

Genetic algorithms for optimal image enhancement

Sankar K. Pal *, Dinabandhu Bhandari, Malay K. Kundu

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

Received 26 October 1992; revised 23 August 1993

Abstract

Pal, S.K., Genetic algorithms for optimal image enhancement, Pattern Recognition Letters 15 (1994) 261–271.

Genetic algorithms represent a class of highly parallel adaptive search processes for solving a wide range of optimization and machine learning problems. The present work is an attempt to demonstrate their adaptivity and effectiveness for searching global optimal solutions in selecting an appropriate image enhancement operator automatically.

Keywords. Pattern recognition, image enhancement, genetic algorithms, ambiguity measures.

1. Introduction

Genetic algorithms (GAs) [2,4] are search algorithms based on the mechanics of natural selection and natural genetic systems. They are highly parallel and adaptive. They combine survival of the fittest among string structures with a structured, yet randomized, information exchange to form a search technique with some of the innovative flair of human search. They efficiently exploit historical information to speculate on new search points with expected improved performance using genetically inspired operators on potential solutions in an iterative fashion. GAs deal simultaneously with multiple points and not a single point unlike conventional search techniques. GAs are theoretically and empirically proven to provide robust search in complex spaces, even if the search (e.g., optimization) function spaces are not smooth or continuous, which are very difficult (sometimes impossible) to search using calculus-based methods. GAs are also blind, that is, they use

only the payoff or penalty (i.e., objective) function and do not need any other auxiliary information. Recently, GAs are finding widespread applications in solving problems, requiring efficient and effective search, in business, scientific and engineering circles.

There are many problems in the area of pattern recognition and image processing [1,8,9] where we need to perform efficient search in complex spaces in order to achieve an optimal solution. Let us consider, for example, the problem of contrast enhancement of an image by gray-level modification. Not every kind of nonlinear function will produce a desired (meaningful) enhanced version [3]. Given an image, it is difficult to select a functional form which will be best suited without prior knowledge of the image statistics. Even if we are given the image statistics it is possible only to estimate approximately the function required for enhancement, and the selection of the exact functional form still needs human interaction in an iterative process.

The present article attempts to demonstrate the suitability of GAs in the automatic selection of an image enhancement operator for an unknown image.

* Corresponding author. Email: sankar@isical.ernet.in

The problem is to select automatically an optimum set of 12 parameter values of a generalized enhancement function, that maximizes some fitness function. The algorithm uses both spatial and grayness ambiguity measures as the fitness value. Multiple point genetic cross-over operations have been used for better convergence. Gradual convergence of the enhancement function, enhanced output and fitness value to their optimal states is experimentally demonstrated for images having both compact and elongated objects (bimodal as well as multimodal). Basic principles and key features of GAs are outlined in Section 2. The problem of selecting an image enhancement operator and the relevance of GAs are explained in Section 3. Section 4 describes the algorithm we developed, and Section 5 presents the results.

2. Genetic search algorithm: basic principles and features

To solve an optimization problem, GAs start with the chromosomal (structural) representation of a parameter set $\{x_1, x_2, \dots, x_p\}$. The parameter set is to be coded as a finite-length string over an alphabet of finite length. Usually, the chromosomes are strings of 0's and 1's. For example, let $\{a_1, a_2, \dots, a_p\}$ be a realization of the parameter set and the binary representation of a_1, a_2, \dots, a_p be 10110, 00100, ..., 11001 respectively. Then the string 10110 00100 ... 11001 is a chromosomal representation of the parameter set $\{a_1, a_2, \dots, a_p\}$. It is evident that the number of chromosomes (strings) is 2^l where l is the string length. However, there are other representation schemes [2] such as ordered lists for bin-packing and embedded lists for scheduling problems.

GAs find the global near-optimal solution employing three basic operations over a limited number of strings. The operators are (i) Reproduction/Selection, (ii) Cross-over and (iii) Mutation.

Reproduction is a process in which individual strings are copied according to their objective function values, F , called the fitness function. This operator is an artificial version of natural selection, a Darwinian survival of the fittest among string creatures. More highly fitted strings have a higher number of offspring in the succeeding generation. These

strings are then entered into a mating pool, a tentative new population, for further genetic operator action.

