Applied Intelligence 11, 187–212 (1999)

Evolutionary Approaches to Figure-Ground Separation*

SUCHENDRA M. BHANDARKAR AND XIA ZENG
Department of Computer Science, University of Georgia, Athens, Georgia 30602-7404, USA
suchi@cs.uga.edu
xzeng@atl.dfi-aeronomics.com

Abstract. The problem of figure-ground separation is tackled from the perspective of combinatorial optimization. Previous attempts have used deterministic optimization techniques based on relaxation and gradient descent-based search, and stochastic optimization techniques based on simulated annealing and microcanonical annealing. A mathematical model encapsulating the figure-ground separation problem that makes explicit the definition of shape in terms of attributes such as cocircularity, smoothness, proximity and contrast is described. The model is based on the formulation of an energy function that incorporates pairwise interactions between local image features in the form of edgels and is shown to be isomorphic to the interacting spin (Ising) system from quantum physics. This paper explores a class of stochastic optimization techniques based on evolutionary algorithms for the problem of figure-ground separation. A class of hybrid evolutionary stochastic optimization algorithms based on a combination of evolutionary algorithms, simulated annealing and microcanonical annealing are shown to exhibit superior performance when compared to their purely evolutionary counterparts and to classical simulated annealing and microcanonical annealing algorithms. Experimental results on synthetic edgel maps and edgel maps derived from gray scale images are presented.

Keywords: figure-ground separation, evolutionary computation, genetic algorithm, simulated annealing, microcanonical annealing

1. Introduction

The problem of figure-ground separation is one of fundamental importance in computer vision. The preattentive capability of human vision in being able to effortlessly separate figure from ground has yet to be emulated in state-of-the-art computer vision systems. In order to render the problem tractable, it is necessary to clearly define the shapes of interest (i.e., what exactly constitutes figure) and what constitutes extraneous noise (i.e., the ground). It is also imperative to define a computationally feasible and efficient procedure that is capable of separating the shapes of interest from shapes that could have potentially arisen due to extraneous noise.

In this paper, the problem of figure-ground separation is treated as a combinatorial optimization problem. The shape and noise elements and their spatial interactions are modeled as an interacting spin (Ising) system from quantum physics [1]. In conformity with the Ising model, an energy function is defined over the image elements (i.e., both shape and noise elements). The energy function serves to reinforce the grouping of local shape elements that represent objects of possible interest into global shapes and also simultaneously eliminate noise elements. The figure-ground separation problem thus becomes a combinatorial optimization problem where the global minimum of the energy function corresponds to the optimal separation (i.e., classification) of the image elements into figure and noise elements.

Stochastic optimization techniques (also known as Monte Carlo techniques) have proved very successful at solving global optimization problems. In particular, simulated annealing [2] and microcanonical annealing [3] have been shown to be effective in solving global optimization problems that are known to be NP-complete

*This research was supported in part by the University of Georgia Research Foundation, Athens, Georgia.

or NP-hard in a wide variety of application domains such as VLSI design [4–8], graph theory [9, 10], computational genetics [11, 12], operations research [13], neural computing [14] and image processing [15, 16]. The primary advantage of these stochastic optimization techniques is that they are capable of eluding local optima in the search space while maintaining asymptotic convergence towards the global optimum. This is in contrast to deterministic optimization techniques such as hill-climbing search and dynamic programming that exhibit a tendency to get trapped in a local optimum. Both, simulated annealing and microcanonical annealing emulate the physical process of annealing or gradual cooling used to create highly crystalline solids (in a very low energy state) from their molten (high energy) state. Mean field annealing [17], which is a deterministic approximation to simulated annealing, has also been used successfully in solving global optimization problems such as NP-complete graph-theoretic problems [18–20], image segmentation [21], visual reconstruction [22] and edge detection [23].

This paper explores a more recent class of stochastic optimization techniques termed as *evolutionary* algorithms. Evolutionary algorithms emulate the process of biological evolution which is based on the Darwinian principle of natural selection, popularly known as *survival of the fittest*. Of specific interest in the wider class of evolutionary algorithms, is the genetic algorithm in which a candidate solution to the problem is encoded in the form of a *chromosome*. From a population of *chromosomes*, using operators such as *selection*, *crossover* and *mutation*, which are designed to emulate their real-life genetic counterparts, the genetic algorithm explores the search space for a globally optimum solution by producing successive generations of chromosomes. This paper proposes a class of *hybrid* evolutionary stochastic optimization algorithms that combine the strengths of annealing-based techniques with those of genetic algorithms and alleviate their individual weaknesses, thus resulting in performance that is superior to that of either class of techniques used in isolation.

The remainder of the paper is organized as follows: In Section 2, a review of previous research in figure-ground separation is presented. In Section 3, an Ising model that encapsulates the constraints of the figure-ground separation problem is described. In Section 4, the genetic algorithm, simulated annealing algorithm and microcanonical annealing algorithm are described in the context of the figure-ground separation problem. In Section 5, hybrid evolutionary stochastic optimization algorithms for figure-ground separation are proposed and described. In Section 6, experimental results on synthetic edgel maps and edgel maps derived from gray-scale images are presented. In Section 7, the paper is concluded and future research directions are cited.

2. Review of Previous Work

Figure-ground separation was first studied by Gestalt psychologists [24] in their research on perceptual grouping where certain image elements are organized to construct an emergent figure. Researchers in computer vision and image processing have studied figure-ground separation from the viewpoint of edge/contour grouping where short edge segments or *edgels* need to be grouped into long continuous contours of perceptual significance. Conventional techniques for edgel grouping tend to suffer from two significant drawbacks: First, an analytic description of the underlying curve is often used. This clearly calls for a priori knowledge of the objects and the resulting features in the image. Since figure-ground separation is the first step in the initial domain-independent segmentation of the image, such a priori knowledge is seldom available. Second, the notion of noisy edgel data is often not clearly defined. Noise elimination techniques tend to be ad hoc where low-contrast edgels are often discarded as noisy edgels. Since it is unreasonable to expect all edgels belonging to an object to be high-contrast edgels, this approach tends to discard some of the edgels that describe object shape along with the noisy pixels.

