustively. In every generation a new offspring net was created by recombining pairs of nets from the population. The offspring net was subjected to 50 training epochs, and two pruning and retraining cycles, where retraining was also limited to 50 epochs. After retraining, the offspring net (v_{offs}) competed with a selected net (v_e) chosen from the population according to the tested strategy. Every 10 epochs the population diversity and the distribution of the net functional distances were measured; the results are displayed in Figures 2–7.

In the first set of experiments the behavior of the training system was examined when every new offspring was compared to the least-fit specimen in the population. Figures 2(a), 2(b), and 2(c) display distribution curves for population sizes of 20, 40, and 80 nets, respectively, and Figure 3 shows the changes in diversity across generations. The 20-net population was the first to degenerate, surviving only 120 generations before the entire population became functionally identical. The 40-net population did not survive much longer, with final degeneration occurring after 290 generations. For the 80-net population an interesting result was obtained, with the population partially recovering from degeneration. This phenomenon was caused by a highly fit specimen that overran the population after 580 generations; but in the course of evolution a different, more "powerful" specimen was created, causing a slight restoration of diversity (when population was composed of two degenerated groups) until it too overran the entire population, causing final degeneration at generation 850. These experiments indicate that a larger population could survive longer (as was anticipated by equation (10)), but no matter what size was selected, given enough time the population would degenerate, terminating the evolutionary process and preventing further population enhancement.

In the second set of experiments the behavior of the training system was examined when every new offspring was compared to one of its parents (selected at random). Figures 4(a), 4(b), and 4(c) display distribution curves for population sizes of 20, 40, and 80 nets, respectively, and Figure 5 shows the changes in diversity across generations. For the three population sizes a slow but steady loss of diversity is observed, with smaller populations suffering higher loss rates. The number of functionally equivalent nets grows steadily, as indicated by the accumulating peaks at zero distance on the distribution curves. The diversity loss in this case is a continuous process, whereas in the previous set of experiments it could be categorized as a deteriorating process that ended with a final collapse.

The most interesting results were obtained when new offspring were compared to the most similar specimens in the population. The entire population was scanned in each generation, using equation (22) to compare all specimens to the new offspring and to find the most similar one. If the new offspring was found to be more fit, it was put into the population instead of the most similar specimen; otherwise it was discarded. Figures 6(a), 6(b), and 6(c) display distribution curves for population sizes of 20, 40, and 80 nets, respectively, and Figure 7 shows the changes in diversity across generations. Only minor loss of diversity is observed for the smallest population, while the other two remain intact across generations. It is important to note that the measures taken to preserve the diversity did not damage the speed of learning (as measured by the performance of the most-fit net) with respect to the other strategies, and the quality of genetic search improved, resulting in smaller and more efficient final nets.

This experiment was repeated using a simplified net comparison method, comparing nets only by matching the functional behavior of their output neurons (equation (20)) instead of using equation (22) to compare the entire nets. The computational complexity of every comparison in this method is $O\left(N_{\text{out}}p\right)$ where N_{out} is the number of output neurons and p is the number of training vectors. It is significantly smaller than the complexity of equation (22) ($O\left(N^2p\right)$). The simplified method produced similar results and preserved population diversity as well.

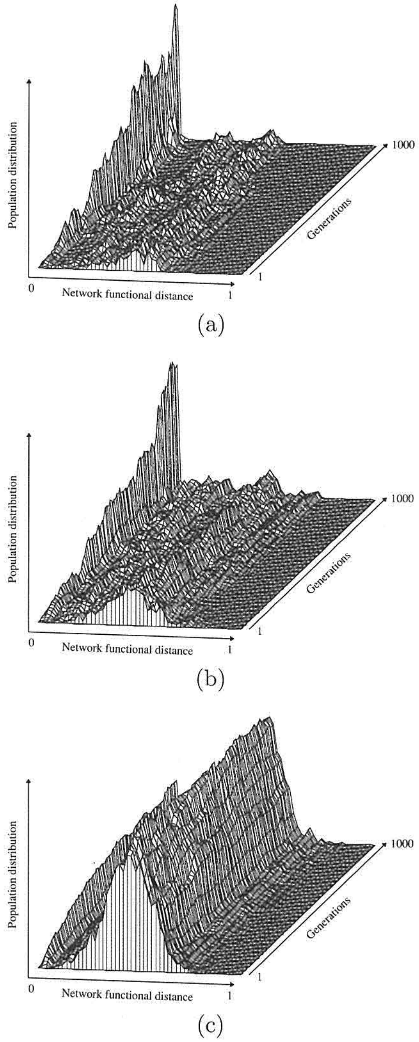


Figure 4: Functional distance distribution curves for genetic back propagation training: using the natural-selection strategy of comparing new offspring to one of the parents (selected at random). A slow but steady loss of diversity is seen across generations, and the accumulating peaks at zero distance indicate the growing number of duplicated specimens in the population. (a) Population size 20. (b) Population size 40. (c) Population size 80.

