

Genotypic and phenotypic assortative mating in genetic algorithm

Susmita De, Sankar K. Pal *, Ashish Ghosh

Machine Intelligence Unit, Indian Statistical Institute, 203 B.T. Road, Calcutta 700 035, India

Received 3 November 1996; received in revised form 3 June 1997; accepted 5 August 1997

Abstract

Three new methods of selection of mating pairs for Genetic Algorithms (GAs) are introduced where the partners are chosen based on either their genotypic similarity (called genotypic assortative mating) or their phenotypic similarity (called phenotypic assortative mating). These methods not only help in exploiting the current search space properly before exploring the new one but also enable one to mimic inbreeding of natural genetics. A comparative study in terms of disruption of schema due to crossover is made between these methods and conventional genetic algorithm (CGA). The superiority of this new methodology over the CGA and the incest prevention algorithm is established on some problems of optimizing complex functions and selecting optimal neural network parameters.

Keywords: Schema disruption; Inbreeding; Optimization; Neural networks

1. Motivation

Genetic Algorithms (GAs) [1–4] are adaptive and robust computational procedures modeled on the mechanics of natural genetic systems. GAs act as biological metaphor and try to emulate some of the processes observed in natural evolution. They are viewed as randomized yet structured search and optimization techniques. GAs iteratively perform the following cycle of operations on a set of coded solutions, called a *population*, until some termination condition is

achieved: *selection* (including evaluation of each solution), *reproduction* (including crossover and mutation), and *reduction/replacement* of the old population with a new one.

One of the main aspects of any search technique is to exploit properly the present search space before exploring into the new regions. In GAs, besides selection, crossover plays a significant role to perform exploitation. (Mutation, on the other hand, can be viewed as an exploration operator.) A key property of crossover operation should, therefore, be not to lose any important characteristic feature of the parent chromosomes. In GA terminology, the term 'important information' is related to *schema* [1]. Therefore, to carry this important information from one generation to the next, the disruption of schema should not be detrimental in nature. The disruption of a schema (due to crossover) depends on the structure of the chromosome pair undergoing crossover operation (mating pair) to some extent. Hence, to make crossover more effective (with respect to exploitation of search regions) there should be some criteria to choose the mating pairs, instead of choosing them randomly.

In this article we propose some methods to choose, in a controlled manner, parent chromosome pairs undergoing crossover operation. These restricted mating strategies can be viewed to emulate some of the processes of natural genetics. In natural genetics, most individuals tend to mate with others similar (closer) to it; mating between the members of a population is not random. Further, if the population size becomes restricted, mating between relatives (known as *inbreeding*) may occur [5,6]. This natural phenomena are also considered here to make GAs more closer to natural genetics. The proposed techniques will also try to exploit the current search space before exploring other regions. A comparative study is made between these new strategies and the conventional GA in terms of schema disruption due to one-point crossover. The performance of these methods is compared with that of the conventional GA and the incest prevention algorithm [7] on some problems of complex function optimization, selection of optimal set of weights in a multilayer perceptron (MLP) [8], and evolving Hopfield type optimum network architectures for object extraction [9]. Since incest prevention is a well known restricted mating strategy where Hamming distance between chromosomes is used to select mating pairs, a comparison with this is also made.

2. Selection of chromosomes for mating

The main purpose of crossover is to exchange information between two parent chromosomes with the aim of not losing any important information (minimum disruption of the coadapted sets of alleles which augment the performance of the corresponding phenotype significantly). In nature we notice that mating partners usually have similar characteristics (with respect to some

features). This motivates us to adopt the following strategies for selecting parent chromosomes for mating. In natural genetics, when matings occur preferentially between related individuals, the parents are sorted based on either their genotypic similarity (known as *genotypic assortative mating* [5,6]) or phenotypic similarity (called as *phenotypic assortative mating* [5,6]). This, in effect, may produce better rare individuals in a greater number with the help of selection. As a consequence, inbreeding has wide application for both natural and artificial selection. In this section we describe three methods for choosing mating pairs. In each case it is assumed that the chromosomes in the mating pool are sorted (in descending order) according to their fitness values. Moreover, each chromosome is selected (for mating) only once. The notation that we will be using to describe the methods is given below.

$p1$	first parent
$p2$	second parent
d	hamming distances of all other chromosomes from $p1$
$p1found$	flag variable representing whether from a given string any string with $d = 0$ is found or not
$idfound$	flag variable representing whether any string identical to a given string ($d = 0$) is found or not.

In each method the initial and terminal conditions are as follows:

Initial: the best fit string of the population is considered to be the first parent of the first mating pair.

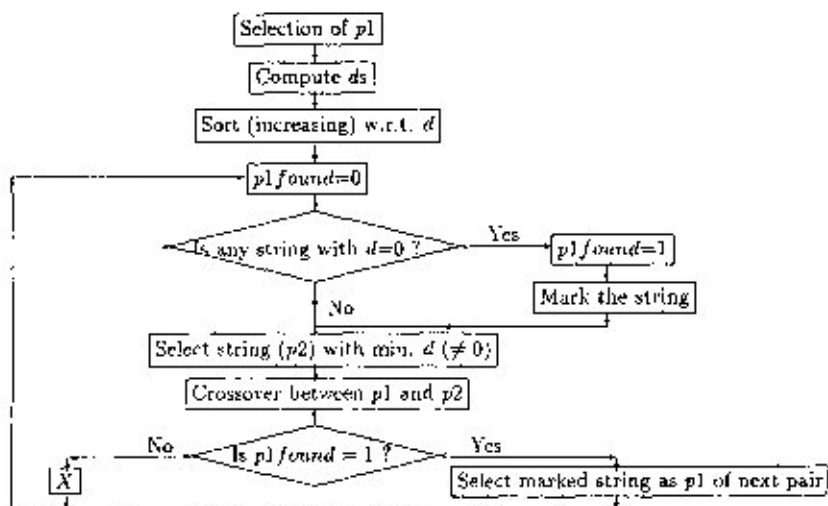


Fig. 1. Basic steps of Method 1.