Parent and Zucker [25] characterize local edgel shape on the basis of curvature computed on a local grid. They cast the curve inference problem as one of global optimization and use a relaxation labeling algorithm to compute the optimum. However, relaxation is a local search-based optimization process that is vulnerable to the presence of local optima in the search space and hence needs good initialization. A similar approach can be found in Sha'ashua and Ullman [26] where curve inference is modeled as search for the best sequence of edge elements that would result in the longest and smoothest image curves. The search itself is carried out using dynamic programming which is also a deterministic optimization process that is prone to get trapped in a local optimum and hence calls for good initialization. In Gutfinger and Sklansky [27] curve/noise separation is viewed as a classification problem where

the classification is done using a method that uses supervised and unsupervised training. Their technique though theoretically appealing, is impractical on real image data.

Sejnowski and Hinton [28] showed the limitations of using deterministic optimization techniques in their formulation of the figure-ground separation problem. In their formulation, the image elements are classified into two possible labels: region and noise. With an initial random labeling the gradient descent procedure is seen to get trapped in one of several local optima of the energy function whereas simulated annealing converges to an optimal solution in which the region elements are bounded by the edge elements. Carnevali et al. [29] use simulated annealing in conjunction with a pixel interaction model to classify pixels in a binary image as object or noise. Peterson [30] has applied mean field theory to the problem of tracking particles in high-energy physics which can be shown to be similar to the grouping problem. However, Peterson's model is not suitable for noisy data and results in n^2 variables and n^4 connections for n point data. Blake [31, 32] has used simulated annealing in the context of visual reconstruction of surface data. Tan et al. [33, 34] have explored local search and simulated annealing in the context of edge detection and edgel grouping. Bhandarkar et al. [35] have applied the genetic algorithm and Acton and Bovik [36] the mean field annealing algorithm to the problem of edge detection and edgel grouping. Roth and Levine [37] have applied a genetic algorithm based on a minimal subset representation of a geometric primitive to the problem of feature extraction. Their technique however, requires that the geometric primitive have an underlying parametric representation which restricts its applicability. Herault and Horaud [1] have explored simulated annealing, mean field annealing and microcanonical annealing in the context of figure-ground separation via edgel grouping. The mathematical model used to represent the figure-ground separation problem is shown to fit the constraints of an Ising model. The results presented in their paper were impressive and brought out the advantages of using the Ising model in conjunction with global optimization techniques for the figure-ground separation problem.

This paper extends the first author's previous work in evolutionary algorithms [35] and the recent work of Herault and Horaud [1]. In particular, this paper explores a class of evolutionary algorithms based on the genetic algorithm and also proposes a class of hybrid evolutionary stochastic optimization algorithms that combine the strengths of annealing-based techniques with those of genetic algorithms while alleviating their individual shortcomings.

3. Mathematical Formulation of the Problem

Herault and Horaud [1] have proposed an Ising model for the figure-ground separation problem. They treat the figure-ground separation problem as one in which figure edgels (i.e., straight and short edge segments) are to be separated from noise edgels. The mathematical formulation of the figure-ground separation problem addressed in this paper, is based largely on the Ising model of Herault and Horaud. However, in the interest of making this paper self-contained, we present a brief synopsis of the Ising model. The interested reader is referred to [1] for a more detailed exposition.

3.1. *The Ising Model*

The Ising model is commonly used in quantum physics to explain electro-magnetic phenomena. The state of the Ising system is described by a spin state vector consisting of N elements $\sigma = [\sigma_1, \sigma_2, \ldots, \sigma_N]$ such that $\sigma_i \in \{+1, -1\}$, i.e., each spin is described by a discrete label *up* $(+1)$ or *down* (-1). A symmetric matrix $\mathbf{J}$ describes the interaction between spins. Element $J_{i,j}$ describes the interaction between spins σ_i and σ_j. We require that $J_{i,i} = 0$ for all $1 \leq i \leq N$. A vector $\mathbf{B} = [B_1, B_2, \ldots, B_N]$ describes the external field that the Ising system is subject to where B_i is value of the field viewed by spin σ_i. The energy function associated with an Ising system subject to an external field $\mathbf{B}$ is given by:

$$E(\sigma_1, \sigma_2, \ldots, \sigma_N) = -\frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} J_{i,j} \sigma_i \sigma_j - \sum_{i=1}^{N} B_i \sigma_i \tag{1}$$

The *ground state* of the Ising model is the one that results in the minimization of the energy E.

3.2. *Figure-Ground Separation and the Ising Model*

The goal of figure-ground separation, as treated in this paper, is to group figure edgels into continuous shape contours and simultaneously reject noise edgels. The requirement is that figure edgels interact in a manner

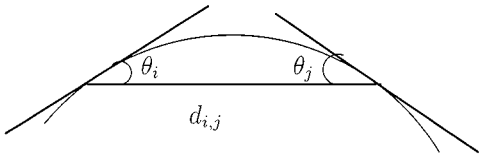

Figure 1. Edgel interaction parameters.

such that they reinforce each other whereas noise edgels interact in a manner such that they nullify each other. Four types of interaction between edgels are modeled based on very *generic* notions of desired shapes: cocircularity, smoothness, proximity, and contrast. Incorporation of generic shape properties spares us from having to make very constraining assumptions about the shapes that underlie the figure edgels.

Cocircularity. Two edgels are considered to be cocircular if they are tangent to the same circle at their respective edgel centers (Fig. 1). From Fig. 1, it is clear that for edgels i and j to be cocircular, the angles θ_i and θ_j have to be equal. Let $\delta_{i,j} = |\theta_i - \theta_j|$ where smaller values of $\delta_{i,j}$ imply a greater degree of cocircularity. The cocircularity coefficient is defined as:

$$c_{i,j}^{\text{COCIRC}} = \left(1 - \frac{\delta_{i,j}^2}{\pi^2}\right) \exp\left(-\frac{\delta_{i,j}^2}{k}\right) \tag{2}$$

where the parameter k is chosen such that $c_{i,j}^{\text{COCIRC}}$ vanishes gradually under conditions of noncircularity. If k is chosen to be large, the effect of $\delta_{i,j}$ will be deemphasized and a greater degree of noncircularity will be tolerated. A value of $k = 10$ was found to give good results in practice.

Parallelism. Two edgels i and j are deemed to be parallel if the sum of θ_i and θ_j equals π (Fig. 1). The parallelism coefficient is defined as:

$$c_{i,j}^{\text{PARA}} = \begin{cases} \cos\left(\dfrac{\pi|\pi - \theta_i - \theta_j|}{2\epsilon}\right) & \text{if } |\pi - \theta_i - \theta_j| \le \epsilon \\ 0 & \text{if } |\pi - \theta_i - \theta_j| > \epsilon \end{cases} \tag{3}$$

where ϵ is an angle threshold. A value of $\epsilon = 5^\circ$ was found to be a good choice in practice.

