Image Segmentation Using Evolutionary Computation
Suchendra M. Bhandarkar, Member, IEEE, and Hui Zhang

Abstract— Image segmentation denotes a process by which a raw input image is partitioned into nonoverlapping regions such that each region is homogeneous and the union of any two adjacent regions is heterogeneous. A segmented image is considered to be the highest domain-independent abstraction of an input image. The image segmentation problem is treated as one of combinatorial optimization. A cost function which incorporates both edge information and region gray-scale uniformity is defined. The cost function is shown to be multivariate with several local minima. The genetic algorithm, a stochastic optimization technique based on evolutionary computation, is explored in the context of image segmentation. A class of hybrid evolutionary optimization algorithms based on a combination of the genetic algorithm and stochastic annealing algorithms such as simulated annealing, microcanonical annealing, and the random cost algorithm is shown to exhibit superior performance as compared with the canonical genetic algorithm. Experimental results on gray-scale images are presented.

Index Terms— Evolutionary computation, genetic algorithm, image segmentation, microcanonical annealing, random cost algorithm, simulated annealing, stochastic optimization.

I. INTRODUCTION

It is widely acknowledged [37] that computer vision is an information processing activity that involves construction of representations at successive levels of abstraction, starting from the raw pixel level and terminating at a semantically meaningful symbolic level. Image segmentation denotes a process by which a raw input image is partitioned into nonoverlapping regions such that each region is homogeneous and connected, and the union of no two spatially adjacent regions is homogeneous. A segmented image is often considered to be the highest domain-independent or data-driven abstraction of an input image. A segmented image constitutes the primary input to high-level vision which then utilizes domain-specific knowledge to interpret and analyze the image contents.

Each region in a segmented image needs to simultaneously satisfy the properties of homogeneity and connectivity [3]. A region is considered homogeneous if all of its pixels satisfy a homogeneity criterion defined over one or more pixel attributes such as intensity, texture, color, range, etc. A region is considered connected if there exists a connected path between any two pixels within the region. More formally, if \( F \) is the set of all image pixels and \( P(\cdot) \) is a homogeneity predicate defined over groups of connected pixels, then image segmentation is a partitioning of the set \( F \) into a set of connected subsets or regions \( \{S_1, S_2, \ldots, S_n\} \) such that

\[
\bigcup_{i=1}^{n} S_i = F \quad \text{with} \quad S_i \cap S_j = \emptyset, \quad i \neq j. \quad (1)
\]

The homogeneity predicate \( P(\cdot) \) is such that

\[
P(S_i) = \text{true} \quad \text{for all regions } S_i
\]

and

\[
P(S_i \cup S_j) = \text{false} \quad \text{for any two adjacent regions } S_i \text{ and } S_j. \quad (2)
\]

Computing such an image partition is a problem of very high combinatorial complexity. Consequently, the research literature in computer vision and image processing is replete with image segmentation techniques. Most of these have been developed with a certain class of images and a problem domain or class of applications in mind. Development of a unified approach to image segmentation remains an open problem to date, making it an area of active research.

Given the astronomical size of the search space, an exhaustive or near-exhaustive enumeration of all possible image partitions to arrive at a segmented image is practically infeasible. As a result, it is expedient to cast the segmentation procedure as one of combinatorial optimization by defining an objective function in addition to the homogeneity predicate. The desired segmented image is then considered to be one that globally optimizes the objective function in addition to satisfying (1) and (2). The objective function is typically multivariate with a landscape that is fraught with several local optima. This renders deterministic optimization techniques such as local hill-climbing search and gradient descent ineffective since they are prone to stalling in a local optimum. Stochastic optimization techniques, on the other hand, are more suitable since they are capable of forgoing the local optima in the solution space in favor of a global optimum.

This paper explores a more recent class of stochastic optimization techniques based on evolutionary computation. Evolutionary computation encompasses a variety of population-based problem solving techniques that mimic the biological process of Darwinian evolution which is based on the principle of natural selection [20]. Evolutionary algorithms provide
versatile problem-solving mechanisms for search, adaptation, learning, and optimization in a variety of application domains. This paper investigates the genetic algorithm (GA) [24], [28], [39], an important member of the wider class of evolutionary algorithms, in the context of image segmentation. A class of hybrid evolutionary optimization algorithms is proposed that combines the strengths of annealing-based techniques with those of the GA while alleviating their individual weaknesses resulting thereby in performance that is superior to that of either class of techniques used in isolation. We begin with a brief review of stochastic annealing algorithms (knowledge of evolutionary algorithms is assumed).

II. STOCHASTIC ANNEALING

TECHNIQUES—A BRIEF OVERVIEW

Stochastic annealing algorithms such as simulated annealing (SA) [31], microcanonical annealing (MCA) [15], and the random cost algorithm (RCA) [4], [51] are a subclass of stochastic hill-climbing search techniques and are characterized by their capacity to escape from local optima in the objective function. A single iteration of a stochastic annealing algorithm consists of three phases: i) perturb, ii) evaluate, and iii) decide. In the perturb phase, the current solution \( x_i \) to a multivariate objective function \( E(x) \), which is to be minimized, is systematically perturbed to yield another candidate solution \( x_j \). In the evaluate phase, \( E(x_j) \) is computed. In the decide phase, \( x_j \) is accepted and replaces \( x_i \) probabilistically using a stochastic decision function. The stochastic decision function is annealed in a manner such that the search process resembles a random search in the earlier stages and a greedy local search or a deterministic hill-climbing search in the latter stages. The major differences between SA, MCA, and the RCA arise from the differences in the stochastic decision function used in the decision phase. But their common feature is that starting from an initial solution, they generate, in the limit, an ergodic Markov chain of solution states which asymptotically converges to a stationary Boltzmann distribution [1]. The Boltzmann distribution asymptotically converges to a globally optimal solution when subject to the annealing process [22].

A. Simulated Annealing

In the decide phase of SA, the new candidate solution \( x_j \) is accepted with probability \( p \) which is computed using the Metropolis function [38]

\[
p = \begin{cases} 
1, & \text{if } E(x_j) < E(x_i) \\
\exp \left( \frac{E(x_j) - E(x_i)}{T} \right), & \text{if } E(x_j) \geq E(x_i)
\end{cases}
\]

or using the Boltzmann function \( B(T) \) [1]

\[
p = B(T) = \frac{1}{1 + \exp \left( \frac{E(x_j) - E(x_i)}{T} \right)}
\]

at a given value of temperature \( T \), whereas \( x_i \) is retained with probability \( (1-p) \).

The Metropolis function and the Boltzmann function give SA the capability of probabilistically accepting new candidate solutions that are locally suboptimal compared to the current solution thus enabling it to climb out of local minima. Several iterations of SA are carried out for a given value of \( T \) which is then systematically reduced using an annealing function. The iterations carried out for a single value of \( T \) are referred to as an annealing step. As can be seen from (3) and (4), at sufficiently high temperatures, SA resembles a completely random search whereas at lower temperatures it acquires the characteristics of a deterministic hill-climbing search or greedy search.

Both the Metropolis function and the Boltzmann function ensure that SA generates an asymptotically ergodic (and hence stationary) Markov chain of solution states at a given temperature value. Gemam and Gemam [22] have shown that logarithmic annealing schedules of the form \( T_k = R / \log k \) for some value of \( R > 0 \) are asymptotically good, i.e., they ensure asymptotic convergence to a global minimum with unit probability in the limit \( k \rightarrow \infty \).