Unlike in biological systems¹, the cross-over generates offspring for the new generation using the highly fitted strings (parents) selected randomly from the mating pool created by the reproduction operation. The cross-over may proceed in two steps. First, members of the reproduced strings in the mating pool are mated at random. Second, each pair of strings undergoes crossing over as follows: an integer position k is selected uniformly at random between 1 and $l-1$, where l is the string length greater than 1. Two new strings are created by swapping all characters from position $k+1$ to l . Let

$$a = 11000 \ 10101 \ 01000 \ \dots \ 01111 \ 10001 \ ,$$

$$b = 10001 \ 01110 \ 11101 \ \dots \ 00110 \ 10100$$

be two strings (parents) selected for the crossing-over operation and the generated random number be 11 (eleven). Then the newly produced offspring (swapping all characters after position 11) will be

$$a' = 11000 \ 10101 \ 01101 \ \dots \ 00110 \ 10100 \ ,$$

$$b' = 10001 \ 01110 \ 11000 \ \dots \ 01111 \ 10001 \ .$$

In the simple GA, mutation is the occasional (with small probability) random alteration of the value of a string position. The mutation operator plays a secondary role in the simple GA. Note that the frequency of mutation to obtain good results in the empirical genetic algorithm studies is of the order of one per thousand. It helps to prevent the irrecoverable loss of potentially important genetic material. A random bit position of a random string is selected and is replaced by an another character from the alphabet. For example, let the third bit of string a , given above, be selected for mutation. Then the transformed string after mutation will be

$$a = 11100 \ 10101 \ 01000 \ \dots \ 01111 \ 10001 \ .$$

GAs are different from the traditional search techniques used in the calculus-based approach, dynamic programming and simulated annealing. They are dif-

¹ In biological systems, cross-over occurs on two heterogamous chromosomes.

ferent from most of the normal optimization and search procedures in four ways:

- GAs work with the coding of the parameter set, not with the parameters themselves.
- GAs search from a population, not from a single point.
- GAs search via sampling, a blind search.
- GAs search using stochastic operators, not deterministic rules.

Coding the parameter set enhances the search space extensively. Strings of length l explore α^l search points, where α is the size of the alphabet. In particular, when the strings are represented in binary coded form then for $l=3$ GAs explore $2^3=8$ different points (e.g. 000, 001, 010, 011, 100, 101, 110, 111) from three strings 100, 010 and 001 using three basic operators.

GAs deal with multiple points and not a single point unlike usual calculus-based techniques. Using the latter type of techniques, there is a high possibility to end up in a local optimum when the search space (Figure 1(a)) has multiple peaks. The performance of calculus-based methods depends highly on the initial choice. They seek optima in the neighborhood of the current point. On the other hand, GAs exploit si-

multaneously many points (a population of points) in parallel.

Genetic algorithms achieve much of their performance by ignoring information except that concerning payoff (survival ability). Other methods rely heavily on such information (for example, calculus-based techniques need derivatives, the search techniques in combinatorial optimization require access to all the entries of the concerned incidence matrix) and they fail in problems where the necessary information is not available or difficult to obtain. Figure 1(b) shows a typical example of a search space where the optimization function is not differentiable at all points and therefore the calculus-based techniques fail to find the optima. GAs, with high probability, are able to successfully detect the global near-optimal solution by exploiting information available in any search problem.

The transition rules of GAs are stochastic, whereas many other methods have deterministic transition rules. Moreover, the distinction between the randomized operators in GAs and other methods that are simple random walks is that GAs use random choice to guide highly exploitative search.

In this note, we propose to demonstrate the capability of GAs in handling complex optimization problems with an application to image enhancement.

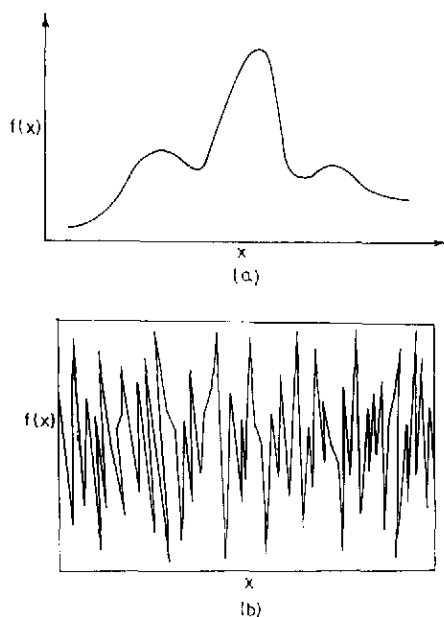


Figure 1. (a) A multiple-peak function. (b) Function unsuitable for search by calculus-based methods.

3. Selection of automatic image enhancement operators

The purpose of image enhancement is to improve the picture quality, more specifically, to improve the quality for visual judgment and/or machine understanding. The utility of image enhancement in computer-vision problems has been sufficiently demonstrated in the literature [7].

Let us consider the case of contrast enhancement by gray-level modification. Here the problem is to select an appropriate transformation/mapping function (operator) for obtaining a desired output. Usually, a suitable nonlinear functional mapping is used to perform this task.

It is to be noted that the process of evaluation of the quality of an image (picture) is subjective, which makes the definition of a well-processed image an illusive standard for comparison of algorithm per-

formance. To make this task objective it is necessary to define an objective function which will provide a quantitative measure for enhancement quality.