The benefits of diversity preservation come into effect when training is done in the presence of local maxima. In such cases, nonpreserving training may cause population degeneration in the vicinity of one of those maxima, while the preserving training may continue to progress across generations.

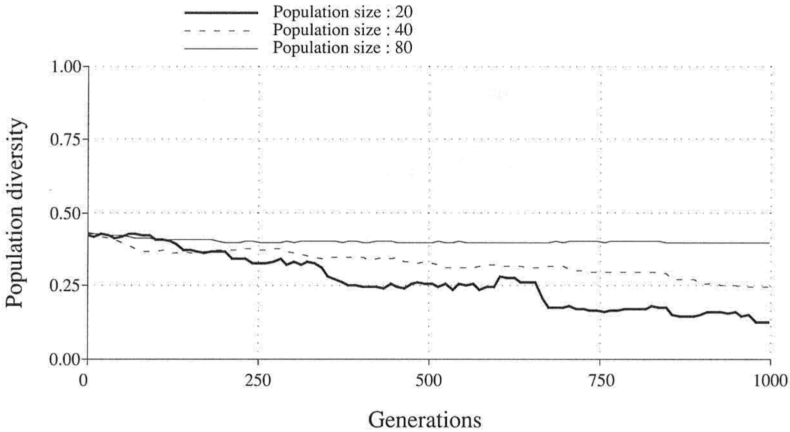


Figure 5: Diversity curves for population sizes of 20, 40, and 80, where the natural-selection strategy of comparing every new offspring to one of the parents (selected at random) has been used. In all experiments a slow but steady loss of diversity is seen across generations, where smaller populations suffer higher loss rates.

Using the Parity-7 data set, such an experiment was performed. Forty of the 128 vectors were used for back propagation training, and the entire set of 128 vectors was used for fitness determination for each offspring net. Experiments indicated that this setup diverts the genetic training into a local maximum in the form of a very simple net, having a single hidden neuron, that responds with correct output polarities to 114 of the 128 training vectors (89%). Note that this setup was chosen to demonstrate the performance of the genetic training in the presence of highly influential local maxima and that this configuration is not the most efficient one for training nets to solve a Parity problem.

The performance of the three natural-selection strategies was examined by performing 300 different genetic–back propagation trainings, 100 for each strategy. In each training a population of 20 nets was evolved, and the performance of the best net in the population was tested after 250, 500, 1000, and 2000 generations. The statistical results are displayed in Tables 1(a)–1(c). Each row in the three tables represents the distribution of best net performance after the 100 populations had evolved for the corresponding number of generations. When new offspring were compared to the least-fit specimen in the population (Table 1(a)), 89 of the 100 trained populations were still trapped at the local maximum of 89%. Since diversity was not preserved, only a small percentage (9%) of the trained populations had overcome the local maximum after 2000 generations. When new offspring were compared to one of their parents (Table 1(b)), there was minor improvement, but again most populations failed to overcome the local maximum. Since diversity was again insufficiently preserved, only 14% of the populations had overcome the local maximum of 89% after 2000 generations.

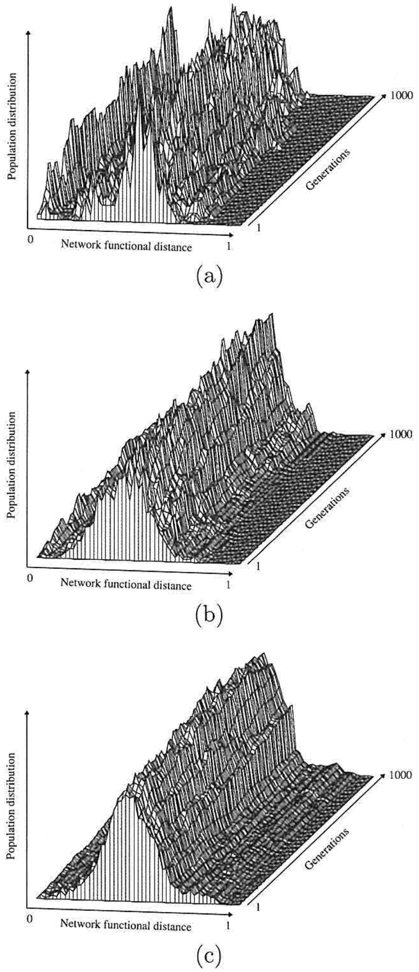


Figure 6: Functional distance distribution curves for genetic back propagation training: using the natural-selection strategy of comparing new offspring to the most similar specimen in the population. Only minor loss of diversity is observed for the smallest population, while the other two remain intact across generations. (a) Population size 20. (b) Population size 40. (c) Population size 80.