Termination: the algorithm terminates when all the chromosomes in a population are selected for mating.

Method 1. The best fit chromosome is considered to be the ideal parent. Compatibility of chromosomes to be partners for crossover is determined based on their Hamming distances from this best fit string. A schematic diagram of the basic structure of Method 1 is shown in Fig. 1.

Block X: Examine string array from the beginning. First string, not yet selected for mating, is taken as $p1$ of next mating pair.

Method 2. The compatibility of two chromosomes to be partners for crossover is determined based on the Hamming distance between them. Two strings, whose Hamming distance is minimum (but not zero) with respect to all other chromosomes of the population, are chosen for mating. This selection of mating pair is performed in the following manner (Fig. 2).

Block Y: Compute Hamming distances (d 's) of all other remaining strings from this $p2$. Select the string with min. d as $p1$ of next mating pair.

Method 3. Two non-identical chromosomes (and not selected for crossover before) whose fitness values are better than the remaining chromosomes of the population are chosen for crossover operation. In other words, two parents are selected for mating whose fitness values are the highest and the second highest in the existing population (i.e., two parents are having phenotypic similarity). A schematic diagram of the basic structure of this method is shown in Fig. 3.

Block Z: Examine string array from the beginning. First string, not yet selected for mating, is taken as $p1$ of next mating pair.

Some Remarks. Note that, the first pair to be picked up for mating is identical for all the methods. Moreover in Method 1, two chromosomes with equal Hamming distance from the best chromosome may not necessarily be identical.

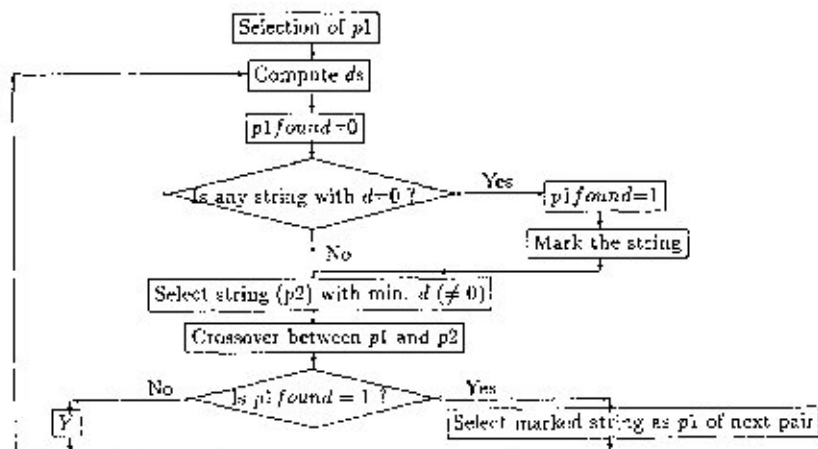


Fig. 2. Basic steps of Method 2.

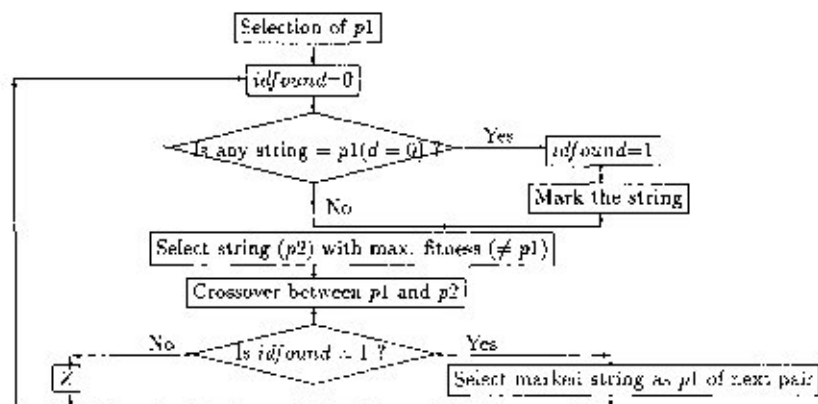


Fig. 3. Basic steps of Method 3.

It is to be mentioned here that in *incest prevention* [7] Eshelman selected the mating pairs in a way such that they are sufficiently diverse (differed by at least two bits); two parents were chosen for mating only if their Hamming distance was above a threshold. Initially, this threshold is set to a high value. If in a generation, no pair can be chosen with this threshold, the value is decreased. On the other hand, in our methods of selection of mating pairs, the pairs are selected in a manner so that they are similar in nature with respect to some features. In Methods 1 and 2 this resemblance is considered at the genotypic level by measuring Hamming distance between the chromosomes, whereas in Method 3, the similarity is considered at the phenotypic level with the help of the fitness value of the chromosomes. Note that, in both the cases two chromosomes with identical bit pattern (i.e., zero Hamming distance) are no way allowed to mate. Hence, our methods not only produce new offspring, but also mimic *inbreeding* (matings between related individuals) [5,6]. Since crossover can be viewed as an exploitation operator, our techniques try to exploit the current search region properly before exploring new ones. On the other hand, by selecting sufficiently diverse chromosomes, incest prevention algorithm tries to bring maximum diversity into the population; thereby reduces the exploiting power of crossover operator.