Some of the most commonly used transformation functions [3,5] are shown in Figures 2(a-d). The mapping function (f_1) depicted in Figure 2(a) increases the contrast within the darker area of the image, while the application of a function (f_3) as in Figure 2(c) will produce effects exactly opposite to that of Figure 2(a). The function (f_2) shown in Figure 2(b) will result in stretching of the middle range gray levels and the function (f_4) in Figure 2(d) will drastically compress the middle range values, and at the same time it will stretch the gray levels of the upper and lower ends. The mathematical forms of the above mentioned mapping functions are given below.

$$f_1(x) = \frac{Ax^2}{1+Ax^2} = \frac{x^2}{par_1 + x^2} \tag{1}$$

or, alternatively,

$$f_1(x) = par_1 \log(x) \tag{2}$$

where par_1 and A are positive constants.

The function in Figure 2(b) is represented by

$$f_2(x) = \left[1 + \left(\frac{x_{max} - x_{min}}{par_6} \right)^{par_7} - 1 \right]^{-1} \tag{3}$$

where x_{min} and x_{max} are the minimum and maximum gray levels in the image.

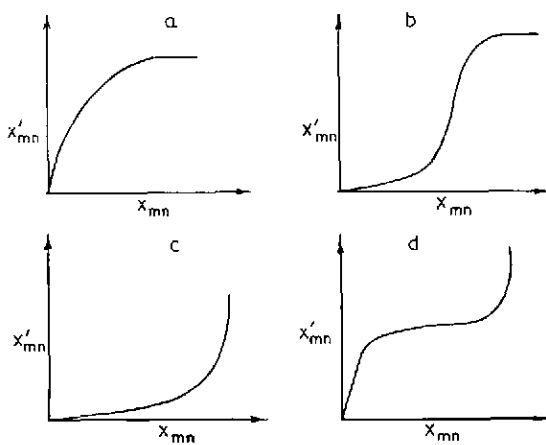


Figure 2. Mapping functions commonly used for image enhancement.

$$f_3(x) = par_2 [G(x)]^2 + par_3 x + par_4, \tag{4}$$

$$0 < par_2, par_3, par_4 < 1$$

where

$$G(x) = x - par_5 \quad \text{for } x > par_5,$$

$$= 0 \quad \text{otherwise,}$$

$$x_{min} < par_5 < x_{max}. \tag{5}$$

Finally, the function in Figure 2(d) is given as:

$$f_4(x) = \frac{1}{x_{max}} \times \left[x_{max} - par_6 \left\{ \left(\frac{x_{max}}{x} + par_8 \right) - 1 \right\}^{-par_7} \right] \tag{6}$$

where par_6 and par_7 are positive constants and par_8 is the value of $f(x)$ for $x=0$.

All these functions perform contrast enhancement of an image. Note that not all nonlinear functions will produce desired (meaningful) enhanced versions [3] of a particular image. The questions that naturally arise are "Given an arbitrary image, which type of nonlinear functional form will be best suited without prior knowledge of image statistics (e.g., in robot vision and remote applications where frequent human interaction is not possible) for highlighting its object?" and "Knowing the enhancement function, how can one quantify the enhancement quality?". Regarding the first question, even if we are given the image statistics, it is possible only to estimate approximately the function required for enhancement and the selection of the exact functional form still needs human interaction in an iterative process. The second question, on the other hand, needs individual judgment which makes the optimal decision subjective. This issue, in the context of a quantitative evaluation function will be discussed in Section 4.3.

Since we do not know the exact function which will be suited for a given image, it seems appealing and convenient to use one general functional form which will yield the four functions mentioned above as special cases and possibly others. As an illustration one may consider a convex combination of these four functions e.g.,

$$f(\cdot) = par_9 f_1(\cdot) + par_{10} f_2(\cdot) + par_{11} f_3(\cdot) + par_{12} f_4(\cdot) \tag{7}$$

under the constraint:

$$par_9 + par_{10} + par_{11} + par_{12} = 1 .$$

Here, the multipliers ($par_9, par_{10}, par_{11}, par_{12}$) are to be chosen according to the importance (suitability) of a function for a given image. On the other hand, parameters ($par_1, par_2, \dots, par_8$) of the respective functions, are to be defined according to the quality of enhancement desired. It may be noted that this combination will enable one to stretch/compress any region of an image one may desire. Therefore, the first question above boils down to determining an optimum set of values of these 12 parameters in order to achieve a desired enhancement.

In the following section we will explain the application of GAs in determining the optimum parameter set (i.e., the exact functional form) for enhancement of an image.

4. A GA for automatic image enhancement

Let us consider here the problem of enhancement by gray-level rescaling. In gray-level rescaling, each pixel is directly quantized to a new gray level in order to improve the contrast of an image. The simplest form of the functional mapping may be expressed as

$$x'_{mn} = x_{max} \cdot f(x_{mn}) \tag{8}$$

where x_{mn} = gray value of the (m, n)th pixel in the input image (original), x'_{mn} = transformed value of the (m, n)th pixel (enhanced), $f(x)$ is the prescribed transformation function defined in Eq. (7), and x_{max} = maximum value of gray-level dynamic range.