Smoothness. Two edgels i and j are deemed to be collinear or smooth if $\theta_i = \theta_j = 0$ (Fig. 1). The smoothness coefficient is defined as:

$$c_{i,j}^{\text{SMOOTH}} = \left(1 - \frac{\theta_i\theta_j}{\pi^2}\right)\left(1 - \frac{(\pi - \theta_i)(\pi - \theta_j)}{\pi^2}\right) \tag{4}$$

Proximity. Two edgels i and j are deemed to be proximate if the distance $d_{i,j}$ between their centers is small compared to the standard deviation σ_d of all the pairwise edgel distances in the edgel map. The proximity coefficient is defined as:

$$c_{i,j}^{\text{PROX}} = \exp\left(-\frac{d_{i,j}^2}{2\sigma_d^2}\right) \tag{5}$$

Contrast. If the average intensities of the two edgels are g_i and g_j then the contrast coefficient is defined as:

$$c_{i,j}^{\text{CONTRAST}} = \frac{g_i g_j}{g_{\max}^2} \tag{6}$$

where $g_{\max}$ is the maximum of the average intensities of all the edgels in the edgel map.

Intensity. Two edgels having the same average intensity g_i and g_j should reinforce each other. An intensity coefficient for this purpose is defined as follows:

$$c_{i,j}^{\text{INTY}} = \begin{cases} \cos\left(\dfrac{\pi|g_i - g_j|}{2\delta}\right) & \text{if } |g_i - g_j| \le \delta \\ 0 & \text{if } |g_i - g_j| > \delta \end{cases} \tag{7}$$

where δ is a threshold for intensity change. We have found $\delta = 5$ to give reliable results when the total number of gray levels is 256 (i.e., 1 byte per pixel).

The overall interaction coefficient is computed as:

$$c_{i,j} = \max\left(c_{i,j}^{\text{COCIRC}}, c_{i,j}^{\text{PARA}}\right) c_{i,j}^{\text{SMOOTH}} \times c_{i,j}^{\text{PROX}} \max\left(c_{i,j}^{\text{CONTRAST}}, c_{i,j}^{\text{INTY}}\right) \tag{8}$$

The rationale behind the individual shape–based interaction coefficients (Eqs. (2)–(7)) and the overall interaction coefficient $c_{i,j}$ (Eq. (8)) is to identify the desired *local* shape structure(s) resulting from suitably interacting edgel pairs in the edgel map while rejecting those edgels that do not contribute to desired *local* shape structure(s). The values of k, ϵ and δ in Eqs. (2), (3) and (7) respectively, were determined empirically after experiments on synthetic edgel maps and edgel maps derived from gray scale images. The same values

of k, ϵ and δ were used in all the experiments described in Section 6.

The energy function associated with the *orderliness* of the edgel map can be written as:

$$E(\sigma_1, \sigma_2, \ldots, \sigma_N) = -\frac{1}{2}\sum_{i=1}^{N-1}\sum_{j=i+1}^{N}(c_{i,j} - \alpha)(1 + \sigma_i\sigma_j + \sigma_i + \sigma_j) \tag{9}$$

Here, σ_i takes on a value of $+1$ or -1 depending on whether or not the corresponding edgel is labeled as *figure* (i.e., included in the final edgel map) or *noise* (i.e., excluded from the final edgel map). From Eq. (9) it is apparent that if $\sigma_i = -1$ then $(c_{i,j} - \alpha)$ will be excluded from E irrespective of the value of σ_j. Also note that α is a threshold value that is estimated from the signal-to-noise ratio (SNR). If the value of $c_{i,j}$ falls below the threshold α then the corresponding edgels are deemed to be weakly interacting and they result in an increase in the total energy E. Edgel maps with higher E values may be deemed to contain a large number of weakly interacting edgels. Thus the goal is to be able to find an edgel map with the lowest E value (i.e., containing edgels that maximally reinforce each other).

Comparing Eqs. (1) and (9) one can show that the Ising model underlying Eq. (9) is given by:

$$E(\sigma_1, \sigma_2, \ldots, \sigma_N) = C - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}J_{i,j}\sigma_i\sigma_j - \sum_{i=1}^{N}B_i\sigma_i \tag{10}$$

where

$$C = \frac{1}{4}\sum_{i=1}^{N}\sum_{j=1}^{N}(c_{i,j} - \alpha) \tag{11}$$

$$J_{i,j} = \frac{1}{2}(c_{i,j} - \alpha) \tag{12}$$

$$B_i = \frac{1}{2}\sum_{j=1}^{N}(c_{i,j} - \alpha) \tag{13}$$

In order to ensure that $J_{i,i} = 0,\ 1 \leq i \leq N$ in conformity with the Ising model, one chooses $c_{i,i} = \alpha, 1 \leq i \leq N$. The energy function of the edgel map is similar to the Ising model except for a constant bias term given by Eq. (11). The desired edgel map, (i.e., one in which the noise edgels are separated from the figure edgels), corresponds to the ground state of the Ising model represented by Eq. (10) (i.e., the state which minimizes the energy E given by Eq. (10)).

4. Evolutionary Algorithms for Figure-Ground Separation

The energy function E (Eq. (10)) associated with the *orderliness* of the edgel map is a multivariate combinatorial function whose landscape is fraught with several local minima. *Deterministic* combinatorial optimization algorithms based on local search of the energy landscape are prone to get trapped in one of the several local minima. Determining a global minimum of the energy function E (Eq. (10)) clearly entails the use of a *stochastic* combinatorial optimization procedure that is capable of forgoing the several local minima in favor of a global minimum. As already mentioned, stochastic combinatorial optimization techniques such as simulated annealing, microcanonical annealing and mean field annealing have been successfully applied to the Ising model representation of the figure-ground separation problem [1].

This paper intends to explore a class of stochastic combinatorial optimization techniques based on the paradigm of evolutionary computation in the context of the figure-ground separation problem. Evolutionary computation is a population-based optimization process that mimics the process of biological evolution encountered in nature [38]. Evolutionary computation has resulted in stochastic optimization techniques that outperform classical optimization techniques when applied to several real-world problems. This paper explores variations of the genetic algorithm (GA) [39–41] which is an important member of the wider class of evolutionary algorithms.