B. Microcanonical Annealing

MCA models a physical system whose total energy, i.e., sum of kinetic energy and potential energy, is always conserved. The potential energy of the system is the multivariate objective function \( E(x) \) to be minimized whereas the kinetic energy \( E_k \) is represented by a demon or a collection of demons. In the latter case the total kinetic energy is the sum of all the demon energies. The demon energy (or energies) serves to provide the system with an extra degree (or degrees) of freedom thus enabling MCA to escape from local minima.

In the decide phase of MCA, if \( E(x_j) < E(x_i) \) then \( x_j \) is accepted as the new solution. If \( E(x_j) \geq E(x_i) \) then \( x_j \) is accepted as the new solution only if \( E_k \geq E_k(x_j) - E_k(x_i) \). If \( E(x_j) \geq E(x_i) \) and \( E_k \leq E_k(x_j) - E_k(x_i) \) then the current solution \( x_i \) is retained. In the event that \( x_j \) is accepted as the new solution, the kinetic energy demon is updated

\[
E_k^{n+1} = E_k^n + [E_k(x_j) - E_k(x_i)]
\]

to ensure the conservation of the total energy. The kinetic energy parameter \( E_k \) is annealed in a manner similar to the temperature parameter \( T \) in SA. MCA can also be shown to converge to a global minimum with unit probability given a logarithmic annealing schedule [9].

C. Random Cost Algorithm

In the RCA, a random noise variable \( U \) with a uniform distribution in the range \([-\delta, +\delta]\) (where \( \delta > 0 \) is the noise amplitude) takes on the role of the temperature parameter \( T \) in SA and the kinetic energy parameter \( E_k \) in MCA. In the decide phase of the RCA, a new candidate solution \( x_j \) replaces the current solution \( x_i \) if \([E(x_j) - E(x_i)] + U < 0\).

Since the value of \( U \) could be either positive or negative, it means that the RCA could forgo, in certain cases, moves toward the local minimum (when \( U > [E(x_j) - E(x_i)] \)) while accepting moves that steer it away from a local minimum (when \( U < [E(x_j) - E(x_i)] \)). The noise amplitude \( \delta \), which determines the bounds of the distribution of \( U \), is annealed in a manner similar to temperature parameter \( T \) in SA and the kinetic energy parameter \( E_k \) in MCA. The RCA can also be
shown to converge to a global minimum with unit probability given a logarithmic annealing schedule \[4\].

III. THE IMAGE SEGMENTATION PROBLEM

A. Image Segmentation—A Brief Review

Image segmentation has been an area of active research for the past two decades resulting in several image segmentation techniques that have been proposed and described in the computer vision and image processing research literature. This proliferation is in part due to the fact that there exist several problem domains and applications that need to process and interpret image data in a domain-specific or application-specific manner. Moreover, depending on the problem domain or application, there are several types of images that could be processed and analyzed such as, light intensity (gray-scale), color, range (depth), thermal (infrared), sonar, X ray (radiographic), nuclear magnetic resonance images (MRI), and so on. Given the importance of image segmentation in an overall image understanding/machine vision system it is hardly surprising that most image segmentation techniques exhibit a strong domain or application dependence.

The exhaustive nature of the topic makes it impossible to review each individual image segmentation technique in the literature. Fu and Mui \[21\], Haralick and Shapiro \[27\], Pal and Pal \[41\], and Sahoo et al. \[48\] have surveyed and categorized several image segmentation techniques. Most existing image segmentation techniques can be broadly classified as based on:

i) gray-level thresholding which includes local, global, deterministic, fuzzy, and stochastic thresholding schemes \[11\], \[30\], \[36\], \[43\], \[45\];
ii) iterative pixel classification which includes deterministic and stochastic relaxation \[32\], \[42\], \[47\];
iii) Markov random field models \[14\], \[17\]–\[19\], \[25\];
iv) neural networks \[8\], \[23\], \[33\], \[35\], \[53\];
v) surface fitting, surface classification and surface/region growing \[5\], \[26\];
vi) parameter space clustering which includes crisp, probabilistic, and fuzzy clustering \[6\], \[29\];
vii) edge detection \[12\], \[40\], \[44\];
viii) cost function optimization \[7\], \[49\], \[50\].

The segmentation technique considered in this paper is based on cost function optimization. Our previous experience \[7\] and that of Tan et al. \[49\], \[50\] in using the cost function optimization approach to edge detection showed this technique to be both broad in its applicability and capable of producing good results. This provided the motivation for our considering the cost function optimization approach to image segmentation. In this paper we consider conventional light intensity (i.e., gray-scale) images but the approach is general enough to be extended to other types of images.

B. Image Segmentation Via Cost Function Minimization

The cost function optimization approach taken here consists of two phases: i) preprocessing and ii) cost minimization. Given a raw intensity image \[I\], the first step in our segmentation procedure is dissimilarity enhancement where the pixels in the image that are likely candidates for edge pixels are selectively enhanced. Next, the enhanced image is subject to iterative thresholding and connected component labeling to obtain an initial assignment of region labels to pixels. As a result, an initial segmented image is generated which is then subject to a boundary tracing algorithm which finds all the edge pixels to obtain an initial edge configuration. A region adjacency graph (RAG) is generated from the initial segmented image. The RAG is an attributed relational graph wherein the nodes represent the regions in the segmented image and an arc between two nodes represents the edge contour or boundary between the corresponding regions. A list of properties or attributes of the corresponding region is attached with each node in the RAG. With each arc between two nodes is attached a list of edge pixels that comprise the edge contour between the two regions.

The RAG is assigned a cost value which is computed using the cost function defined in Section III-B2 and then subject to a minimization procedure. The RAG is modified/manipulated during the cost minimization procedure. When the minimization procedure is deemed to have converged to a global minimum of the cost function, the segmented image corresponding to the RAG is considered to be the desired segmented image.

1) Preprocessing Phase: The preprocessing phase includes image dissimilarity enhancement, thresholding, connected component labeling, edge tracing, RAG generation, and initial cost evaluation. Edge pixels in an image separate regions that are dissimilar. During dissimilarity enhancement, the properties of such pixels are enhanced i.e., large values are assigned to those pixels in the image that possess this property. The enhanced image \[D = \{D(i, j); 1 \leq i \leq M, 1 \leq j \leq N\}\] is a collection of pixels where each pixel value is proportional to the degree of region dissimilarity that exists at that pixel site. The pixel values in \[D\] lie in the range \([0, 1]\). Pixels with values close to one are good candidates for being considered as edge points. The dissimilarity enhancement algorithm used in this paper is similar to those described in \[7\], \[49\], and \[50\] but is briefly presented here for the sake of completeness.

A set of edge templates of size \(3 \times 3\) is defined based on the following criteria \[49\].

1) An edge pixel has none or one other neighboring edge pixel.
2) An edge pixel has two other neighboring edge pixels and the resulting edge structure does not turn by an angle \(\theta \geq 45^\circ\) as shown in Fig. 1.
3) An edge pixel has three other neighboring edge pixels and forms one of the eight edge structures shown in Fig. 2.
4) An edge pixel has four other neighboring edge pixels and forms one of the two edge structures shown in Fig. 3.

The enhanced image \[D\] is obtained using the following procedure \[50\].