It is clear from Figs. 1–3 that Method 1 requires more time than Method 2, and Method 3 takes the least time. Moreover, all these methods are computationally expensive than the CGA. In incest prevention, the time complexity depends on the threshold value and the structures of the chromosomes in the population. Generally, at the starting of execution of a GA, the chromosomes are diverse and the algorithm takes less time to cross this threshold even if it is set to a high value. Whereas, at the end of the run of the GA, it may happen that no pair can cross the threshold; thereby the value is decremented in steps and all the chromosomes need to be checked at every step until at least a pair is

selected; and thus the time required will be high. Hence, it is clear that the execution time of incest prevention algorithm is variable whereas in the CGA and in the proposed restricted mating strategies it is fixed.

2.1. Analysis of schema disruption due to crossover

The Schema Theorem [1] estimates the lower bound of the number of instances of different schemata present in a population at any point of time. According to this theorem, a short-length, low-order, above-average schema will receive exponentially increasing instances in subsequent generations at the expense of below average ones. In this section we analyze the Schema Theorem for the proposed methods. We also make a comparison with that of the conventional genetic algorithm (CGA). The notations that we will be using for this purpose are as follows.

h	an above average schema
$m(h, t)$	m instances of a schema h in a population at generation t for the CGA
$\delta(h)$	the defining length of schema h
$o(h)$	order of schema h
L	length of a chromosome
\bar{f}_h	average fitness value of schema h
\bar{f}	average fitness value of the population
p_c	probability of crossover
p_m	probability of mutation

The expected number of instances of schema h at the $(t + 1)$ th generation, i.e., $m(h, t + 1)$ is

$$m(h, t + 1) \geq m(h, t) \times \frac{\bar{f}_h}{\bar{f}} \times \{1 - p_c \times x' - p_m \times y'\}, \quad (1)$$

where x' and y' , represent the probability of disruption of the schema due to crossover and mutation, respectively.

To analyze disruption of schema due to one-point crossover operation of a GA one should consider the chromosome segment between the first and the last fixed positions of the schema ($\delta(h)$) consisting of $o(h)$ number of fixed bits. Let the Hamming distance between two parent chromosomes and that within the above mentioned segment be k and k_s . The probability that the cross-site lies within this segment is $\delta(h)/(L - 1)$ and then depending on the structure of the two parent chromosomes the bits are disrupted. Let I be a random variable which denotes the number of bits disrupted. Then the probability of disruption of i bits (p_i) is given by

$$p_i = P\{I = i\} = \frac{\binom{o(h)}{i} \binom{\delta(h) - o(h)}{k_s - i}}{\binom{\delta(h)}{k_s}}, \quad (2)$$

where $i \leq \min(o(h), k_s)$. Again, $i \geq 0$ and $(\delta(h) - o(h)) \geq (k_s - i)$; so $i \geq \max(0, o(h) + k_s - \delta(h))$.

The probability of distribution defined by Eq. (2) is known as *Hypergeometric distribution*. It may be noted that (i) $p_i \geq 0$ and (ii) $\sum p_i = 1$, $i = 0, 1, \dots, \min(o(h), k_s)$. $i = 0$ represents the case of no disruption. The expected value of i , i.e., $\langle E(i) \rangle$ is [10]

$$E(i) = \frac{k_s}{\delta(h)} \times o(h), \quad (3)$$

which represents the average number of bits disrupted.

2.2. Analysis of disruption of schema due to proposed mating strategies

Genotypic assortative mating (Methods 1 and 2): Let the Hamming distance between two parent chromosomes and that within the defining length of the schema for the CGA be k' and k'_s , respectively and the corresponding values for Methods 1 and 2 be k'' and k''_s . Now, if $k'_s > k''_s$ then from Eq. (3) it is seen that the average number of bits disrupted in the CGA will be greater than that for Methods 1 and/or 2. Note that, $k' > k''$ does not necessarily mean that k'_s will always be greater than k''_s .

In Method 1, let h_1 and h_2 ($h_1 < h_2$) be the Hamming distances of the chromosomes (with respect to the best one) chosen for mating. Now these h_1 bit positions of the first parent may be fully overlapped with the h_2 bit positions of the second one, or totally exclusive with the h_2 . In the first case, the Hamming distance of one string with respect to the other one is $(h_2 - h_1)$ and in the second case it is $(h_1 + h_2)$. Hence after crossover, both the offspring may have a minimum of $(h_2 - h_1)$ to maximum of $(h_1 + h_2)$ Hamming distance apart from the best string of the previous generation.

Phenotypic assortative mating (Method 3): To analyse the disruption of schema due to Method 3, let us assume that the chosen mating pairs are the instances of schema h_1 and schema h_2 , respectively and the number of bit positions to be considered for each string (say, z) – bit values common to both h_1 and h_2 – fixed positions of h_1 (not carrying the bit values of h_2 at those positions) – fixed positions of h_2 (not carrying the bit values of h_1 at those positions). Now, the string is divided into $(z + 1)$ number of segments where each segment starts at any of the above mentioned bit position and ends just before the position of the next fixed bit. The disruption involves the position of cross-site in these segments and the bit values of the corresponding strings, and one needs to concentrate on a single segment where the cross-site lies. It can be shown that depending on the position of the cross-site and the bit values of the corresponding strings, in some cases schema h_1 and/or schema h_2 can survive.

Note: One of the key features of any crossover technique is that two low order schemata (having below average fitness values) can also combine and form a high order schema with above average fitness value.

3. Implementation

The effectiveness of the proposed concept has been demonstrated on some problems of optimizing complex functions and selecting neural network parameters.

3.1. Function optimization

Table 1 shows the functions $f1$, $f2$, and $f3$ which have varying degrees of complexity. The first one is univariate, the second one is a bivariate [2], and the third function is having P variables (n is an odd integer) [11]. The complex behavior of the functions is depicted in graphical form in Figs. 4–6. $f1$ has 10 maxima with global maximum at $x = 0$. $f2$ is a rapidly varying multimodal function with several close oscillating hills and valleys with a global maximum at $x = y = 0$. The function $f3$ has one global optimum and P^n local optima. Here we have considered $P = 5$ and $n = 5$.