4.1. Genetic Algorithms—An Overview of Key Concepts

Central to the genetic algorithm (GA) are the concepts of *chromosome*, *population*, *fitness*, *selection*, *crossover* and *mutation*. A potential solution to a combinatorial optimization problem is represented as a bit string or *chromosome*. A collection of potential solutions or chromosomes constitutes a *population*. With each chromosome is attached a *fitness* value which is a measure of goodness of the corresponding solution. The fitness value is computed using a *fitness function*

Figure 2. Crossover operation.

which is derived from the objective function to be optimized and the constraints underlying the combinatorial optimization problem. The chromosomes from a given population are chosen using a *selection* operator to form a mating pool for reproduction. The *roulette wheel* selection operator which selects each chromosome with a probability proportional to the ratio of the fitness of the chromosome to the overall fitness of the population, is a popular choice. This ensures that the mating pool contains a higher percentage of fitter chromosomes. The *tournament* selection operator selects two members at random from the current generation and compares their fitness values. The fitter chromosome is then inserted in the mating pool. In the tournament selection operator, the selection is based on the *fitness rank* of the chromosome relative to the other chromosomes in the population rather than its actual *fitness value*.

Two mates, selected at random from the mating pool, reproduce via the *crossover* operator. During the crossover operator, a point along the length of the chromosome is selected at random and the ends of the chromosomes swapped with a predefined crossover probability to generate a pair of offspring for the next generation (Fig. 2). Each of the offspring is subject to random localized change via a *mutation* operator, which in our case amounts to flipping each bit of the offspring with a predefined mutation probability.

There are two principal variants of the genetic algorithm based on the population replacement strategy employed: the canonical genetic algorithm (CGA) and the steady state genetic algorithm (SSGA). In the CGA, the offspring created from the mating pool replace the entire current generation. The population replacement strategy or evolution strategy of the CGA is modeled along short-lived biological species such as insects where parents lay eggs and die. In the SSGA, on the other hand, only a few of the weakest members of the current population are replaced by the offspring created from the mating pool. The evolution strategy in the SSGA emulates the population replacement encountered in long-lived species where the parents and children often coexist at any given time. A more detailed comparison between the CGA and SSGA can be found in [42]. The population replacement is repeated for several iterations (i.e., generations) of the GA. The GA is deemed to have converged when the fitness value of the best chromosome in the population has not changed over the past k successive generations.

4.2. The GA and Figure-Ground Separation

In order to apply the GA to the figure-ground separation problem, a suitable chromosomal representation of the edgel map needs to be formulated. The edgel map is modeled as a bit string where the ith bit position represents the classification of the ith edgel in the edgel map; if the ith edgel is classified as a figure (noise) edgel then the ith bit in the bit string is 1 (0) and conversely. For an edgel map with N edgels, there are 2^N possible bit strings. The ideal edge map is one in which all the edgels have been correctly classified. The problem of determining this ideal string is NP-complete since the only algorithm that ensures an optimal solution is one that carries out an exhaustive search of the space of all the 2^N possible bit strings resulting in an exponential-time algorithm.

At every stage in the evolution process, the GA maintains a population of chromosomes where each chromosome represents an edgel map. The raw fitness value F for a chromosome is defined as $F = -E$ where E is the energy associated with the corresponding edgel map (Eq. (9)). This ensures that edgel maps with lower energy are associated with higher fitness values. In order to circumvent the problem of negative fitness values, the fitness values are normalized. The normalized fitness value F_n is given by:

$$F_n = F - F_{\min} + \beta(F_{\max} - F_{\min}) \qquad (14)$$

where $F_{\min}$ and $F_{\max}$ are the minimum and maximum F values respectively, in the current population. If $\beta \geq 0$, the value of F_n is positive for all chromosomes in the population. The value of β determines the selection pressure against weak chromosomes. The pressure is strongest when $\beta = 0$.

4.2.1. Advantages and Shortcomings of the CGA and SSGA. The advantages of the CGA-based and the SSGA-based figure-ground separation algorithms are:

(a) The selection operator and crossover operator 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.
(b) The CGA and SSGA are naturally parallel. In fact it has been shown that the CGA and the SSGA exhibit both *explicit* and *implicit* parallelism [40]. Implicit parallelism arises from the fact that by evaluating a certain chromosome, the GA simultaneously and implicitly evaluates all the schema of which the chromosome is an instance. Explicit parallelism can be attributed to the fact that the selection, crossover and mutation operators can be performed in parallel over all the chromosomes in the population.
(c) The population of candidate solutions enables one to explore a diversity of solutions and hence a larger fraction of the search space. The Schema Theorem [39, 40] enables the GA to sample a large fraction of the search space even with a relatively small population size. This increases the chances of the GA being able to arrive at a globally optimal solution.

Some of the problems with the CGA- and SSGA-based figure-ground separation techniques are:

(a) The performance (especially that of the CGA) is extremely sensitive to the manner in which the chromosome is encoded. It is crucial that strongly interacting edgels have their corresponding bits positioned very close to each other on the chromosome and vice versa.
(b) The results are sensitive to the value of α especially when the encoding order is random. The value of α corresponding to the best observed result changes with the encoding scheme used.
(c) In the absence of a hill-climbing mechanism, the number of generations (and hence the execution time) needed for convergence is fairly large.
(d) With the incorporation of a deterministic hill-climbing mechanism the CGA and SSGA exhibit premature convergence to a suboptimal solution [35].

In order to alleviate some of the aforementioned shortcomings, we explore the possibilities of combining GA-based techniques with stochastic hill-climbing techniques such as simulated annealing and/or microcanonical annealing.

4.3. Stochastic Annealing-based Algorithms for Figure-Ground Separation

Stochastic annealing algorithms such as simulated annealing (SA) [2] and microcanonical annealing (MCA) [3], are a subcategory 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 $\mathbf{x}_i$ to a multivariate objective function $E(\mathbf{x})$, which is to be minimized, is systematically perturbed to yield another candidate solution $\mathbf{x}_j$. In the evaluate phase, $E(\mathbf{x}_j)$ is computed. In the decide phase, $\mathbf{x}_j$ is accepted and replaces $\mathbf{x}_i$ *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 difference between SA and MCA arises from the difference 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 [14]. The Boltzmann distribution asymptotically converges to a globally optimal solution when subject to the annealing process [16].