1) All pixels \(D(i, j)\) are initially set to zero.
2) At each pixel site \((i, j)\), steps a) and b) are performed:
   a) Each of the edge structures in the basis set shown in Fig. 4 is fitted to the pixel site \((i, j)\). The pixel site
$(i, j)$ is chosen to be the center pixel for each fitted edge structure from the basis set. The difference in average gray-level values $f(R_1, R_2)$ between regions $R_1$ and $R_2$ as indicated in Fig. 4 is computed for each fitted edge structure. The edge structure that results in the maximum value of $f(R_1, R_2)$ is deemed to be the best fitting edge structure. Note that each edge structure in the basis set contains exactly three edge pixels. For the best fitting edge structure, the sites of the three edge pixels are denoted by $(i, j), (i_1, j_1)$, and $(i_2, j_2)$ (Fig. 4).

b) Nonmaxima suppression is performed by shifting the location of the best fitting edge structure in a direction determined by the edge structure. For vertical, horizontal, and diagonal edge structures, the shift operation is performed by moving the edge location by one pixel in each of the directions perpendicular to the edge. For other edge structures, the shift operation is performed by moving the edge location by one pixel in each of the four directions: up, down, left, and right as shown in Fig. 5. For each shifted edge structure, the new regions $R_1$ and $R_2$ and the corresponding value of $f(R_1, R_2)$ are determined. One of the following two cases is considered:

i) If no larger value of $f(R_1, R_2)$ results from shifting the best fitting edge structure, then $\delta = f(R_1, R_2)/3$ where $f(R_1, R_2)$ is computed from the best fitting edge structure. The value of each of the pixels $D(i, j), D(i_1, j_1)$, and $D(i_2, j_2)$ is incremented by $\delta$.

ii) If the shifting results in a larger value of $f(R_1, R_2)$ then the pixel values $D(i, j), D(i_1, j_1)$, and $D(i_2, j_2)$ are left unaltered.

3) The values of $D(i, j)$ are truncated to a maximum value of 1 at all sites $(i, j)$ to ensure that the final dissimilarity values in image $D(i, j)$ lie in the range $[0, 1]$. The enhanced image $D$ is subject to an iterative thresholding scheme, followed by connected component labeling. The iterative thresholding scheme results in a binary image where pixels are classified as belonging to either foreground or background. The connected component labeling algorithm results in the formation of numerous connected regions in the image. The result of the connected component labeling algorithm is a membership label array $L$, where each array location $L(i, j)$ contains the region label of pixel $(i, j)$. The initial set of edge (boundary) pixels for each of these regions is then obtained by traversing $L$ in raster scan order starting from the first pixel in the foreground. If any of the four-nighbors of the current pixel $p$ is found to have a different region label than $p$, then $p$ is marked as an edge pixel and a nonedge pixel otherwise. The result is a binary image $B$ where the pixels with value 1 represent edge pixels and those with value 0 represent nonedge (i.e., interior) pixels. Note that this procedure causes the interior pixels in the initial regions to be four-connected (which is a more conservative criterion than eight-connectivity) and the initial edge contours to be composed of horizontal and vertical segments (resulting in kink contours with a large number of corner points). The “weak” corners are removed during the cost minimization procedure and the “strong” corners preserved. The pixels along the image boundary are treated as edge pixels and are not visited during the cost minimization process. After the RAG is generated, the cost of the initial segmented image is computed using a cost function described in the following subsection.

2) Cost Function Evaluation: The cost function for image segmentation is defined as

$$E = w_1 \left( \sum_{i=1}^{R} \sigma_i^2 \right) + w_2 C_e$$  

(5)

where $\sigma_i^2$ is the gray-level variance of region $i$, $R$ is the total number of regions, and $C_e$ is the cost associated with the
Fig. 4. Regions of interest for valid two-neighbor edge structures. The lighter and darker shaded areas are denoted by $R_1$ and $R_2$ in the text.

Fig. 5. Examples of nonmaximum suppression.
resulting edge configuration. The edge cost function $C_e$ is similar to the one used by Bhandarkar et al. [7] and Tan et al. [49], [50] in their work on edge detection.

The edge cost function $C_e(i, j)$ at each pixel site $(i, j)$ is a weighted sum of the following terms.

1) $c_1$ which is a measure of local region dissimilarity. This term tends to place edge pixels at sites of high region dissimilarity and penalizes pixel sites with little gray-level variation. If the pixel $(i, j)$ is an edge pixel, then $c_1(i, j) = 0$ else $c_1(i, j) = D(i, j)$.

2) $c_2$ which penalizes pixel sites that result in thick edges while favoring pixel sites that result in thin edges i.e., edges that are one pixel thick. A thick edge pixel is a pixel the presence of which causes multiple links between neighboring (edge) pixels in the edge structure as shown in Fig. 6. The cost factor $c_2$ is computed by examining the pixel $p$ at position $(i, j)$ in the edge image $B$. If pixel $p$ is a thick edge pixel then $c_2(i, j) = 1$, otherwise, $c_2(i, j) = 0$.

3) $c_3$ which penalizes nonendpoint edge pixels based on a local measure of curvature. This term tends to smooth out or remove highly kinky edges. The cost factor $c_3(i, j)$ is computed for each nonendpoint edge pixel at site $(i, j)$ in the edge image $B$ (an endpoint is an edge pixel that has at most one neighboring edge pixel). If there is a pair of neighboring edge pixels in $B$ connected via the edge pixel at site $(i, j)$ such that the resulting edge structure turns by more than $45^\circ$, then $c_3(i, j) = 1$. If there is no pair of neighboring edge pixels that causes the resulting edge structure to turn by more than $45^\circ$, but there exists a pair of edge pixels that causes the resulting edge structure to turn by $45^\circ$, then $c_3(i, j) = 0.5$. If there is no pair of neighboring edge pixels that causes the resulting edge structure to turn by more than $0^\circ$ then $c_3(i, j) = 0$.

4) $c_4$ which penalizes nonendpoint edge pixels that correspond to fragmented edges. This term links up or removes fragmented edges. If the edge pixel at site $(i, j)$ in the edge image $B$ does not have any neighboring edge pixel, then $c_4(i, j) = 1$. If the edge pixel at site $(i, j)$ has only one neighboring edge pixel, then $c_4(i, j) = 0.5$. For all other cases, $c_4(i, j) = 0$.

5) $c_5$ which assigns a unit cost to each edge pixel, thereby preventing an excessive number of edge pixels from being detected. Note that the cost factor $c_1$ favors the detection of edge pixels where $f(R_1, R_2)$ is nonzero, thus leading to the detection of an excessive number of edge pixels. The cost factor $c_5$ is formulated to suppress this tendency. If the pixel at site $(i, j)$ is an edge pixel, then $c_5(i, j) = 1$ otherwise $c_5(i, j) = 0$.

Each of the cost factors can be seen to place different and often conflicting requirements on the final edge image, which therefore can be considered to be the one that best satisfies all the constraints imposed by the cost factors, i.e., it can be looked upon as a problem in constrained optimization. The edge cost function $C_e(i, j)$ at an individual pixel site $(i, j)$ is given by

$$C_e(i, j) = \sum_{k=1}^{5} w_k c_k$$

where the $w_k$'s are empirically predetermined weights assigned to the respective terms.

The edge cost function for an entire image of size $M \times N$ pixels is given by

$$C_e = \sum_{i=1}^{M} \sum_{j=1}^{N} C_e(i, j).$$

The cost function associated with region assignment (i.e., the function that assigns a region label to each pixel in the image) is defined as the sum of the gray-level variances of all the resulting regions. The total cost associated with the image segmentation is the weighted sum of the edge configuration cost and the region assignment cost.

The initial segmented image is then subject to a cost minimization procedure. An efficient state transition strategy or solution perturbation strategy is critical to implement the mutation operator in a GA-based cost minimization procedure.

The following state transition strategy was implemented.