Only when the strategy of comparing new offspring to the most similar specimen in the population was used did the population diversity prevail and training successfully overcome the 89% obstacle. Observing Table 1(c) one can clearly see the improvement where 21%, 49%, 85%, and 95% of the experiments successfully overcame the 89% local maximum after 250, 500, 1000, and 2000 generations, respectively. These results demonstrate

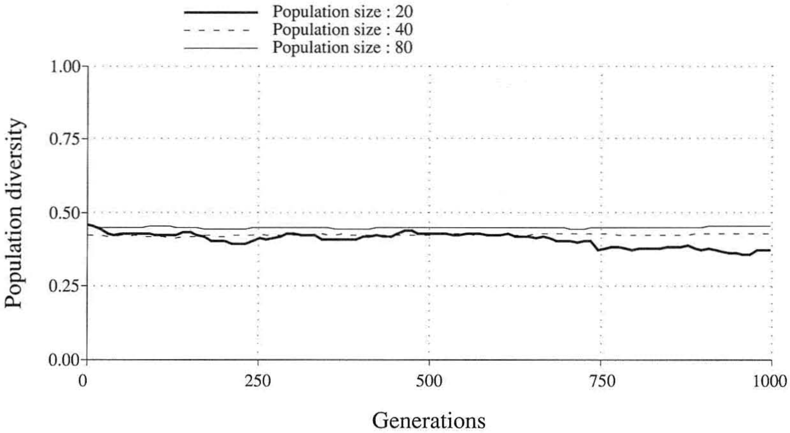


Figure 7: Diversity curves for population sizes of 20, 40, and 80, where the natural-selection strategy of comparing new offspring to the most similar specimen in the population has been used. Only the smallest population suffers minor diversity loss across generations while the other two remain perfectly intact.

the vitality of diversity preservation to the success and continuation of the genetic process.

It is important to note that when the strategy of comparing new offspring to the most similar specimen is used, each new offspring must be compared to all specimens in the population, whereas no such comparisons are required when one of the other methods is used. These additional computations are negligible in comparison to the total training time since the computation time required to train each offspring is much more significant. In addition, since population variety is maintained across generations, one can reduce the population size required to complete the training, thus reducing significantly the training time as well as the number of comparisons required in the third method. We have not encountered any practical difficulties implementing the third strategy in our simulations.

5. Conclusion

In this work the diversity changes caused by three natural selection strategies—comparing new offspring to the least-fit specimen in the population, to one of the parents, and to the most similar specimen in the population—were analyzed theoretically and demonstrated experimentally. Using Hamming distances, the changes in diversity induced by those strategies were analyzed for an evolving population of binary strings; using the functional behavior of neurons measure, the changes in diversity were demonstrated for evolving populations of neural networks trained by a Parity data set. Both theoretical analysis and experimental results indicate the superiority of the last strategy in its ability to maintain population diversity throughout genetic evolution,

	Best net success rate (%)													
Generation	85	86	87	88	89	90	91	92	93	94	95	96	97	98
250	2				91	3	2				1			1
500	2				90	2	4				1			1
1000	2				89	1	3	1		1	1	1		1
2000	2				89		4			1	1		2	1

(a)

	Best net success rate (%)													
Generation	85	86	87	88	89	90	91	92	93	94	95	96	97	98
250				1	86	3	6	2	2					
500					86	2	8				2		1	1
1000					86	1	9				2		1	1
2000					86		10					2	1	1

(b)

	Best net success rate (%)													
Generation	85	86	87	88	89	90	91	92	93	94	95	96	97	98
250				4	75	10	9	1		1				
500				1	50	13	28	2	3	2	1			
1000					15	16	47	7	6	5	4			
2000					5	6	42	10	6	10	13	4	2	2

(c)

Table 1: Results of statistical performance tests for (a) comparing new offspring to the least-fit specimen in the population, (b) comparing new offspring to one of the parents, and (c) comparing new offspring to the most similar specimen in the population. In each training the performance of the best net was tested after 250, 500, 1000, and 2000 generations, and the distribution of results is displayed according to the corresponding natural-selection strategy. Each row in the three tables represents the distribution of best net performance after the 100 populations have evolved for the corresponding number of generations. The majority of trainings in the first two strategies were trapped at the local maximum of 89%. However, when the strategy of comparing new offspring to the most similar specimen in the population was used, 21%, 49%, 85%, and 95% of the experiments successfully overcame the 89% local maximum after 250, 500, 1000, and 2000 generations, respectively, demonstrating the significance of diversity preservation to the success of the genetic process.

and statistical experimental results demonstrate its ability to overcome obstacles in the course of training (such as local maxima). The influence of population size on the diversity loss rate is predicted by the theoretical analysis and is demonstrated by computer simulations for evolving populations of neural networks. The successful use of the functional behavior of neurons measure for evaluating net similarity provides the means for integrating other diversity-maintaining schemes such as crowding, uniqueness, and sharing into genetic algorithms used for neural net training. These promising possibilities provide fertile ground for further research into making better use of genetic algorithms for neural net training.