To optimize these functions, the following steps are adopted. Binary coding is used for chromosomes. Substring length for each parameter (variable) is taken as 22. Population size is kept fixed at 20. The initial population is chosen randomly. Generational replacement technique is used.

The objective function is the identity function. Therefore, the higher the functional value, the better is the chromosome. Both the elitist model [2] (by copying the best member of previous generation into the present one to replace the lowest fitted string, if the fitness value of the best member of the previous generation exceeds the fitness value of the best member of the present generation) and the standard GA (i.e., non-elitist model) are implemented.

Linear normalization selection procedure (which works better in a close competitive environment) is adopted. The difference between successive fitness

Table 1
Functions used for optimization

Function	Functional form	Domain
$f1$	$\sum_{i=1}^{20} (x - 2i)$	[0, 20]
$f2$	$0.5 \cdot \left\{ \frac{\sin \sqrt{0.5 + 0.5x^2}}{1 + 0.001x^2} \right\}^{2.205}$	[-100, 100]
$f3$	$\sum_{i=1}^P 1 - \frac{\sin \frac{2\pi x_i}{1000} }{0.001} - 1$	[-0.5, 0.5]

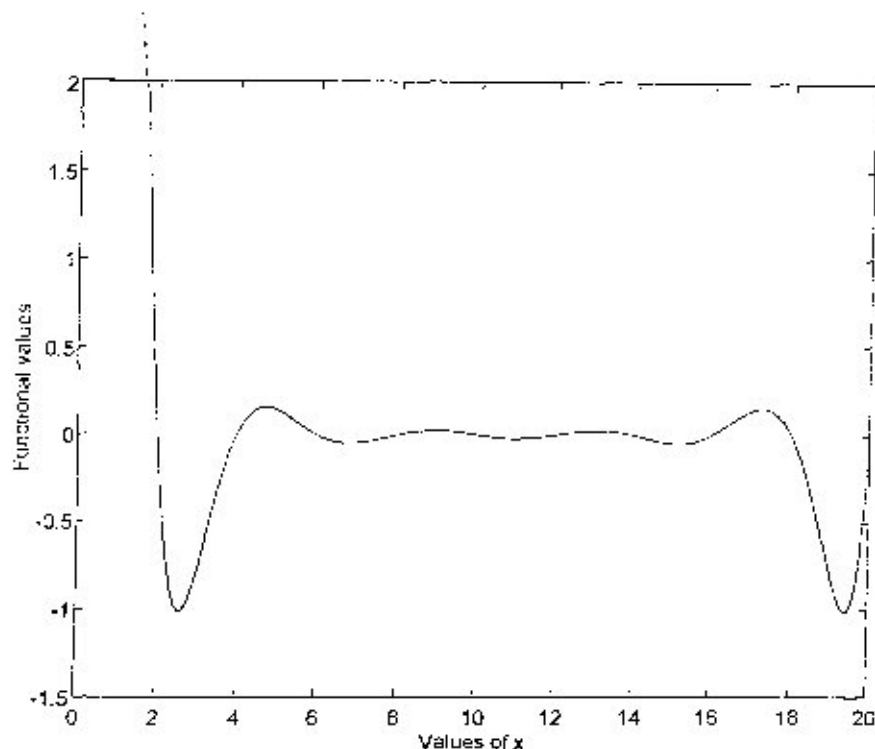


Fig. 4. Sketch of the function $\sum_{i=1}^{11} (x - 2i)$.

values and the minimum fitness value are taken as 1. Number of copies produced by the i th individual (chromosome) with normalized fitness value f_i in a population of size m is taken as round (c_i) , where

$$c_i = \frac{m \times f_i}{\sum_{i=1}^m f_i}. \quad (4)$$

Crossover probability is taken as 0.8 and two bitwise mutation probabilities are considered, namely, 0.008 and 0.05. Multi-point crossover operation is performed where for each substring (each parameter encoded as a part of the chromosome) the crossover operation is one-point. Hence, the number of cross-sites is taken to be equal to the number of parameters encoded in a chromosome. These cross-sites are chosen randomly. The algorithm has been run for 1000 generations in each simulation. The initial population is taken to be the same for all the methods. In incest prevention, the threshold value is initially set to $\frac{1}{3}m$ (where, m denotes the size of the population).

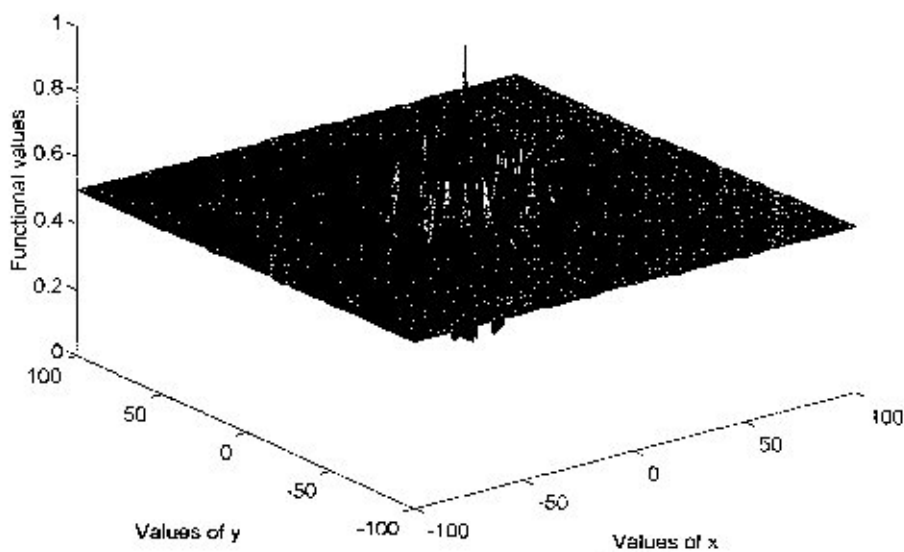


Fig. 5. Sketch of the function $0.5 - \frac{0.5 \sqrt{x^2 + y^2} + 0.5}{1 + 0.000001(x^2 + y^2)}$.