1) Let $p$ be the current pixel, starting with the first pixel.
2) Let $q_k$, $1 \leq k \leq 8$, be the eight-neighbors of pixel $p$.
3) Generate a random number $n$ between 1 and 8.
4) If region_label($p$) = region_label($q_n$), go to 6 else let region_label($p$) = region_label($q_n$) i.e., assign $p$ to the neighboring region, update the RAG, go to 5.
5) Since region_label($p$) is changed, update its edge configuration cost and that of its eight-neighboring pixels $q_k$, $1 \leq k \leq 8$.
6) Let $p = $ next pixel in raster scan order, go to 2.
7) Stop.

After generating a new candidate state, it would be necessary to recompute the cost for the new candidate state. Since it is inefficient to recalculate the cost function from scratch for every new candidate state, the cost functions associated with the edge configuration and region assignment are incrementally computed for each state transition. Since the edge configuration cost function is computed locally at each pixel site, it is amenable to incremental computation [7], [49], [50]. To render the region assignment function suitable for incremental computation, the following information is maintained in each region node of the RAG: the number of pixels in each region $n$, the sum of the gray-level values of the pixels in the region $\sum_k g_k$, the sum of the squares of the
gray-level values of the pixels in the region \( \sum_k g_k^2 \), and the square of the sum of the gray-level values of the pixels in the region \( (\sum_k g_k)^2 \). After each transition, the RAG is updated and the mean \( \mu \) and the variance \( \sigma^2 \) of each region involved in the state transition are incremented as follows:

\[
\Delta \mu = \frac{1}{n + \Delta n} \Delta \left( \sum_k g_k \right) - \frac{\Delta n}{n(n + \Delta n)} \left( \sum_k g_k \right)
\]

\[
\Delta \sigma^2 = \frac{1}{n + \Delta n} \Delta \left( \sum_k g_k^2 \right) - \frac{1}{(n + \Delta n)^2} \left[ \Delta \left( \sum_k g_k \right) \right]^2
\]

Equation (9) is used in computing the change \( \Delta E \) in the cost function \( E \) defined in (5). Equation (8) is used to update the average gray level of each region which in turn is used in the rendering of the segmented image wherein each region is represented by its average gray level.

C. Cost Function Minimization Using the GA

Two basic issues need to be addressed in designing a GA for image segmentation via cost function minimization: a) an appropriate representation for each potential solution and b) the formulation of an appropriate fitness function in terms of the cost function. In the traditional GA, an individual is usually represented by a binary string. The fact that we are dealing with images which are two-dimensional matrices and the need to include both region and edge information in the representation of an individual prompted us to use an alternative representation. Each individual in the population represents a segmented image and contains the following:

a) the membership label array \( L \);

b) the edge image \( B \);

c) the RAG associated with \( L \) and \( B \).

The fitness of the \( k \)th individual in the current generation is computed as

\[
F_k = (E_w - E_k)^2
\]

where \( E_w \) is the cost associated with the worst individual and \( E_k \) the cost associated with the \( k \)th individual in the current generation. Both \( E_w \) and \( E_k \) are computed using (5).

The GA-based segmentation algorithm can be implemented as a simple iterative procedure.

1) An initial population is created from the initial segmented image and the fitness for each individual is evaluated using (10).

2) Two mates are selected for reproduction with probabilities that are proportional to their fitness values using roulette wheel selection.

3) The two-point crossover operator is applied to the two mates, and two offspring are generated.

4) The mutation operator which incorporates the state transition strategy described in Section III-B2 is applied to the newly generated offspring.

5) The fitness values for the offspring are computed.

6) Steps 2)–5) are repeated until an entirely new population of individuals is generated.

7) The previous population is replaced with the new population.

8) If the stopping criterion is satisfied, go to step 9) otherwise go to step 2).

9) Output the individual with the best fitness value and stop.

The solution found by the GA is used to generate a segmented image. Each of the above steps is elaborated in the following.

Initial Population Generation: The initial population is generated from the initial preprocessed and segmented image via a series of random mutations.

Selection Operator: A pair of individuals is selected from the current population for mating using a biased roulette wheel selection procedure. Each individual in the population has a roulette wheel slot sized in proportion to its fitness value. This results in fitter individuals contributing, on average, a greater number of offspring in the succeeding generation(s) and the gradual elimination of the less fit individuals.

Crossover Operator: Crossover is applied to the newly selected (parent) individuals to generate two offspring. Since our representation is two dimensional, two-point crossover is employed [7] on the membership label arrays \( L \) of the parents to generate the membership label arrays of the offspring (Fig. 7). The edge image \( B \) of each offspring is derived from its membership label array \( L \) as explained in Section III-B1. The crossover operation is the primary mechanism used by the GA in its exploitation of the search space.

Mutation Operator: The mutation operator incorporates the state transition strategy described in Section III-B2. In our implementation only edge pixels are candidates for mutation (i.e., pixels in the interior of a region are not considered for mutation). This is so because only a mutation performed on a region boundary pixel using the aforementioned state transition strategy could possibly result in a change in the region label assignment for that pixel. The mutation operator, by introducing random changes in the individuals, introduces diversity into the gene pool thus preventing the inadvertent loss of useful genetic material in the earlier phases of evolution. In terms of optimization, the mutation operator prevents a GA from being trapped in an undesired local optimum.

Advanced and Meta-Level Genetic Algorithm Operators: The operations of population initialization, mate selection, crossover, mutation, and population replacement constitute a canonical GA. In the context of the image segmentation problem, the convergence rate of the canonical GA was found to be very slow [52]. An elitism strategy, the engineered conditioning operator, and adaptation of basic canonical GA operators were found to accelerate convergence [7]. In a canonical GA, the entire previous generation is replaced by the new generation. This scheme suffers from a drawback namely that there exists a nonzero probability of good individuals in the current generation not being selected for the succeeding generation. The elitism strategy ensures that the best individual(s) in the current generation always survive into the following generation thereby preventing a potential inadvertent loss of this(these) individual(s). In our implementation, the best individual in the current generation always survived into the next generation.
Parents

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>2</th>
<th>2</th>
<th>3</th>
<th>3</th>
<th>3</th>
<th>3</th>
<th>3</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>

Offspring

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>2</th>
<th>2</th>
<th>2</th>
<th>2</th>
<th>2</th>
<th>2</th>
<th>3</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>

Fig. 7. The special two-dimensional crossover operation used. Note the exchange of region labels between the two parents (left) inside the box.

---

Initialize the temperature parameter to a high value $T = T_0$;
Generate an initial population $G = G_0$;
repeat
  |
  From present generation $G$ form a mating pool of $2n$ individuals
  using roulette-wheel selection;
  |
  for $i := 0$ to $n$ do
  |
  From parents $P[2i]$ and $P[2i+1]$ in the mating pool generate two
  offspring $C[2i]$ and $C[2i+1]$ using the GA crossover operator;
  |
  Bring about a random localized change in each offspring using
  the GA mutation operator;
  |
  Replace parent $P[2i]$ in $G$ with offspring $C[2i]$ with probability
  $1 - B(T); /* B(T) is the Boltzmann decision function */$
  |
  Replace parent $P[2i+1]$ in $G$ with offspring $C[2i+1]$ with probability
  $1 - B(T);$ |
  |
  Reduce the temperature using the annealing function $T := A(T);$ |
  until the convergence criterion is met;

---

Fig. 8. The SA-GA hybrid algorithm.