4.3.1. Simulated Annealing. In the decide phase of the classical SA algorithm, the new candidate solution $\mathbf{x}_j$ is accepted with probability p which is computed using the Metropolis function [43]

$$p = \begin{cases} 1 & \text{if } E(\mathbf{x}_j) < E(\mathbf{x}_i) \\ \exp\left(-\dfrac{E(\mathbf{x}_j) - E(\mathbf{x}_i)}{T}\right) & \text{if } E(\mathbf{x}_j) \geq E(\mathbf{x}_i) \end{cases} \tag{15}$$

or using the Boltzmann function $B(T)$ [14]

$$p = B(T) = \frac{1}{1 + \exp\left(\frac{E(\mathbf{x}_j) - E(\mathbf{x}_i)}{T}\right)} \tag{16}$$

at a given value of temperature T, whereas $\mathbf{x}_i$ 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 Eqs. (15) and (16), 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 local 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. Geman and Geman [16] 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 $k \to \infty$.

4.3.2. Microcanonical Annealing. The classical MCA algorithm 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 $E(\mathbf{x})$ to be minimized whereas the kinetic energy $E_k > 0$ 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) serve(s) to provide the system with an extra degree (or degrees) of freedom enabling MCA to escape from local minima.

In the decide phase of MCA, if $E(\mathbf{x}_j) < E(\mathbf{x}_i)$ then $\mathbf{x}_j$ is accepted as the new solution. If $E(\mathbf{x}_j) \geq E(\mathbf{x}_i)$ then $\mathbf{x}_j$ is accepted as the new solution only if $E_k \geq E(\mathbf{x}_j) - E(\mathbf{x}_i)$. If $E(\mathbf{x}_j) \geq E(\mathbf{x}_i)$ and $E_k < E(\mathbf{x}_j) - E(\mathbf{x}_i)$ then the current solution $\mathbf{x}_i$ is retained. In the event that $\mathbf{x}_j$ is accepted as the new solution, the kinetic energy demon is updated $E_k^{n+1} = E_k^n + [E(\mathbf{x}_i) - E(\mathbf{x}_j)]$ in order 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 [44].

4.3.3. Advantages and Shortcomings of Stochastic Annealing-Based Techniques. The stochastic annealing-based approaches like the classical SA and classical MCA algorithms have the following advantages:

(a) The stochastic hill climbing mechanism in SA and MCA guarantees asymptotic convergence to a global optimum [3, 16].
(b) The performance of both, SA and MCA is resilient to the manner in which the candidate solutions to the problem are encoded.

However, both classical SA and classical MCA are seen to suffer from the following major drawbacks:

(a) Since the perturbation operation in the classical SA and MCA algorithms is typically local, the search procedure is fairly localized. Thus, SA and MCA typically do not explore the same diversity of solutions that GAs do. This causes the annealing schedule needed for asymptotic convergence to a globally optimum solution to be computationally intensive.
(b) Since the techniques are based on generating an asymptotically ergodic Markov chain from an initial starting state, they are inherently serial. Attempts to parallelize classical SA and classical MCA have met with limited success primarily because the parallel algorithms, in their attempt to maximize speedup and efficiency of processor utilization, invariably compromise on the ergodic Markov chain property and hence on the convergence characteristics of the classical SA and MCA algorithms.
(c) The classical SA and MCA algorithms do not exploit previously encountered *good* subsolutions in their future explorations of the search space. Since the Markov chain of solution states is strictly of the first order, the next state is dependent only on the present state. As a consequence, the classical SA and MCA algorithms do not incorporate the *building-blocks* property that GAs do.

5. Hybrid Evolutionary Stochastic Optimization Algorithms for Figure-Ground Separation

Our efforts towards designing hybrid evolutionary stochastic optimization algorithms for figure-ground separation were motivated by the desire to combine the

advantages and alleviate the shortcomings of both evolutionary and stochastic annealing-based approaches outlined in Sections 4.2.1 and 4.3.3. In a sense, the stochastic annealing techniques could be looked upon as single element population genetic algorithms with a stochastic hill-climbing-based mutation operator. There are two broad ways in which the evolutionary and stochastic annealing-based algorithms could be combined:

1. *GA with SA-like or MCA-like operators*: The overall structure of the hybrid algorithm is like the CGA or SSGA. However, the probabilities assigned to the individual GA operators such as crossover and mutation are annealed in a manner similar to the SA and MCA [45]. However, such a hybrid algorithm, like the GA, does not guarantee asymptotic convergence.
2. *An SA or MCA with GA-like operators*: The SA or MCA 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 SA or MCA algorithm. The population replacement strategy incorporates a Boltzmann tournament between the parents and the offspring where the Boltzmann selection function is annealed in a manner similar to SA or MCA [46]. Such an algorithm can be shown to retain the asymptotic convergence properties of the SA or MCA algorithm [46] 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.

5.1. *A Hybrid CGA-SA Algorithm for Figure-Ground Separation*

A hybrid algorithm for figure-ground separation that combines the CGA and SA algorithms is termed as the CGA-SA algorithm (Fig. 3) and modeled along the algorithm presented by Mahfoud and Goldberg [46]. At each temperature value, from the current generation of $2n$ edgel maps, a mating pool of size $2n$ is created using either roulette-wheel or tournament selection. For each pair of parents P_{2i} and P_{2i+1} from the mating pool, a pair of offspring C_{2i} and C_{2i+1} is created using the CGA crossover and mutation operators. Offspring C_{2i} replaces parent P_{2i} with probability $1 - B(T)$ where $B(T)$ is the Boltzmann function given by:

$$B(T) = \frac{1}{1 + \exp\left(\frac{E(P_{2i}) - E(C_{2i})}{T}\right)} \tag{17}$$

Here, $E(P_{2i})$ and $E(C_{2i})$ denote the energy values associated with the edgel maps corresponding to chromosomes P_{2i} and C_{2i} respectively. Parent P_{2i+1} is

```
T = T_init; {Initialize the temperature parameter to a high value}
G = G_init; {Start with an initial population of 2n edgel maps}
repeat
  {
    From present generation G form a mating pool of size 2n using
    tournament selection or 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 CGA crossover operator;

      Bring about a random localized change in each offspring using a CGA
      mutation operator;

      Replace parent P[2i] in G with offspring C[2i] with probability 1 - B(T);
      /* where B(T) is the Boltzmann function */

      Replace parent P[2i+1] in G with offspring C[2i+1] with probability 1 - B(T);
    }
    T = A(T); /* Reduce temperature using the annealing function */
  } until the convergence criterion is met;
```