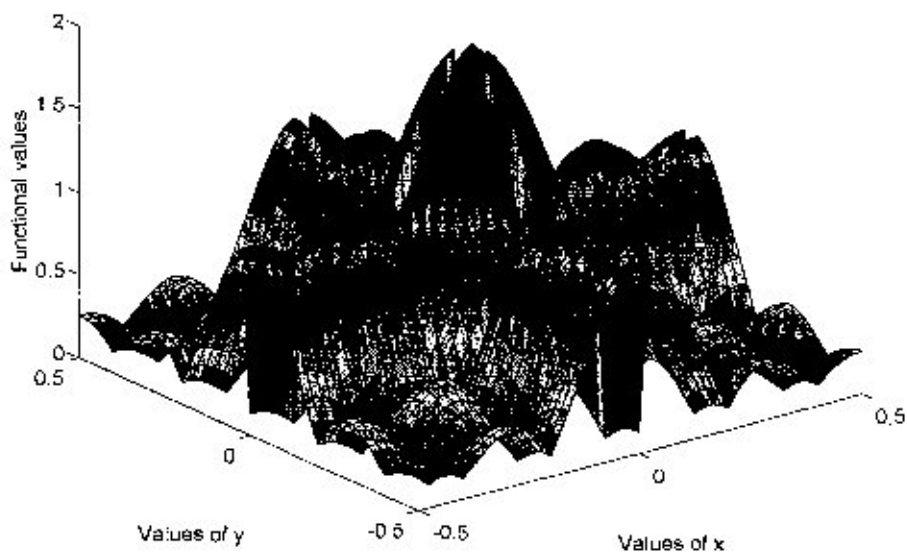


Fig. 6. Sketch of the function $\sum_{i=1}^p \frac{\sin(24 \pi x)}{125^i - 0.000001}$.

3.2. Selection of MLP parameters for an XOR problem

To determine an optimal set of connection weights and thresholds in an MLP for classification problem the overall error to be minimized is defined as [8]

$$Error = \frac{1}{s} \sum_s \sum_{j=1}^{n_{out}} (t_{sj} - V_{sj})^2, \quad (5)$$

where s and out represent, respectively, the number of training samples and the number of neurons in the output layer. t_{sj} and V_{sj} denote the target and obtained output for the j th neuron (i.e., the activation of j th output neuron) respectively, corresponding to the s th training pattern. The weights and thresholds are modified (using GAs) so that *Error* is minimized. Each neuron j in the output and hidden layers is associated with a set of p' input values I_{ij} , $1 \leq i \leq p'$, a threshold value θ_j , a set of interconnection weights w_{ij} , an activation function (which is taken as sigmoidal), and an output value

$$V_j = \frac{1}{1 + \exp(I_j w_j - \theta_j)}. \quad (6)$$

Input values are in binary form. Total number of patterns in the data set is 148. These patterns are generated by replicating the four input-output patterns of two-input XOR function 32 times. The size of the training set is considered to be 10% and 30% of the data set and these samples are taken randomly. In this problem, there are two neurons in the hidden layer. Since the problem is a two-class one, number of neurons in the output layer is 2. Hence, total number of parameters of the problem (including threshold values of the neurons of hidden layer and output layer) = 12. Note that the number of parameters to be determined here is more than that of the function optimization problem (Table 1). Hence due to space limitations, each parameter (w_{ij}, θ_j) is encoded using 10 bits only. Value of these parameters lie in $[-25, +25]$. Each string, thus represents, a set of thresholds and weights of a complete network. The objective function to be minimized is the error value. The lower this value is, the higher is the fitness. As in the case of function optimization problem, other parameters such as the population size, crossover and mutation probabilities, selection criteria etc. remain the same. The algorithm is run for 1000 generations with a single initial population set.

3.3. Selection of Hopfield type network architecture for object extraction

For the problem of determining the optimum architecture of Hopfield type neural network for object extraction, we considered a noisy image as input. This noisy version is generated by adding $N(0, \sigma^2)$ noise to each pixel of the synthetic binary (two-tone) image shown in Fig. 7. The size of the image is

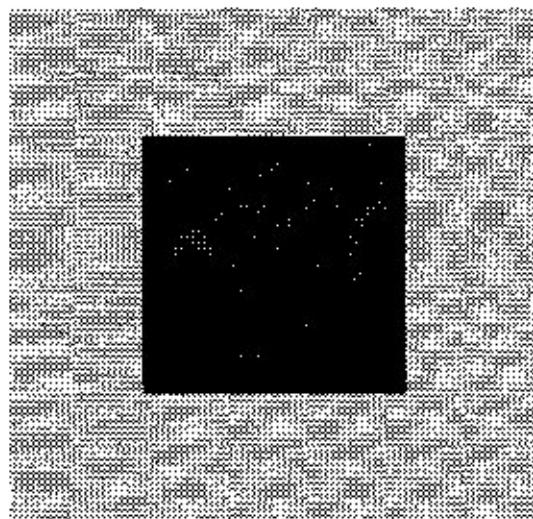


Fig. 7. Original synthetic image.