The engineered conditioning (EC) operator [7] employed in our version of the GA works in combination with the canonical operators and is used for local improvement in the search space. The EC operator represents a greedy or local search of the solution space where a small portion of the best individuals in the current generation is mutated. The original individual is compared with the modified or conditioned individual. The original individual is replaced with the conditioned individual only if the conditioned individual is found to be better. The EC operator is characterized by two parameters: $c_1$ which denotes the fraction of individuals in the current population to be subject to the EC operator and $c_2$ which denotes the number of pixels in the chosen individual to be subject to the EC operator. In our implementation, in the beginning of the GA run, $c_1 = 0.1$ and $c_2 = 0.2$. Both $c_1$ and $c_2$ are incremented by a small amount after each generation. Eventually, $c_1 = 1$ which means all individuals are conditioned; so the very late stages of the GA are characterized by a large number of local hill-climbing moves. The initial values of $c_1$ and $c_2$ and the corresponding increment values were chosen empirically after several experiments. The rationale is that in the later stages of evolution it is highly likely that the GA is in the vicinity of a global optimum. In this situation it is imperative that the GA focus on local search.
Start with an initial population of $2n$ individuals $G = G_{\text{init}}$;
Initialize the kinetic energy demon value to a high value $E_k = E_{k_{\text{max}}}$;
Repeat
{
Create a mating pool of $2n$ individuals using roulette-wheel selection;
Compute the kinetic energy of the mating pool $E_{k_{\text{mat}}} = E_{\text{pop}} + E_k - E_{\text{mat}}$;
/* $E_{\text{pop}}$ = total potential energy of the population
$E_{\text{mat}}$ = total potential energy of the mating pool
$E_{k_{\text{mat}}}$ = kinetic energy value of the mating pool */
for($i = 0; i < n; i++$)
{
From parents $P[2i]$ and $P[2i+1]$ in the mating pool generate two
offspring $C[2i]$ and $C[2i+1]$ using the GA crossover operator;
 Bring about a random localized change in each offspring using the GA
mutation operator;
(1) if ($E(C[2i]) < E(P[2i])$) /* $E(C[2i])$ and $E(P[2i])$ are the potential
energies of $P[2i]$ and $C[2i]$ respectively */
{
(1.a) add $C[2i]$ to the next generation $G_{\text{next}}$;
(1.b) $E_{k_{\text{mat}}} = E_{k_{\text{mat}}} + E(P[2i]) - E(C[2i])$
}
(2) else if (($E(C[2i]) - E(P[2i])$) < $E_{k_{\text{mat}}}$)
{
(2.a) add $C[2i]$ to the next generation $G_{\text{next}}$;
(2.b) $E_{k_{\text{mat}}} = E_{k_{\text{mat}}} + E(P[2i]) - E(C[2i])$
}
Repeat steps (1) and (2) with $P[2i+1]$ and $C[2i+1]$;
$G = G_{\text{next}}$;
$E_k = E_{k_{\text{mat}}}$;
Reduce demon kinetic energy using annealing function $E_k = A(E_k)$;
} until the convergence criterion is met;

Fig. 9. The MCA-GA hybrid algorithm.

In a canonical GA the crossover and mutation operators
are assigned prespecified relatively high and low probabilities
respectively. In our GA implementation, a high probability
is assigned to the crossover operator in the initial stages of
the GA run and the crossover probability is decreased by
a small amount with every generation. In the later stages
of the GA run, only the mutation operator contributes to
the improvement of the solutions, and since all individuals
are subject to the EC operator, the GA degenerates into a
greedy or local search. The initial values of the crossover
and mutation probabilities and the corresponding decrement
and increment values respectively were chosen empirically
after several experiments. As in the case of the EC operator,
the rationale here is to enable the GA in the later stages of
evolution to focus on local search via mutation while forgoing
exploration of large regions of the search space via crossover.

Convergence Criterion: Convergence was declared when
the fitness value of the best individual in the population had
not changed over the past $k = 5$ consecutive generations.

D. Advantages and Limitations of GA-Based Segmentation

Some of the advantages of the GA-based image segmentation
 technique are the following.

a) The selection and crossover operators enable useful
subsolutions, referred to as building blocks or schema
in the GA literature, to be propagated and combined
to construct better and more global solutions with every
succeeding generation. We observed that compact
homogeneous regions in a partially segmented image
served as building blocks and were propagated from one
generation to the next as the segmentation was refined.
b) The GA can be easily parallelized since the selection,
crossover, and mutation operators can be performed in
parallel over all the individuals in a population.

The GA-based image segmentation techniques, however, are
seen to suffer from the following limitations.
a) The performance of the GA is extremely sensitive to
the manner in which the individual is encoded. For the
building blocks property of the GA to hold, the encoding
technique should ensure the existence of small-length,
high-fitness value schema that can be propagated and
combined over the course of succeeding generations
with a low probability of disruption by crossover. In the
context of image segmentation, it is important that spatial
locality in the image be preserved in the representation
of an individual (i.e., features that are spatially proximate
in the image have their encoded representations
positioned in mutual proximity in the representation of
the individual and vice versa).
Initialize the noise amplitude to a high value \( \Delta = \Delta_{\text{init}} \);
Start with an initial population of individuals \( G = G_{\text{init}} \);
repeat
\{ 
From present generation \( G \) form a mating pool of \( 2n \) individuals using roulette-wheel selection;
for (i=0; i < n; i++) 
\{ 
From parents \( P[2i] \) and \( P[2i+1] \) in the mating pool generate two
offspring \( C[2i] \) and \( C[2i+1] \) using the GA crossover operator;
Bring about a random localized change in each offspring using the GA
mutation operator;
Generate a random variable \( U \) in the range \([-\Delta, +\Delta]\);
Replace parent \( P[2i] \) in \( G \) with offspring \( C[2i] \)
    \( \text{if } (E(C[2i]) - E(P[2i])) + U < 0; \)
Replace parent \( P[2i+1] \) in \( G \) with offspring \( C[2i+1] \)
    \( \text{if } (E(C[2i+1]) - E(P[2i+1])) + U < 0; \)
\} 
Reduce noise amplitude using the annealing function \( \Delta = A(\Delta) \);
\} until the convergence criterion is met;

Fig. 10. The RCA-GA hybrid algorithm.

---

The Boeing image.

b) In the absence of a hill-climbing mechanism, the number
   of generations (and hence the execution time) needed for
   convergence is fairly large.

c) With the incorporation of a deterministic hill-climbing
   mechanism (such as the EC operator) the GA exhibits
   premature convergence to a suboptimal solution [7].

The CRT image.
Segmentation techniques based on stochastic annealing (such as SA, MCA, and the RCA) on the other hand, have the following advantages.

a) The stochastic hill climbing mechanism guarantees asymptotic convergence to a global optimum [4], [15], [22].

b) The performance is more resilient to the manner in which the candidate solutions to the problem are encoded. The primary requirement of the encoding scheme is that the changes in the cost function be easy to compute. In practice, this is a much weaker requirement than in the case of the GA.

c) The perturbation operators can be relatively easily designed so that efficient computation of the change in the cost function is possible.