Figure 3. Outline of the hybrid CGA-SA algorithm.

```
T = T_init; {Initialize the temperature parameter to a high value}
G = G_init; {Start with an initial population of n edgel maps}
repeat
 {
   repeat
   {
     Select two members P_1 and P_2 from the current generation G
     using roulette-wheel selection or tournament selection;

     Create two offspring C_1 and C_2 using the CGA crossover operator;

     Bring about a random localized change in each offspring using a CGA
     mutation operator;

     If neither offspring exists in the current generation then
        replace the weakest member P in the current generation by
        the stronger of the two offspring C with probability 1 - B(T);
     Else
        replace the weakest member P in the current generation by
        the offspring C that does not already exist in the current
        population with probability 1 - B(T);

    } until a certain predefined percentage of the weakest members in
      the current generation have been replaced;
  T = A(T); /* Reduce the temperature using the annealing function */
 } until the convergence criterion is met;
```

Figure 4. Outline of the hybrid SSGA-SA algorithm.

replaced by the offspring C_{2i+1} in a similar manner. As can be seen from Eq. (17), at very high temperatures the replacement is purely random since the probability of replacement is 50% irrespective of the energy values of the parent and offspring. At lower temperatures, the replacement resembles hill-climbing since there is a greater preference for the chromosome (either parent or offspring) with lower energy. The convergence criterion could be GA-based where the algorithm is presumed to have converged to a globally optimal solution if the energy value of the fittest member in the population has not changed for a predetermined number of successive generations. Alternatively, the convergence criterion could be SA-based where the algorithm is presumed to have converged to a globally optimal solution if the temperature reaches a certain final value.

5.2. *A Hybrid SSGA-SA Algorithm for Figure-Ground Separation*

The hybrid SSGA-SA algorithm proposed in this paper is similar to the CGA-SA algorithm except that at each generation, only p of the least fit members of the current population are replaced. The pseudocode description of the SSGA-SA algorithm is given in Fig 4. At each temperature value, from the present generation of n edgel maps, a pair of chromosomes P_1 and P_2 are selected for mating. Of the resulting offspring C_1 and C_2, one that is fitter and not present in the current generation (denoted by C) is selected and compared with the least fit member P in the current generation. P is replaced by C with probability $1 - B(T)$ where

$$B(T) = \frac{1}{1 + \exp\left(\frac{E(P)-E(C)}{T}\right)} \tag{18}$$

As in the case of the CGA-SA algorithm, the replacement is purely random at higher temperatures whereas at lower temperatures the replacement asymptotically favors the fitter member with unit probability. Here too, both GA-based and SA-based convergence criteria are possible.

5.3. *A Hybrid CGA-MCA Algorithm for Figure-Ground Separation*

The hybrid CGA-MCA algorithm proposed in this paper, treats the entire population as a microcanonical

ensemble, that is, a system that is thermally insulated from its surroundings. The *entire* population has a kinetic energy demon E_k associated with it. The potential energy E assigned to each chromosome in the population is the energy associated with the corresponding edgel map (Eq. (10)). The CGA-MCA algorithm proceeds in a manner identical to the CGA-SA algorithm.

A mating pool of potential parents is generated using a roulette-wheel or tournament selection procedure. The kinetic energy demon E_{km} associated with the mating pool is chosen to conserve the total energy, i.e., $E_{\text{km}} = \sum_{\text{pop}} E - \sum_{\text{mp}} E + E_k$ where $\sum_{\text{pop}} E$ is the total potential energy associated with the population and $\sum_{\text{mp}} E$ the total potential energy associated with the mating pool. Two parents P_1 and P_2 are chosen at random from the mating pool and two offspring chromosomes C_1 and C_2 are created using the CGA crossover operator. The offspring C_1 and C_2 are also subject to random local change using the CGA mutation operator. The offspring C_1 is compared with parent P_1. If E_{C_1}, i.e., the potential energy associated with C_1 is lower than E_{P_1}, i.e., the potential energy associated with P_1, then C_1 is added to the next generation and E_{km} is *increased* by $E_{P_1} - E_{C_1}$. If $E_{C_1} \geq E_{P_1}$ and $E_{\text{km}} \geq E_{C_1} - E_{P_1}$ then C_1 is added to the next generation else P_1 is added. In the event that C_1 is added to the next generation, E_{km} is *decreased* by $E_{C_1} - E_{P_1}$. The same procedure is carried out in the case of parent P_2 and offspring C_2. For the newly created generation, the kinetic energy demon E_k is initialized such that $E_k = E_{\text{km}}$. The value of E_k is reduced slowly using an annealing function similar to the one used in the MCA algorithm. The pseudocode description of the CGA-MCA algorithm is given in Fig. 5.

5.4. *A Hybrid SSGA-MCA Algorithm for Figure-Ground Separation*

The hybrid SSGA-MCA approach proposed in this paper, treats the entire population as a microcanonical ensemble just as the CGA-MCA algorithm does. However, in the case of the SSGA-MCA algorithm, each member in the population has a kinetic energy demon associated with it. As in the case of the CGA-MCA algorithm, the potential energy E assigned to each chromosome in the population is the energy associated with the corresponding edgel map (Eq. (10)).

The SSGA-MCA algorithm is similar to the CGA-MCA hybrid algorithm except that at each generation, only p of the least fit members of the population are replaced. The pseudocode description of the SSGA-MCA algorithm is given in Fig. 6. From the present generation of n edgel maps, a pair of chromosomes P_1 and P_2 is selected for mating using either the roulette wheel or tournament selection procedure. Of the resulting offspring C_1 and C_2, one that is fitter and not present in the current generation (denoted by C) is selected and compared with the least fit member P in the current generation. P is replaced by C if E_C (the potential energy associated with C) is lower than E_P (the potential energy associated with P), or $E_{\text{kp}} \geq E_C - E_P$ where E_{kp} is the kinetic energy associated with P. E_{kc}, the kinetic energy associated with C, is initialized such that $E_{\text{kc}} = E_{\text{kp}} + E_P - E_C$, i.e., the sum of the kinetic energy and potential energy is conserved. This procedure is carried out until p of the weakest members in the original population are replaced. The value of the kinetic energy demon is then slowly reduced using an annealing function similar to the one used in the MCA algorithm.