The main steps in solving a problem using GAs are:

1. Chromosomal representation of the solution (parameter set) of the problem.
2. Creation of an initial population of possible solutions.
3. Development of an evaluation (fitness) function that plays the role of the environment, rating solution in terms of their 'fitness'.
4. Definition of genetic operators that alter the composition of children during reproduction.
5. Establishing values for the parameters (popula-

tion size, probabilities of applying genetic operators) that the genetic algorithm uses.

In the following subsections we explain the way of formulating these five components in the context of automatic image enhancement.

4.1. Representation of parameters

Let $f(\cdot)$ be a transformation function having p parameters $par_1, par_2, \dots, par_p$. These parameters may have different domains [$par_{i,min}, par_{i,max}$], $i = 1, 2, \dots, m$. A binary string of length pq can be considered as a chromosomal representation of the parameter set. Here each substring of length q is assumed to be the representative of each parameter. For example, let us consider Eq. (7). There are twelve parameters including eight ($par_1, par_2, \dots, par_8$) for four basic functions (f_1, f_2, f_3 and f_4) and four ($par_9, par_{10}, par_{11}$ and par_{12}) for multipliers. Note that the domains of those parameters are different. A binary string of length 10 (e.g., 1011000101) is used for each parameter. So a chromosome for 12 parameters will be of length 120 such as

$$\begin{matrix} 1100010101 & 0100011010 & \dots & 0111110001 \\ par_1 & par_2 & \dots & par_p \end{matrix}$$

After applying the genetic operators (as discussed in Section 4.4) the resulting substrings of length q are decoded into a real number v_i ; $i = 1, 2, \dots, p$; $0 \leq v_i \leq 1$. For example, if $b_1 b_2 \dots b_q$ ($b_i \in \{0, 1\}$) is a binary representation of par_i then its decoded version is $\sum_{i=1}^q b_i 2^{-i}$. This will result in a maximum error of the order of $\sum_{i=q+1}^{\infty} 2^{-i} = 2^{-q}$. These parameter values are then linearly transformed to make them lie into their respective domains in order to be used for computing the enhanced version. In our application, the domains of the different parameters are given below, where $gray_{min}$ and $gray_{max}$ are the minimum and maximum value of the gray-level dynamic range.

Parameters	Range
$par_1 - par_4$ & $par_6 - par_{12}$	[0,1]
par_5 & par_6	($gray_{min}, gray_{max}$)
par_7	[1,3]

4.2. Creation of initial population

GAs search for the global, near-optimal solution under complete lack of knowledge about the search spaces. Usually in GAs, the initial population consists of entirely random strings (chromosomes). However, random binary strings, each of length pq (q bits for each of the parameters) can be considered as chromosomes for individuals of the initial population.

4.3. Evaluation function

As mentioned in Section 3, we need an evaluation function for quantifying the desired enhanced output, i.e., to objectivate the subjective evaluation. The present algorithm considers this evaluation function as the fitness function of GA. Here we have used entropy (H), compactness ($COMP$), index of area coverage ($IOAC$) and their combinations as the quantitative indices for evaluating picture quality, as they have been successfully used as grayness and spatial ambiguity measures for image enhancement and segmentation problems. Their definitions may be found in [5,6].

Entropy of an image (X) considers the global information and provides an average amount of fuzziness in grayness of X , i.e., the degree of difficulty (ambiguity) in deciding whether a pixel would be treated as black (dark) or white (bright). Compactness and $IOAC$ on the other hand, take into account the local information and reflect the amount of fuzziness in shape and geometry (spatial domain) of an image. Therefore, the concept of minimization of these ambiguity measures may be considered as the basis of a fitness (evaluation) function. One may also use a composite measure (e.g., product of both grayness and spatial ambiguity measures) as the evaluation function so that minimization of this composite measure implies achieving minimum ambiguity (fuzziness) in an image from both points of view.

4.4. Genetic operators

The reproduction process is executed (as described in Section 2) by copying the individual strings, according to their fitness function values, into the mating pool for the purpose of cross-over and mutation operations.

Since the size of the parameter set for this image enhancement problem is not small, it is intuitive that the single point cross-over operation (as described in Section 2) may not be useful for fast convergence. Therefore, instead of applying the cross-over operation at a single point over the entire string, we applied this (multiple cross-over) operation on each substring (chromosome representation of a parameter). The proposed cross-over operation is demonstrated below for a substring length $q=10$.