The stochastic annealing-based image segmentation techniques, however, are seen to suffer from the following limitations.

a) The search procedure is fairly localized, preventing them from exploring the same diversity of solutions that GA’s can. This causes the annealing schedule needed for asymptotic convergence to a global optimum to be computationally intensive.

b) Since SA, MCA, and the RCA are based on generating an asymptotically ergodic Markov chain from an initial starting state, they are inherently serial. Attempts at parallelization have met with limited success primarily because the parallel algorithms, in their attempt to maximize speedup and efficiency of processor utilization, invariably compromise the basic convergence properties of the corresponding serial algorithms.

c) The methods do not exploit previously encountered good subsolutions in their future explorations of the search space, i.e., the next state is dependent only on the present state. As a consequence, the annealing techniques do not incorporate the building-blocks property of GA’s.
TABLE I
RESULTS OBTAINED FROM THE PREPROCESSING PHASE

<table>
<thead>
<tr>
<th>Image</th>
<th>No. of Initial Regions</th>
<th>Initial Edge Cost</th>
<th>Initial Region Cost</th>
<th>Total Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boeing</td>
<td>200</td>
<td>170663.00</td>
<td>15810.40</td>
<td>185573.40</td>
</tr>
<tr>
<td>Boeing, ( \alpha^2 = 25 )</td>
<td>209</td>
<td>175864.00</td>
<td>46271.60</td>
<td>222105.60</td>
</tr>
<tr>
<td>Boeing, ( \alpha^2 = 100 )</td>
<td>272</td>
<td>294561.00</td>
<td>390031.00</td>
<td>684592.00</td>
</tr>
<tr>
<td>CRT</td>
<td>70</td>
<td>97901.40</td>
<td>9619.91</td>
<td>107521.31</td>
</tr>
<tr>
<td>CRT, ( \alpha^2 = 25 )</td>
<td>158</td>
<td>106514.00</td>
<td>352081.00</td>
<td>459198.00</td>
</tr>
<tr>
<td>CRT, ( \alpha^2 = 100 )</td>
<td>193</td>
<td>130599.00</td>
<td>561472.00</td>
<td>692071.00</td>
</tr>
<tr>
<td>Computer</td>
<td>478</td>
<td>214531.00</td>
<td>97805.60</td>
<td>312431.60</td>
</tr>
<tr>
<td>Phone</td>
<td>277</td>
<td>253302.00</td>
<td>16048.10</td>
<td>269350.10</td>
</tr>
</tbody>
</table>

TABLE II

<table>
<thead>
<tr>
<th>Image</th>
<th>Algorithm</th>
<th>Final Cost</th>
<th>CPU Time (min.)</th>
<th>No. of Generations or Annealing Steps</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boeing</td>
<td>GA</td>
<td>73,928.54 ± 359.63</td>
<td>704.27 ± 11.81</td>
<td>81.17 ± 0.92</td>
</tr>
<tr>
<td></td>
<td>MCA-GA</td>
<td>63,438.77 ± 239.75</td>
<td>434.15 ± 6.34</td>
<td>62.94 ± 0.54</td>
</tr>
<tr>
<td></td>
<td>RCA-GA</td>
<td>66,851.34 ± 254.07</td>
<td>385.87 ± 7.87</td>
<td>56.31 ± 0.64</td>
</tr>
<tr>
<td></td>
<td>SA-GA</td>
<td>63,693.42 ± 214.71</td>
<td>507.42 ± 5.37</td>
<td>62.75 ± 0.43</td>
</tr>
<tr>
<td>CRT</td>
<td>GA</td>
<td>38,963.31 ± 212.37</td>
<td>698.42 ± 10.73</td>
<td>71.43 ± 0.81</td>
</tr>
<tr>
<td></td>
<td>MCA-GA</td>
<td>36,767.41 ± 141.27</td>
<td>214.77 ± 3.21</td>
<td>31.23 ± 0.31</td>
</tr>
<tr>
<td></td>
<td>RCA-GA</td>
<td>37,853.73 ± 143.91</td>
<td>170.54 ± 3.63</td>
<td>23.47 ± 0.31</td>
</tr>
<tr>
<td></td>
<td>SA-GA</td>
<td>35,299.63 ± 123.73</td>
<td>275.52 ± 3.08</td>
<td>32.57 ± 0.27</td>
</tr>
<tr>
<td>Computer</td>
<td>GA</td>
<td>101,037.63 ± 545.74</td>
<td>856.83 ± 10.37</td>
<td>83.64 ± 1.27</td>
</tr>
<tr>
<td></td>
<td>MCA-GA</td>
<td>74,246.57 ± 229.33</td>
<td>435.82 ± 6.03</td>
<td>55.38 ± 0.52</td>
</tr>
<tr>
<td></td>
<td>RCA-GA</td>
<td>85,392.78 ± 231.11</td>
<td>399.72 ± 6.02</td>
<td>52.67 ± 0.61</td>
</tr>
<tr>
<td></td>
<td>SA-GA</td>
<td>75,908.57 ± 252.63</td>
<td>509.38 ± 5.54</td>
<td>56.82 ± 0.41</td>
</tr>
<tr>
<td>Phone</td>
<td>GA</td>
<td>76,905.54 ± 412.37</td>
<td>796.91 ± 12.32</td>
<td>75.28 ± 1.03</td>
</tr>
<tr>
<td></td>
<td>MCA-GA</td>
<td>66,103.75 ± 254.33</td>
<td>422.92 ± 6.37</td>
<td>51.73 ± 0.47</td>
</tr>
<tr>
<td></td>
<td>RCA-GA</td>
<td>72,307.43 ± 241.89</td>
<td>409.72 ± 7.97</td>
<td>50.82 ± 0.57</td>
</tr>
<tr>
<td></td>
<td>SA-GA</td>
<td>68,827.33 ± 238.48</td>
<td>486.33 ± 4.46</td>
<td>54.22 ± 0.39</td>
</tr>
<tr>
<td></td>
<td>GA</td>
<td>73,506.43 ± 455.23</td>
<td>795.86 ± 12.37</td>
<td>80.51 ± 1.17</td>
</tr>
<tr>
<td>Boeing</td>
<td>( \alpha^2 = 25 )</td>
<td>62,417.42 ± 295.77</td>
<td>483.87 ± 7.57</td>
<td>68.43 ± 0.77</td>
</tr>
<tr>
<td></td>
<td>SA-GA</td>
<td>64,123.33 ± 257.66</td>
<td>523.57 ± 8.67</td>
<td>65.68 ± 0.61</td>
</tr>
<tr>
<td></td>
<td>GA</td>
<td>84,522.43 ± 715.72</td>
<td>809.54 ± 16.72</td>
<td>97.73 ± 1.56</td>
</tr>
<tr>
<td>Boeing</td>
<td>( \alpha^2 = 100 )</td>
<td>65,001.77 ± 433.72</td>
<td>475.77 ± 10.17</td>
<td>68.14 ± 0.86</td>
</tr>
<tr>
<td></td>
<td>MCA-GA</td>
<td>71,587.91 ± 501.23</td>
<td>465.62 ± 11.95</td>
<td>63.37 ± 1.07</td>
</tr>
<tr>
<td></td>
<td>SA-GA</td>
<td>67,298.46 ± 357.85</td>
<td>577.64 ± 8.23</td>
<td>72.27 ± 0.68</td>
</tr>
<tr>
<td></td>
<td>GA</td>
<td>50,824.76 ± 321.43</td>
<td>707.54 ± 11.08</td>
<td>80.44 ± 1.00</td>
</tr>
<tr>
<td>CRT</td>
<td>GA</td>
<td>41,927.45 ± 193.06</td>
<td>313.48 ± 4.85</td>
<td>44.67 ± 0.53</td>
</tr>
<tr>
<td></td>
<td>MCA-GA</td>
<td>43,844.49 ± 209.83</td>
<td>275.91 ± 5.79</td>
<td>40.12 ± 0.59</td>
</tr>
<tr>
<td></td>
<td>SA-GA</td>
<td>41,808.37 ± 166.72</td>
<td>291.38 ± 4.81</td>
<td>48.73 ± 0.51</td>
</tr>
<tr>
<td></td>
<td>GA</td>
<td>54,697.73 ± 457.52</td>
<td>713.87 ± 13.68</td>
<td>81.17 ± 1.29</td>
</tr>
<tr>
<td>CRT</td>
<td>( \alpha^2 = 25 )</td>
<td>49,457.83 ± 326.42</td>
<td>304.24 ± 6.47</td>
<td>42.67 ± 0.59</td>
</tr>
<tr>
<td></td>
<td>MCA-GA</td>
<td>50,272.78 ± 361.82</td>
<td>252.47 ± 6.80</td>
<td>35.84 ± 0.68</td>
</tr>
<tr>
<td></td>
<td>SA-GA</td>
<td>40,054.55 ± 288.27</td>
<td>405.71 ± 5.77</td>
<td>50.27 ± 0.52</td>
</tr>
</tbody>
</table>

1) **GA with SA-Like, MCA-Like, or RCA-Like Operators:** The overall structure of the hybrid algorithm is like the GA. The probabilities assigned to the individual GA operators such as crossover and mutation, however, are annealed in a manner similar to SA, MCA, or the RCA [2]. Such a hybrid algorithm, like the GA, however, does not guarantee asymptotic convergence.