6. Experimental Results

The various evolutionary algorithms, CGA, SSGA, CGA-SA, SSGA-SA, CGA-MCA, and SSGA-MCA, were tested on synthetic edgel maps as well as on edgel maps derived from gray-scale images. Three of the synthetic edgel maps are shown in Figs. 7–9 and are referred to as *Synthetic-1*, *Synthetic-2* and *Synthetic-3* respectively. *Synthetic-1* contains 96 figure edgels and 160 noise edgels, *Synthetic-2* contains 96 figure edgels and 96 noise edgels and *Synthetic-3* contains 100 figure edgels and 120 noise edgels. The noise edgels have lengths that are uniformly distributed in the range $[0, 40]$ pixels, intensities uniformly distributed in the range $[0, 255]$, orientations uniformly distributed in the range $[0, \pi]$, and x and y coordinates of the edgel center uniformly distributed in the range $[0, 511]$ for an image of size 512×512 pixels.

Two of the gray-scale images *Saturn* and *Spaceship* are shown in Figs. 10 and 11 respectively and the corresponding edgel maps in Figs. 12 and 13 respectively. The edgel maps for the gray-scale images were generated using a Canny edge detector [47] and a simple edgel following algorithm proposed by Nalwa and Binford [48] that tracks edge points in the direction of the local gradient. The edge detector is computationally efficient since it is a local window-based operator. The edgel following algorithm of Nalwa and Binford is also computationally efficient since it employs a local

```
G = G_init; /* Start with an initial population of 2n edgel maps */
Ek = Ek_max;
/* Initialize the kinetic energy demon of the population to very high value */
repeat
  {
    Create a mating pool of 2n chromosomes using roulette-wheel
    selection or tournament selection;

    Ek_mat = E_pop + Ek - E_mat; /* E_pop is the total potential
    energy of the population and E_mat is the total potential energy and
    Ek_mat the kinetic energy demon of the mating pool */

    for (i = 0; i < n; i++)
    {
     From parents P[2i] and P[2i+1] in G generate two offspring C[2i]
     and C[2i+1] using the CGA crossover operator;

     Subject C[2i] and C[2i+1] to the CGA mutation operator;

     (1) if (E_C1 < E_P1) /* E_P1 and E_C1 are the potential energies
                          of  P[2i] and C[2i] respectively */
         {
          (1.a) add C[2i] to the next generation G_next;
          (1.b) Ek_mat = Ek_mat + (E_P1 - E_C1);
         }

     (2) else if ((E_C1 - E_P1) <  Ek_mat)
         {
          (2.a) add C[2i] to the next generation G_next;
          (2.b) Ek_mat = Ek_mat - (E_C1 - E_P1);
         }
      Repeat steps (1) and (2) with  P[2i+1] and C[2i+1];
     }

     G = G_next;
     Ek = Ek_mat;
     Ek = A(Ek); /* Reduce demon kinetic energy using the annealing function */
  } until the convergence criterion is met;
```

Figure 5. Outline of the hybrid CGA-MCA algorithm.

(i.e., greedy) search. Thus the time taken by the edgel preprocessing phase (consisting of edge detection and edgel following) in the case of gray scale images represents a negligible fraction of the total time taken by the figure-ground separation procedure.

6.1. *Performance of the CGA and SSGA*

The performance of the CGA and SSGA were compared using the synthetic edgel maps. The population size was chosen to be 100, the crossover probability to be 0.7, the mutation probability to be 0.05 and the halting criteria to be the fact that the best member in the population had not changed over the past 5 consecutive generations. For the purpose of comparison, a figure of merit M was defined as

$$M = \frac{1}{1 + \gamma \frac{F_d}{F} + \delta \frac{N_r}{N}} \tag{19}$$

where F is the total number of figure edgels, F_d is the number of figure edgels deleted in the final edgel map, N is total number of noise edgels, N_r is the number of noise edgels retained in the final edgel map, and γ and δ are penalty factors.

```
G = G_init; /* Start with an initial population of n edgel maps */

for (i = 0; i < n; i++) Ek[i]  = Ek_max;
/* Initialize kinetic energy demon of each member to a very high value */

repeat
 {
  repeat
   {
    Select two members P_1 and P_2 from the current generation G;

    Create two offspring C_1 and C_2 using the CGA crossover operator;

    Subject C_1 and C_2 to the CGA mutation operator;

    if neither offspring exists in the current generation
      C = stronger of C_1 and C_2;
    else if one offspring already exists in current generation
      C = offspring that does not exist in current generation;
    else C = null;

    if (C != null)
    {
      P = weakest member in the current population;
      if (E_C < E_P) then
      {
        replace P with C in the current population;
        Ek_C = Ek_P + (E_P - E_C);
      }
      else if ((E_C - E_P) < Ek_P)
      {
        replace P with C in the current population;
        Ek_C = Ek_P - (E_C - E_P);
      }
    }
  } until a certain predefined percentage of the weakest members in
    the current generation have been replaced;

for (i = 0; i < n; i++) Ek[i]  = A(Ek[i]);
/* Reduce the kinetic energy demon value of each member */

 } until the convergence criterion has been met;
```

Figure 6. Outline of the hybrid SSGA-MCA algorithm.

Table 1 tabulates the performance of the CGA on *Synthetic-1* (Fig. 7), *Synthetic-2* (Fig. 8) and *Synthetic-3* (Fig. 9), for both cases; where the edgels have been generated and encoded in the chromosome in a raster scan manner and where the edgels occupy random bit positions on the chromosome. Our experiments have shown that the CGA is sensitive to the value of α in that very low values of α drive the algorithm towards premature convergence to a suboptimal solution where most of the noise edgels persist in the resulting image. On the other hand, a high value of α removes a larger number of noise edgels but simultaneously removes some of the figure pixels as well. Recall that α is a threshold for the interaction coefficient $c_{i,j}$; values of $c_{i,j} > \alpha$ are deemed to denote strongly interacting (i.e., mutually reinforcing) edgels whereas values of $c_{i,j} < \alpha$ denote weakly interacting edgels. CGA_{ras} where the edgels are encoded in the chromosome in

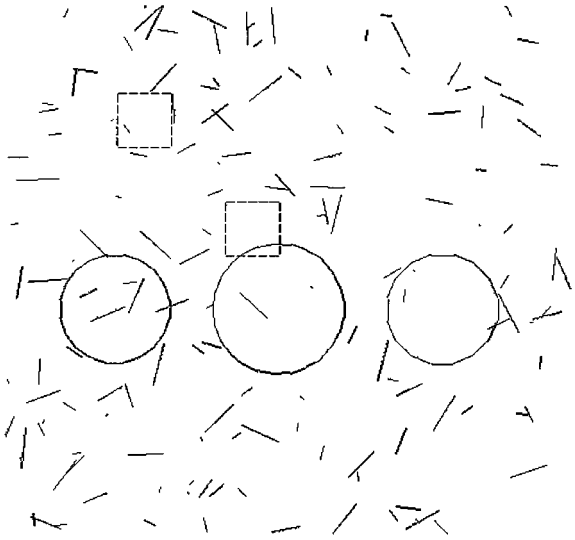

Figure 7. Synthetic edgel map *Synthetic-1*.