Let

$$a = 1100010101\ 0100011010 \dots 0111110001 ,$$

$$b = 1000101110\ 1110110001 \dots 0011010100$$

be two strings (parents) selected for crossing over. Let the random numbers generated by the cross-over operation be 7, 5, ..., 4. Then the newly produced offspring will be

$$a' = 1100010110\ 0100010001 \dots 0111010100 ,$$

$$b' = 1000101101\ 1110111010 \dots 0011110001 .$$

A random bit position of a random string is selected for mutation and then a random number (rm) is generated in $[0, 1]$. If rm is smaller than a predefined value (0.001, say) then the bit is replaced by another character from the alphabet (e.g., 0 is replaced by 1 and vice versa).

Following the concept of De Jong's *elitist model* [4], the best fit string obtained so far is included in the new population. It helps to preserve the best structure in the population.

4.5. Domains of parameters

The parameters to be specified in defining a GA for a specific problem, are (i) the population size, i.e., the number of chromosomes in each generation; (ii) the number of generations to be generated; (iii) the probability of mutation.

Depending on the problem, one has to choose these parameter values. In our experiment we considered

$$\text{size of the population} = 100 ,$$

$$\text{number of generations} = 30 ,$$

$$\text{probability of mutation} = 0.001 .$$

5. Implementation and results

The block diagram of the proposed algorithm is given in Figure 5.

Two images, one of them (blurred chromosome in Figure 3) having bimodal histograms and Lincoln in Figure 4 having multimodal histogram, have been considered here to test the proposed algorithm. The function $f(\cdot)$ (Eq. (7)) having 12 parameters is taken as the transformation or mapping function for gray-level rescaling, and the composition (product) of the measures entropy, compactness and IOAC of the images is used as the evaluation function. We have used the standard S -function of Zadeh to calculate μ_{mn} . The

reciprocal of the evaluation function is considered to be the fitness function. An initial population is generated with 100 random binary strings of length 120 (10 bits for each parameter).

The intermediate and the final configurations of the transformation functions (e.g., functional forms after 15, 20, 25 and 30 generations) for the blurred chromosome and the corresponding enhanced outputs are shown in Figures 3(c-d), when the product of entropy and compactness is taken as the fitness function. This shows how this GA makes the enhancement function gradually converge to the optimal state. This is also apparent from the successive improvement (Figure 3(d)) in the enhanced versions of the

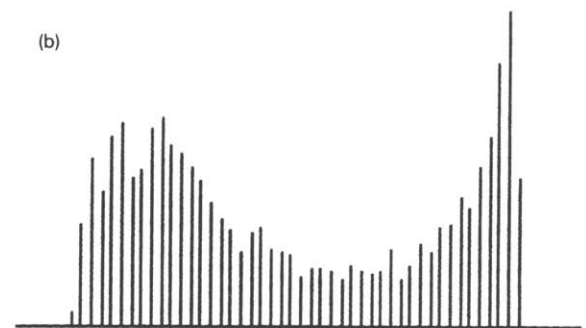
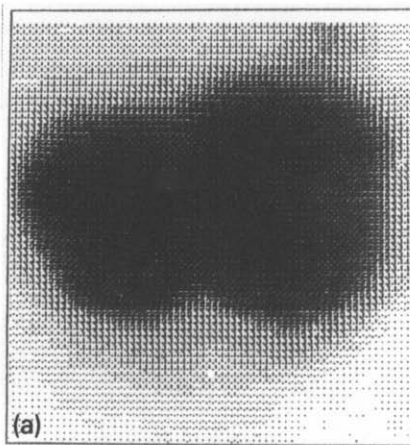


Figure 3. Blurred chromosome. (a) Input image. (b) Histogram.

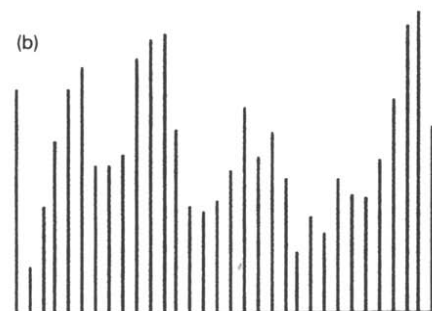
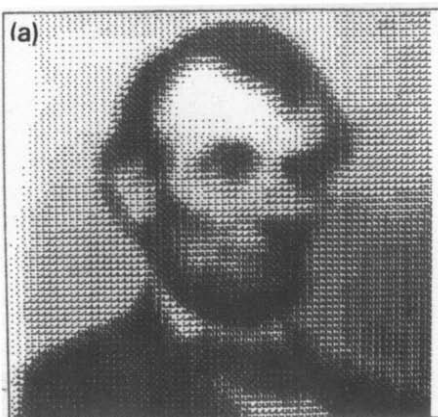


Figure 4. Abraham Lincoln. (a) Input image. (b) Histogram.