2) **SA, MCA, or the RCA with GA-Like Operators:** SA, MCA, or the RCA maintains a population of candidate solutions. The GA operators (i.e., crossover and mutation) are treated as solution perturbation strategies in an overall population-based stochastic annealing algorithm. The population replacement strategy incorporates a stochastic tournament between the parents and the offspring where the stochastic selection function is annealed in a manner similar to SA, MCA, or the RCA [34]. Such an algorithm can be shown to retain the asymptotic convergence properties of SA, MCA, or the RCA [34] and yet benefit from faster convergence resulting from the diversity of solutions in the population. This is the approach that is taken in this paper.

A. The SA-GA Hybrid Algorithm

The algorithm that combines the GA and SA, as originally proposed by Mahfood and Goldberg [34], is depicted in Fig. 8. The annealing function \( A(T) \) is of the form \( A(T) = \alpha T \) where \( \alpha = 0.95 \). Although the aforementioned geometric annealing schedule does not strictly ensure asymptotic convergence to a global optimum as the logarithmic annealing schedule does, it is much faster and has been found to yield very good solutions in practice [13], [46]. Population replacement is done via a Boltzmann tournament between a parent and its offspring. A parent is chosen at random and paired with one of its offspring which is also chosen randomly. The other parent is paired with the other offspring. The Boltzmann tournament replaces the parent \( P \) with the child \( C \) at a given temperature value \( T \) with probability \( p = B(T) \) where \( B(T) \) is the Boltzmann function given by

\[
B(T) = \frac{1}{1 + \exp \left( \frac{E(C) - E(P)}{T} \right)}. \tag{11}
\]

As can be seen from (11), at sufficiently high temperature values, the replacement is purely random, i.e., \( p = 0.5 \),...
whereas at lower temperatures, the fitter individual is preferred resulting in a greedy or local search. The convergence criterion used is the same as that for the GA.

B. The MCA–GA Hybrid Algorithm

The MCA–GA algorithm is depicted in Fig. 9. The entire population is treated as a microcanonical ensemble. From a given population of individuals, a mating pool is created using roulette wheel selection. The value of the kinetic energy demon for the mating pool is computed in accordance with the law of conservation of total energy. As in the case of the SA–GA algorithm, a parent is chosen at random and paired with one of its offspring which is also chosen randomly. The other parent is paired with the other offspring. The parent is replaced with the child in accordance with the MCA decision function discussed in Section II-B. The kinetic energy demon value is annealed using the same geometric annealing schedule as in the SA–GA hybrid algorithm. The convergence criterion used is the same as that for the GA.

C. The RCA–GA Hybrid Algorithm

The RCA–GA algorithm is depicted in Fig. 10. A noise variable is associated with the entire population. From a given population of individuals, a mating pool is created using roulette wheel selection. As in the case of the SA–GA and MCA–GA algorithms, a parent is chosen at random and paired
Fig. 19. Convergence curves for noisy Boeing image, $\sigma_n^2 = 100$.

Fig. 20. Convergence curves for noisy CRT image, $\sigma_n^2 = 100$.

with one of its offspring which is also chosen randomly. The other parent is paired with the other offspring. The parent is replaced with the child in accordance with the RCA decision function discussed in Section II-C. The noise amplitude $\delta$ is annealed using the same geometric annealing schedule as in the SA–GA hybrid algorithm. The convergence criterion used is the same as that for the GA.

V. EXPERIMENTAL RESULTS

The segmentation algorithms based on the GA and the three hybrid stochastic optimization algorithms SA–GA, MCA–GA, and RCA–GA were implemented and tested on several gray-scale images. In this section we present and compare results obtained from each of these algorithms on four different images shown in Figs. 11–14 and referred to as Boeing, CRT, Computer, and Phone, respectively. To test the segmentation algorithms in the presence of noise, each of these images was corrupted with zero mean Gaussian noise with varying variance ($\sigma_n^2$) values. For example, Fig. 15 shows the Boeing image corrupted with zero mean Gaussian noise having variance $\sigma_n^2 = 100$. All the original and noisy images were subject to the preprocessing phase which included dissimilarity enhancement, thresholding, connected component labeling, RAG generation, and initial cost evaluation. Dissimilarity
enhancement yielded an enhanced image in which each pixel value was between zero and one where pixels with the values close to one were considered good candidates for edge pixels. Iterative thresholding on the enhanced image resulted in a binary image which was then subject to connected component labeling, RAG generation, and initial cost evaluation. Fig. 16 shows the result of preprocessing on the Boeing image. Table I summarizes the results obtained from the preprocessing phase for all the four images.

Table II compares the results of each of the evolutionary algorithms (the GA, MCA-GA, RCA-GA, and SA-GA) on the noise-free and noisy gray-scale images with varying values of noise variance. The performance figures in Table II for each of the algorithms (i.e., execution times, final cost, and number of annealing steps) represent average values over 40 runs of that algorithm. The values arising from 40 runs of each algorithm could be looked upon as a random sample of size 40 drawn from a population with unknown distribution.

The central limit theorem [10] states that for a sample $X_1, X_2, \ldots, X_N$ of size $N$, where $N$ is large, the distribution of the sample mean $\bar{X} = 1/N \sum_{i=1}^{N} X_i$ is approximately normal with mean $\mu$ and variance $\sigma^2/N$ where $\mu$ and $\sigma^2$ are the mean and variance of the population distribution, respectively. In other words, the distribution of the random variable $Z = (\bar{X} - \mu)/\sigma(\sqrt{N})$ approaches $\mathcal{N}(0, 1)$ in the limit $N \to \infty$ regardless of the distribution of the underlying population. For values of $N \geq 30$, the distribution of $\bar{X}$ could be considered to be normal for most practical applications [10]. In this case, the 95% confidence interval for the mean $\bar{X}$ is given by $[\bar{X} - 1.96(\sigma/\sqrt{N}), \bar{X} + 1.96(\sigma/\sqrt{N})]$. When $N \geq 30$, the 95% confidence interval could be approximated by $[\bar{X} - 1.96(s/\sqrt{N}), \bar{X} + 1.96(s/\sqrt{N})]$ where $s^2$ is the sample variance [10]. The 95% confidence interval implies that all possible null hypotheses $\mu = \mu_0$ will be rejected at a level of significance 0.05 if $\mu_0$ lies outside the interval. In Table II the confidence interval is expressed as $\bar{X} \pm 1.96(s/\sqrt{N})$ for each of the performance parameters (where $N = 40$ in our case).

Figs. 17 and 18 depict the convergence curves of the GA, MCA-GA, RCA-GA, and SA-GA for the noise-free gray-scale images Boeing and CRT, whereas Figs. 19 and 20 depict the convergence curves of the four algorithms for the corresponding noisy images with noise variance $\sigma^2_n = 100$. 