40×40 . For an $m1 \times n1$ image, each pixel (neuron) being connected to at most $k1$ of its neighbors, the length of a chromosome is $m1 \times n1 \times k1$. If a neuron is connected to any of its neighbors, the corresponding bit of the chromosome is set to 1, else 0. Hence for an image of size 40×40 , a binary string of length $40 \times 40 \times 8$ (here we consider eight neighbors for a pixel) has been used for chromosome representation. Each string represents a possible network architecture for object extraction. Since the number of parameters to be determined here is very large, we considered population size of 30 which is larger than that used in the previous two problems. Fitness of a chromosome is taken as a function of the energy value (Appendix A) of the (converged) network. The lower the energy value, the better is the chromosome. Unlike the previously mentioned two experiments, mutation probability is chosen as 0.002 (lower than the above two cases). Crossover probability and selection procedure are the same as in previous experiments.

4. Analysis of results

The mean (averaged over 20 simulations) fitness values of the best chromosomes at the last generation using the CGA (denoted as Method 0), proposed methods (Methods 1, 2, and 3), and incest prevention (denoted as Method 4) for the functions $f1$, $f2$, and $f3$ (Table 1) are shown in Tables 2 and 3 corresponding to mutation probabilities 0.008 and 0.05 (NE and E denote, respectively, the non-elitist model and the elitist model) which provides a

comparative study among different methods based on the performance attained at the end of evolutionary process. Table 4 illustrates the relative merit of different methods based on the performance (in terms of average fitness value of the population) at every generation using non-elitism for the functions $f1$, $f2$, and $f3$. The meaning of each symbol, used in this table, is given below:

$A = B$: Performances of Schemes A and B are the same for every generation.

$A > B$: Either performance of Scheme A is better than Scheme B at every generation (Fig. 8), or performance of Scheme A is ultimately better than Scheme B irrespective of its performance at earlier generations (Fig. 8).

$A \geq B$: Performance of Scheme A is better than Scheme B up to some generations, and ultimately they become equal (Fig. 8).

Table 2

Maximum fitness value (averaged over 20 simulations) with mutation probability 0.008

Method	Function $f1$		Function $f2$		Function $f3$	
	NE	E	NE	E	NE	E
0	3.16×10^9	3.16×10^9	0.956921	0.970995	26.456	28.071
1	3.16×10^9	3.34×10^9	0.982783	0.980223	26.045	25.642
2	3.34×10^9	3.34×10^9	0.974529	0.973191	26.196	29.691
3	3.34×10^9	3.53×10^9	0.976924	0.978891	29.368	31.715
4	3.16×10^9	3.16×10^9	0.974162	0.965709	25.370	28.071

Table 3

Maximum fitness value (averaged over 20 simulations) with mutation probability 0.05

Method	Function $f1$		Function $f2$		Function $f3$	
	NE	E	NE	E	NE	E
0	3.71×10^9	3.72×10^9	0.868921	0.992227	4.376	16.045
1	3.71×10^9	3.72×10^9	0.920388	0.992227	4.561	19.768
2	3.71×10^9	3.72×10^9	0.916514	0.993199	4.435	26.170
3	3.71×10^9	3.72×10^9	0.874062	0.994170	4.469	19.432
4	3.71×10^9	3.72×10^9	0.818443	0.992227	4.465	19.399

Table 4

Comparison of different methods (for non-elitism) based on average fitness value of the population

Mutation probability	Function $f1$	Function $f2$	Function $f3$
0.008	$3 \geq 2 > 1 \approx 4 \geq 0$	$1 \geq 2 \approx 3 \approx 4 > 0$	$3 > 2 > 0 \geq 1 > 4$
0.05	$2 > 1 > 0 \approx 3 \approx 4$	$2 \geq 1 \approx 3 > 0 \approx 4$	$3 > 1 > 0 \approx 2 \approx 4$

$A \approx B$: Performance of Schemes A and B are very similar, but not identical (Fig. 8).

It is seen from Tables 2-4 that the CGA (Method 0) performs the worst in all the cases than our methods except three cases in Table 2 (e.g., $f3$, NE, Methods 1 and 2 and $f3$, E, Method 1) and one case in Table 4 (e.g., with mutation probability = 0.008 in $f3$, Method 1). Again, these methodologies always performed better than the incest prevention algorithm (Method 4) except one case in Table 2 (e.g., $f3$, E, Method 1) and another case in Table 3 (e.g., $f3$, NE, Method 2).

For clarity of presentation, the variation of average fitness value with generations using the non-elitist version of different methods for the function $f3$ with mutation probability 0.008 is shown in Fig. 9. It is evident from the tables and the figures that the proposed methodology for selecting mating pairs improves the performance.

Table 5 shows the error values of the best chromosomes in the last generation using various methods for the problem of selecting MLP parameters. The rate of mutation is considered to be 0.008. Results show that our methods perform better than the CGA for all the cases. They are also better than the incest prevention method except for elitism with Method 1 (using 30% training samples).

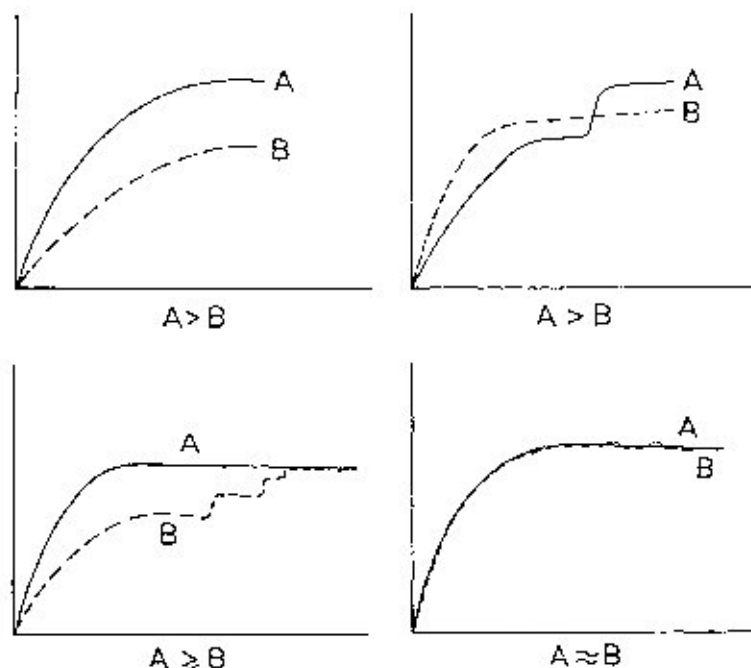


Fig. 8. Figures illustrating the notations $A > B$, $A \geq B$, and $A \approx B$.