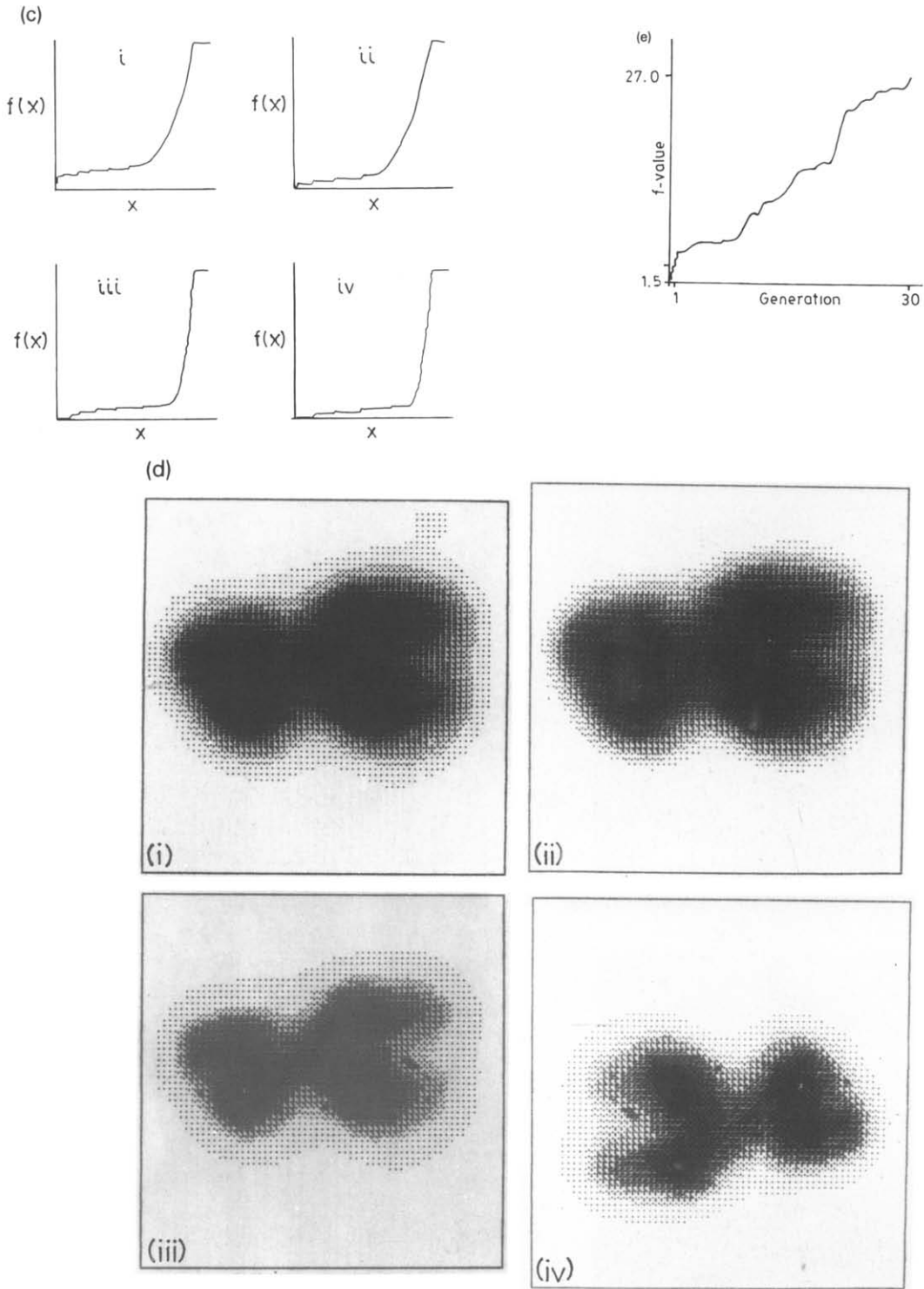


Figure 3. (c) Enhancement functions after (i) 15 generations, (ii) 20 generations, (iii) 25 generations, (iv) 30 generations. (d) Enhanced output after (i) 15 generations, (ii) 20 generations, (iii) 25 generations, (iv) 30 generations. (e) Fitness value as a function of generation number.