---

**Fig. 21.** Segmented Boeing image resulting from the GA.

**Fig. 22.** Segmented Boeing image resulting from the MCA-GA.

**Fig. 23.** Segmented Boeing image resulting from the RCA-GA.

**Fig. 24.** Segmented Boeing image resulting from the SA-GA.
Figs. 21-36 show the segmentation results of each of the evolutionary algorithms on the noise-free gray-scale images whereas Figs. 37-44 depict the segmentation results of each of the evolutionary algorithms on the noisy Boeing and CRT images with noise variance \( \sigma^2_n = 100 \).

The segmentation results (Figs. 21-44), the convergence curves (Figs. 17-20), and the performance statistics (Table II) prompt us to make the following observations: As explained above, these observations are statistically significant at a level of significance 0.05.

i) In general, the hybrid evolutionary algorithms, yielded better segmentation results than the GA. The final cost achieved by the hybrid evolutionary algorithms is lower than that achieved by the GA (Table II). This is particularly true of noisy images, implying thereby that the hybrid evolutionary algorithms are more robust to noise than the GA. The visual quality of the segmented images generated by the hybrid evolutionary algorithms is much better than that generated by the GA (Figs. 21-44). This underscores our earlier conjecture (Section III-D) that the incorporation of a stochastic hill-climbing mechanism in the GA would alleviate some of its major shortcomings and improve its performance. It is also to be noted that the 95% confidence intervals associated with the performance parameters of the GA are larger than those of the corresponding performance parameters of the hybrid evolutionary algorithms. This trend is amplified with increasing levels of noise in the image. This shows that the incorporation of a stochastic hill-climbing mechanism in the GA reduces the degree of uncertainty in the final result, especially in the case of noisy images.

ii) Of the three hybrid evolutionary algorithms, the RCA-GA was found to have the fastest convergence rate and the SA-GA the slowest in terms of CPU time. This could be partly attributed to the fact that the stochastic hill-climbing mechanism in SA (or SA-GA) entails the computation of an exponential function (3) and (4) whereas in the case of the RCA (or RCA-GA) only an addition and comparison are needed. The MCA-GA was found to be slightly slower than the RCA-GA. This could be attributed to the fact that although the stochastic hill-climbing mechanism of
the MCA (or MCA-GA), like that of the RCA (or RCA-GA), needs only an addition and comparison there is an extra overhead of having to update the demon energy at each iteration.

iii) Of the three hybrid evolutionary algorithms, the SA-GA performed the best in terms of the visual quality and the cost value of the final segmentation. This can be attributed to the fact that the Metropolis function (3) or the Boltzmann function (4) used in SA (or SA-GA) results in a superior stochastic hill-climbing mechanism that is capable of asymptotically approaching a Markov chain of solution states with a stationary Boltzmann distribution with greater accuracy than either MCA (or MCA-GA) or the RCA (or RCA-GA). Recall that the stationary Markov chain property plays a critical role in the convergence of all the three stochastic optimization algorithms: SA, MCA, and the RCA. Also note that the 95% confidence intervals associated with the performance parameters of the SA-GA are smaller than those associated with the corresponding parameters of the MCA-GA or the RCA-GA. This is particularly true in the case of noisy images. This could be attributed to the superior stochastic hill-climbing mechanism used in SA (or SA-GA) which increases the predictability of the final result.
VI. CONCLUSIONS

The problem of image segmentation was tackled from the viewpoint of combinatorial optimization. This paper explored the use of evolutionary algorithms, in particular the genetic algorithm, in the context of image segmentation. The problem was cast as one of cost function minimization where the cost function incorporated both region contour or edge information and region gray-scale uniformity which was quantified using gray-level variance values. The desired segmentation was deemed to be the one that corresponded to a global minimum of the cost function.

The GA was seen to suffer from certain inherent drawbacks in the context of image segmentation. With a view toward alleviating these shortcomings, the paper considered the use of hybrid evolutionary algorithms that combined the stochastic hill-climbing and asymptotic convergence properties of stochastic annealing algorithms with the building blocks property of the GA.

The paper examined three hybrid evolutionary algorithms in the context of image segmentation via cost function:
- SA-GA, a combination of the GA and SA originally proposed by Mahfoud and Goldberg [34];
- MCA-GA, a combination of the GA and MCA proposed in this paper;
- RCA-GA, a combination of the GA and RCA proposed in this paper.
Fig. 37. Segmented noisy Boeing image ($\sigma_n^2 = 100$) resulting from the GA.

Fig. 40. Segmented noisy Boeing image ($\sigma_n^2 = 100$) resulting from the SA-GA.

Fig. 38. Segmented noisy Boeing image ($\sigma_n^2 = 100$) resulting from the MCA-GA.

Fig. 41. Segmented noisy CRT image ($\sigma_n^2 = 100$) resulting from the GA.

Fig. 39. Segmented noisy Boeing image ($\sigma_n^2 = 100$) resulting from the RCA-GA.

Fig. 42. Segmented noisy CRT image ($\sigma_n^2 = 100$) resulting from the MCA-GA.
The hybrid evolutionary algorithms were found to outperform the GA in the context of segmentation of gray-scale images in terms of both the visual quality of the final segmentation and execution time thus underscoring the advantages of hybrid evolutionary algorithms.

Future research will investigate various refinements of the basic GA operators in the context of image segmentation. In particular, mutation operations over a connected region of pixels in the label array (instead of mutations on individual pixels) and crossover operators that cut the label array along edges already found (instead of randomly chosen locations) will be investigated. Other problems in computer vision such as shape from stereo, motion analysis, and image segmentation will also be investigated.

ACKNOWLEDGMENT

The authors wish to thank the anonymous reviewers for their insightful comments and helpful suggestions which greatly improved the paper.


Suchendra M. Bhandarkar (S’84–M’89) received the B.Tech. degree in electrical engineering from the Indian Institute of Technology, Bombay, India, in 1983 and the M.S. and Ph.D. degrees in computer engineering from Syracuse University, Syracuse, NY, in 1985 and 1989, respectively.

He was a Syracuse University Fellow for the academic years 1986–1987 and 1987–1988. He is currently an Associate Professor and the Director of the Visual and Parallel Computing Laboratory (VPCL) in the Department of Computer Science at the University of Georgia, Athens, where he has been a faculty since 1989. He has conducted and directed research in computer vision, multimedia systems, image processing, artificial intelligence and parallel algorithms and parallel architectures for computer vision, multimedia systems and computational genomics. He has more than 70 published research articles in these areas including 30 articles in peer-reviewed archival journals. He has also coauthored the book 3-D Object Recognition from Range Images (New York: Springer-Verlag, 1992).

Dr. Bhandarkar is a member of AAIA, ACM, and SPIE. He is also a member of Phi Kappa Phi and Phi Beta Delta.

Hui Zhang received the B.S. degree in computer science from the University of Science and Technology, China, in 1986 and the Ph.D. degree in biochemistry from the University of Georgia, Athens, in 1995. He also received the M.S. degree in computer science from the University of Georgia in 1996.

From 1990 to 1995, he was a Research Assistant in the Department of Chemistry at the University of Georgia where he conducted research in computational modeling and simulation of protein folding. From 1995 to 1996, he was a Teaching Assistant in the Department of Computer Science at the University of Georgia where he instructed undergraduate students and also conducted research on heuristic optimization algorithms for image segmentation. Since August 1996, he has been with DataStream Systems Inc., Greenville, SC, where he is presently a Software Engineer engaged in the development of software for enterprise-wide client/server maintenance and facility management. His research interests are in heuristic optimization algorithms, protein folding simulation, and image segmentation.