Acknowledgments

We thank J. Shamir for helpful discussions, advice, and support.

References

[1] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning* (Reading, MA: Addison-Wesley, 1989).

[2] D. E. Goldberg, "Simple Genetic Algorithms and the Minimal Deceptive Problem", pages 74–88 in *Genetic Algorithms and Simulated Annealing*, edited by L. Davis (London: Pitman, 1987).

[3] D. E. Goldberg, K. Deb, and B. Korb, "Messy Genetic Algorithms Revisited: Studies in Mixed Size and Scale," *Complex Systems*, **4** (1990) 415–444.

[4] J. F. C. Kingman, "Mathematics of Genetic Diversity," Society for Industrial and Applied Mathematics, Philadelphia (1980).

[5] W. H. Li, "Maintenance of Genetic Variability Under Mutation and Selection Pressures in a Finite Population," *Proceedings of the National Academy of Sciences*, **74** (1977) 2509–2513.

[6] K. Shahookar and P. Mazumder, "A Genetic Approach to Standard Cell Placement Using Meta-Genetic Parameter Optimization," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, **9** (1990) 500–511.

[7] N. N. Schraudolph and R. K. Belew, "Dynamic Parameter Encoding for Genetic Algorithms," *Machine Learning*, **9** (1992) 9–20.

[8] K. A. De Jong, "An Analysis of the Behavior of a Class of Genetic Adaptive Systems," *Dissertation Abstracts International*, **36** (1976) 5140B.

[9] M. L. Mauldin, "Maintaining Diversity in Genetic Search," pages 247–250 in *Proceedings of the National Conference on Artificial Intelligence* (1984).

[10] N. Shamir, D. Saad, and E. Marom, "Neural Net Pruning Based on Functional Behavior of Neurons," *International Journal of Neural Systems*, **4** (1993) 143–158.

[11] N. Shamir, D. Saad, and E. Marom, "Using the Functional Behavior of Neurons for Genetic Recombination in Neural Nets Training," *Complex Systems*, forthcoming.

[12] D. E. Goldberg and J. Richardson, "Genetic Algorithms with Sharing for Multimodal Function Optimization," pages 42–50 in *Genetic Algorithms and Their Applications: Proceedings of the Second International Conference on Genetic Algorithms*, edited by J. Grefenstette (Hillsdale, NJ: Lawrence Erlbaum, 1987).

[13] D. T. Pham and Y. Yang, "Optimization of Multi-Modal Discrete Functions Using Genetic Algorithms," *Proceedings of the Institution of Mechanical Engineers, Part D [Journal of Automobile Engineering]*, **207** (1993) 53–59.

[14] K. Deb and D. E. Goldberg, "An Investigation of Niche and Species Formation in Genetic Function Optimization," pages 42–50 in *Proceedings of the Third International Conference on Genetic Algorithms*, edited by L. Davis (San Mateo, CA: Morgan Kaufmann, 1989).

[15] M. E. Palmer and S. J. Smith, "Improved Evolutionary Optimization of Difficult Landscapes: Control of Premature Convergence through Scheduled Sharing," *Complex Systems*, **5** (1991) 443–458.

[16] D. Whitley and T. Hanson, "The GENITOR Algorithm: Using Genetic Recombination to Optimize Neural Networks," Technical Report CS-89-107, Department of Computer Science, Colorado State University (1989).

[17] D. Whitley and T. Starkweather, "GENITORII: A Distributed Genetic Algorithm," *Journal of Experimental Theoretical Artificial Intelligence*, **2** (1990) 189–214.

[18] D. Whitley, T. Starkweather, and C. Bogart, "Genetic Algorithms and Neural Networks: Optimizing Connections and Connectivity," *Parallel Computing*, **14** (1990) 347–361.

[19] D. J. Montana and L. Davis, "Training Feedforward Networks Using Genetic Algorithms," pages 762–767 in *Eleventh International Joint Conference on Artificial Intelligence (Detroit 1989)*, edited by N. S. Sridharan (San Mateo, CA: Morgan Kaufmann, 1989).

[20] A. Papoulis, *Probability, Random Variables and Stochastic Processes* (Auckland: McGraw-Hill, 1989).