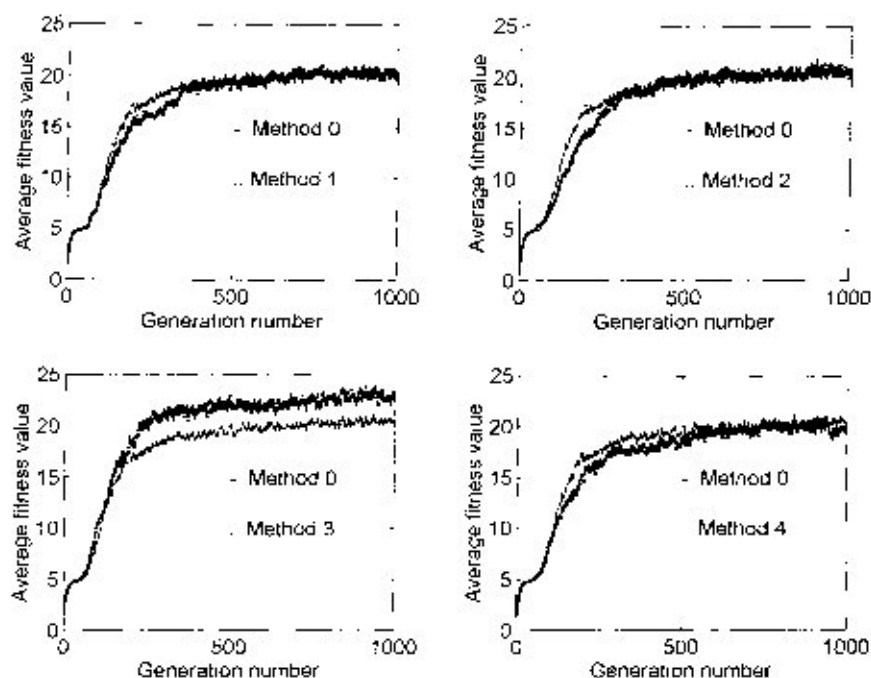


Fig. 9. Variation of average fitness value with generations for the function f_3 using the non-elitist model with mutation probability 0.008 for (a) the CGA and Method 1; (b) the CGA and Method 2; (c) the CGA and Method 3; (d) the CGA and Method 4 (incest prevention).

For the problem of extracting object regions from noisy images using Hopfield type neural networks, we have demonstrated in Fig. 10 the variation of average fitness (energy) value of the network with generations using non-elitism for $\sigma = 20$. Fig. 10(a) provides the comparison only among those methods (Methods 1, 2, 4) where the mating partners are selected using Hamming

Table 5

Minimum error value for MLP problem with mutation probability 0.008

Method	Training sample used			
	10%		30%	
	NE	E	NE	E
0	2.000553	5.000072	13.2746	8.1009
1	0.000000	5.000016	12.0005	7.9998
2	0.000077	3.464297	6.9699	0.0003
3	0.000000	4.498895	2.6235	0.0000
4	2.000061	4.498895	14.7916	6.9699

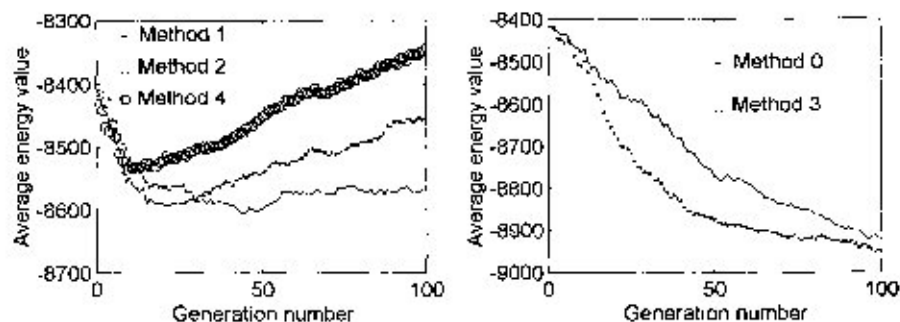


Fig. 10. Variation of average fitness value with generations using the non-elitist model for noisy image with $\sigma = 20$ for (a) Methods 1, 2, and 4; (b) the CGA and Method 3.

distances between them. It is seen from this figure that both Methods 1 and 2 are better than incest prevention. From Fig. 10(b), where the relative performance of the CGA and Method 3 is depicted, it is also found that Method 3 is better than the CGA.

Note that, although we have considered binary coded representation of chromosomes, the proposed methods will remain valid for other types of coding.

5. Conclusions

Three strategies for selecting mating pairs of GAs, instead of choosing them randomly, have been described which try to mimic inbreeding of natural genetics. The criterion of selecting mating pairs is based either on their genotypic similarity or on their phenotypic similarity. These strategies enable one to exploit the current search region extensively before exploring new ones. An analysis for the disruption of schema due to these new strategies is provided along with a comparison with the disruption of schema by the conventional genetic algorithm (CGA). The effectiveness of this methodology over the CGA and incest prevention is demonstrated experimentally on some problems of complex function optimization and network parameter selection. It is found that our methods produce better result than the CGA in most of the cases. While Method 1 is seen, sometimes, to be inferior to incest prevention algorithm, Method 3 always comes out to be the best.