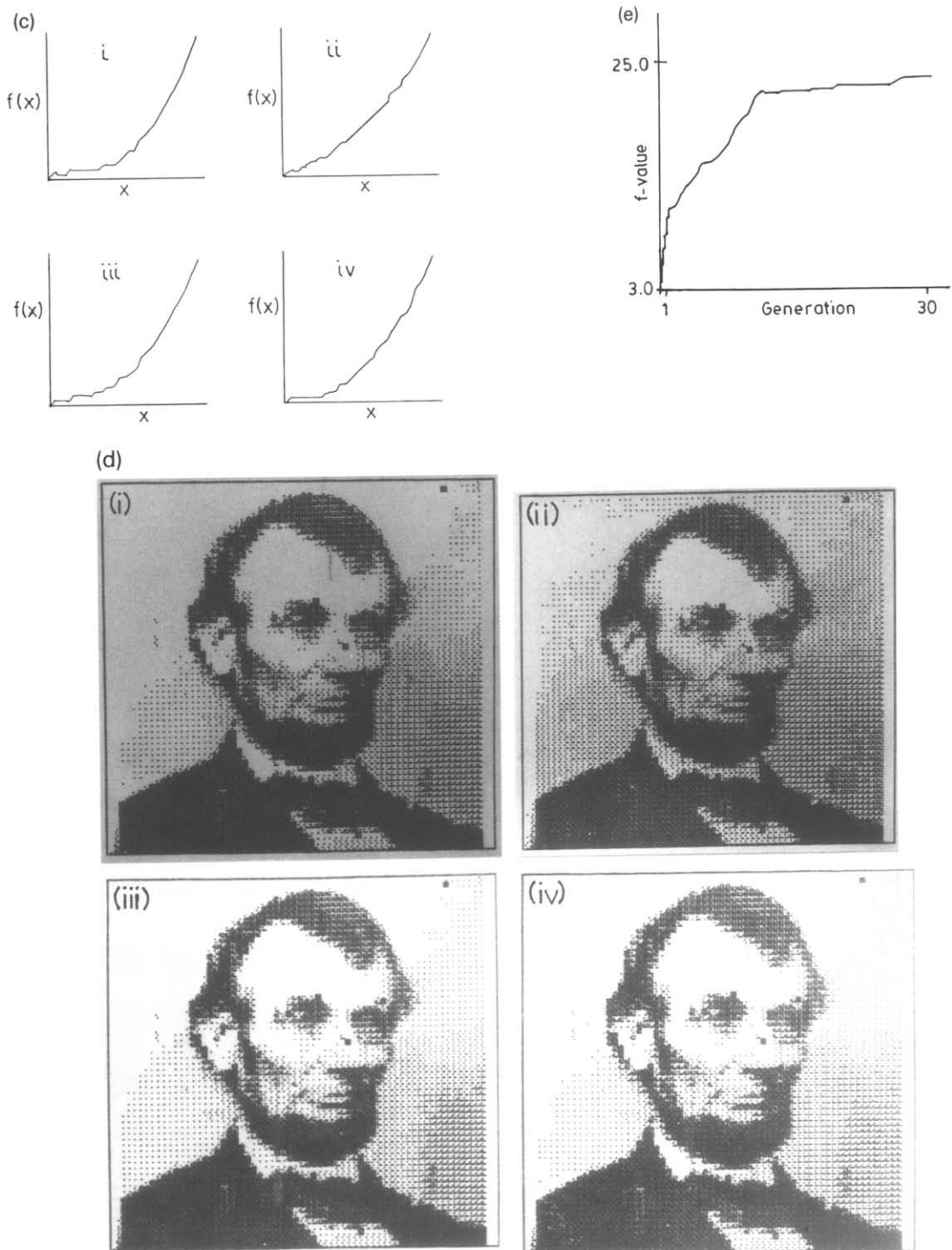


Figure 4. (c) Enhancement functions after (i) 15 generations, (ii) 20 generations, (iii) 25 generations, (iv) 30 generations. (d) Enhanced output after (i) 15 generations, (ii) 20 generations, (iii) 25 generations, (iv) 30 generations. (e) Fitness value as a function of generation number.