Acknowledgements

The authors gratefully acknowledge Mr. S.K. Mitra for his helpful discussion. A part of this work was done when Ms. Susmita De held a research fellowship of the Department of Atomic Energy, Government of India.

Appendix A. Hopfield type neural network architecture for object extraction

To use a Hopfield type neural network for object background classification [12], a neuron is assigned corresponding to every pixel. Each neuron can be connected to its neighbors (over a window) only. The connection can be full (a neuron is connected with all of its neighbors) or can be partial (a neuron may not be connected with all of its neighbors). The network topology for a fully connected third order neighborhood is depicted in Fig. 11. Here the maximum number of connections of a neuron with its neighbors is 8. In practice, all these connections may not exist. Again, different neurons may have different connectivity configuration within its neighbors. The initial status and input bias of each neuron are set depending on the gray value of the corresponding pixel. The status updating rules are similar to those of Hopfield's model. The objective function to be minimized for object extraction is similar to the expression of energy of the above mentioned network.

The energy function of this model has two parts. The first part is due to the local field or local feedback and the second part corresponds to the input bias of the neurons. In terms of images, the first part can be viewed as the impact of the gray levels of the neighboring pixels, whereas the second part can be attributed to the gray value of the pixel under consideration. The total energy contributed by all pixel pairs will be $\sum_i \sum_j W_{ij} S_i S_j$, where S_i, S_j are the status of

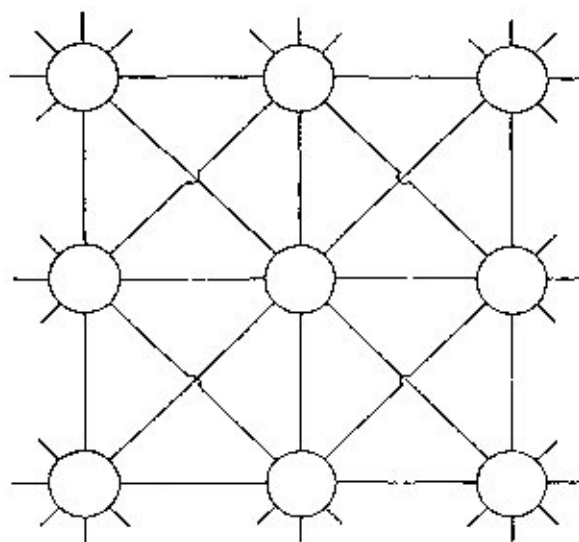


Fig. 11. Topology of the neural network with third order connectivity (in the proposed system all connections may not exist).

i th and j th neurons, respectively and W_{ij} is the connection strength between these two neurons.

For every neuron i , the initial input bias B_i and the initial state S_i are taken to be proportional to the actual gray level for the corresponding pixel. If a gray level of a pixel is high (low), the corresponding intensity value of the scene is expected to be high (low). The input bias value is taken in the range $[-1, 1]$. Under this framework an ON (1) neuron corresponds to an object pixel and the OFF (-1) one as background pixel. So the threshold between object and background can be taken as 0. Thus the amount of energy contributed by the input bias values is $-\sum_i B_i S_i$. So the expression of energy for the object extraction problem takes the form

$$\text{Energy} = -\sum_i \sum_j W_{ij} S_i S_j - \sum_i B_i S_i \quad (7)$$

Stable states of the network (the local minima of the energy function) are assumed to correspond to the partitioning of a scene into compact regions. So from a given initial state, the status of a neuron is modified iteratively to attain a stable state.

References

- [1] D.E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Addison Wesley, MA, 1989.
- [2] L. Davis (Ed.), Handbook of Genetic Algorithms, Van Nostrand Reinhold, New York, 1991.
- [3] Z. Michalewicz, Genetic Algorithms + Data Structure = Evolution Programs, Springer, Berlin, 1994.
- [4] M. Mitchell, An Introduction to Genetic Algorithms, The MIT Press, Cambridge, MA, 1996.
- [5] M.W. Strickberger, Genetics, Prentice-Hall of India, New Delhi, 1985.
- [6] E.J. Gardner, M.J. Simmons, D.P. Snustad, Principles of Genetics, Wiley, New York, 1991.
- [7] L.J. Eshelman, Preventing premature convergence in the genetic algorithms by preventing incest, in: R.K. Belew, J.B. Booker, (Eds.), Proceedings of the Fourth International Conference on Genetic Algorithms, San Diego, July 1991, pp. 115–122.
- [8] D. Whitley, T. Starkweather, C. Bopart, Genetic algorithms and neural networks: Optimizing connections and connectivity, Parallel Computing 14 (1990) 347–261.
- [9] S.K. Pal, S. De, A. Ghosh, Designing Hopfield type networks using genetic algorithms and its comparison with simulated annealing, International Journal of Pattern Recognition and Artificial Intelligence 11 (1997) 447–461.
- [10] V.K. Rohatgi, An Introduction to Probability Theory and Mathematical Statistics, Wiley Eastern, New Delhi, 1986.
- [11] M. Srinivas, L.M. Patnaik, Adaptive probabilities of crossover and mutation in genetic algorithms, IEEE Transactions on Systems, Man and Cybernetics 24 (1994) 656–667.
- [12] A. Ghosh, N.R. Pal, S.K. Pal, Object background classification using Hopfield type neural network, International Journal of Pattern Recognition and Artificial Intelligence 6 (1992) 989–1008.