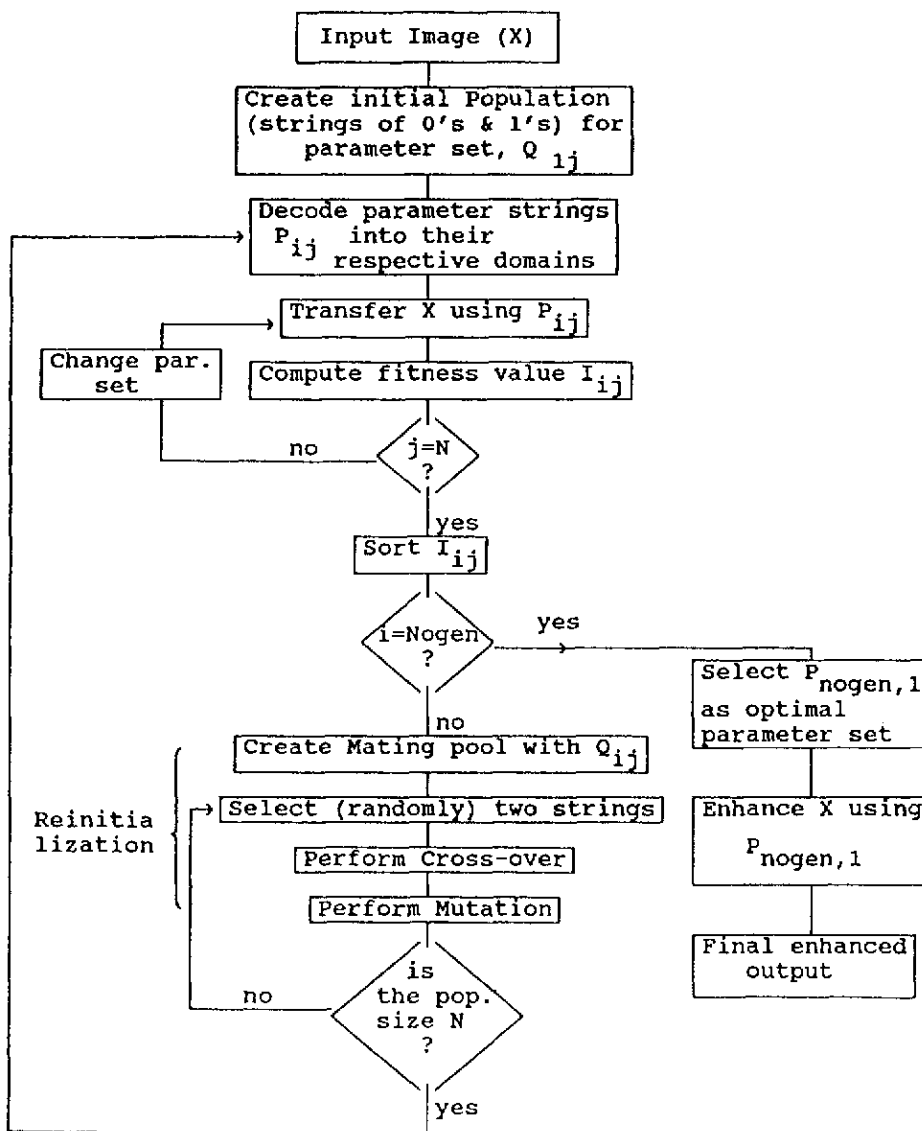


Figure 5. Block diagram of the proposed algorithm (Q_{ij} , P_{ij} and I_{ij} represent the strings of 0 and 1, its decoded version and the fitness values, respectively, corresponding to the j th parameter set of the i th generation, N is the population size and $Nogen$ is the number of generations to be executed).

blurred chromosome (where the shape becomes more and more distinct). Figure 3(e) shows the almost monotonic nondecreasing behavior of the fitness function value in this respect.

Similar results are also found for the Lincoln image (Figures 4(c-e)). Note that we have used IOAC, instead of compactness, in computing the fitness value

for the Lincoln image, because the IOAC measure works better for images having noncompact objects.

6. Conclusions

The effectiveness of GAs in the automatic selec-

tion of an optimum image enhancement operator is demonstrated for both bimodal and multimodal images. The algorithm does not need iterative visual interaction and prior knowledge of image statistics in order to select the appropriate enhancement function. Convergence of the algorithm is experimentally verified. Although fuzziness measures have been used as fitness values, one may use other measures depending on the problems.

The algorithm determines the optimum parameter set rather than the individual parameters in selecting an appropriate enhancement function. Although it considers a large search space (which increases the possibility of getting better results), it requires, in practice, a much smaller number of points to achieve the result (e.g., the algorithm considered 3000 points out of 2^{120}). Note that in this application, the domains of the parameters are continuous. Therefore, to obtain a more accurate solution one needs to increase the length of the strings, though this will increase the computation time.

References

- [1] Ankerbrandt, C.A., B.P. Unckles and F.E. Petry (1990). Scene recognition using genetic algorithms with semantic nets. *Pattern Recognition Lett.* 11, 285-293.
- [2] Davis, L., Ed. (1987). *Genetic Algorithms and Simulated Annealing*. Pitman, London.
- [3] Ekstrom, M.P. (1984). *Digital Image Processing Techniques*. Academic Press, New York, 1984.
- [4] Goldberg, D.E. (1989). *Genetic Algorithms: Search, Optimization and Machine Learning*. Addison-Wesley, Reading, MA.
- [5] Kundu, M.K. and S.K. Pal (1990). Automatic selection of object enhancement operator with quantitative justification based on fuzzy set theoretic measure. *Pattern Recognition Lett.* 11, 811-829.
- [6] Pal, S.K. and A. Ghosh (1992). Fuzzy geometry in image analysis. *Fuzzy Sets and Systems* 48, 23-40.
- [7] Rosenfeld, A. and A.C. Kak (1982). *Digital Picture Processing*. Academic Press, New York.
- [8] Siedlecki W. and I. Sklansky (1989). A note on genetic algorithms for large-scale feature selection. *Pattern Recognition Lett.* 10, 335-347.
- [9] *Proceedings of the Fourth Internat. Conf. on Genetic Algorithms* (1991). University of California, San Diego.