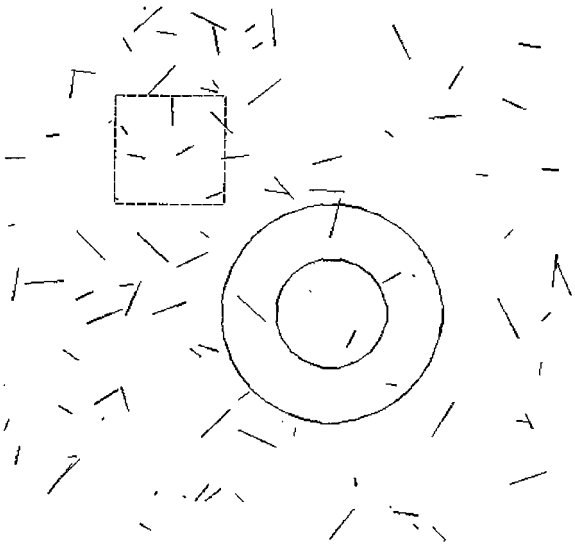

Figure 8. Synthetic edgel map *Synthetic-2*.

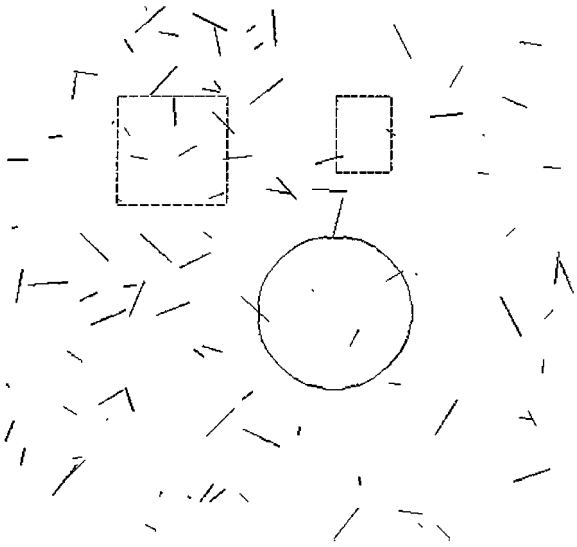

Figure 9. Synthetic edgel map *Synthetic-3*.

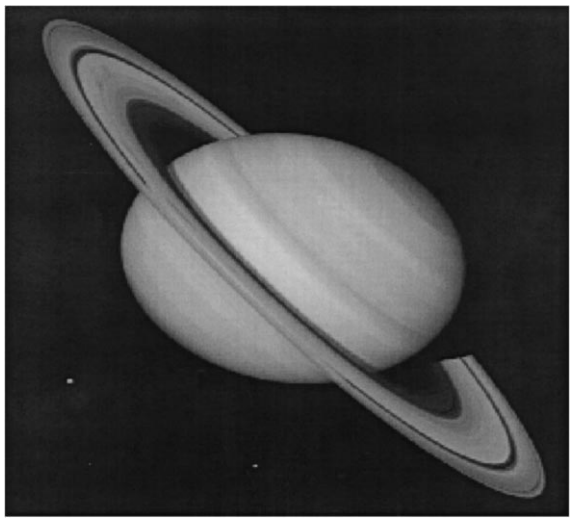

Figure 10. Gray-scale image *Saturn*.

raster scan order exhibits superior performance compared to CGA_{ran} in which the edgels are encoded in the chromosome in random order. The former exhibits a higher value of M over a large range of α values thereby implying a greater degree of noise removal while retaining a greater number of figure edgels. This goes to show that the CGA is sensitive to the chromosome encoding scheme employed.

Table 2 tabulates the performance of the SSGA on the three synthetic edgel maps *Synthetic-1* (Fig. 7), *Synthetic-2* (Fig. 8) and *Synthetic-3* (Fig. 9), for both raster scan and random encoding of edgels in the chromosome. The SSGA replaced 10% of its weakest chromosomes in each generation. The other parameters were chosen to be identical to the ones for the CGA. The SSGA showed greater insensitivity to the encoding technique used and to the value of α as far as the figure of merit of the final result was concerned. The encoding scheme however, did affect the performance of the SSGA in terms of the number of generations needed to arrive at the final result. Although the SSGA needed a greater number of generations to converge to a solution than did the CGA, the number of crossover and

Table 1. Performance of CGA on the synthetic edgel maps. CGA_{ras}: edgels encoded in chromosome in raster scan order, CGA_{ran}: edgels encoded in chromosome in random order, NG: no. of generations, T: execution time in ms, $\gamma = \delta = 1$.

		CGA_{ras}					CGA_{ran}				
Edgel map	α	F_d	N_r	M	NG	T	F_d	N_r	M	NG	T
Synthetic-1	$1/N$	0	111	0.59	151	1277	0	119	0.57	173	1465
$F = 96$, $N = 160$	$5/N$	0	53	0.75	156	1315	0	70	0.70	177	1498
	$10/N$	4	14	0.89	155	1307	8	28	0.79	174	1472
	$15/N$	8	12	0.86	159	1340	11	32	0.76	174	1473
	$20/N$	14	10	0.83	157	1325	13	38	0.73	178	1510
Synthetic-2	$1/N$	0	42	0.70	143	1211	0	57	0.63	158	1337
$F = 96$, $N = 96$	$5/N$	0	30	0.76	147	1245	0	41	0.70	161	1363
	$10/N$	0	22	0.81	143	1210	3	29	0.75	163	1379
	$15/N$	0	20	0.83	145	1225	7	25	0.75	162	1370
	$20/N$	11	10	0.82	144	1220	17	17	0.74	159	1346
Synthetic-3	$1/N$	0	93	0.56	138	1165	0	98	0.55	153	1295
$F = 100$, $N = 120$	$5/N$	0	62	0.66	138	1170	0	74	0.62	151	1275
	$10/N$	0	28	0.81	140	1184	3	37	0.75	155	1310
	$15/N$	5	24	0.80	141	1190	10	32	0.73	153	1293
	$20/N$	8	21	0.80	140	1185	13	25	0.75	152	1285

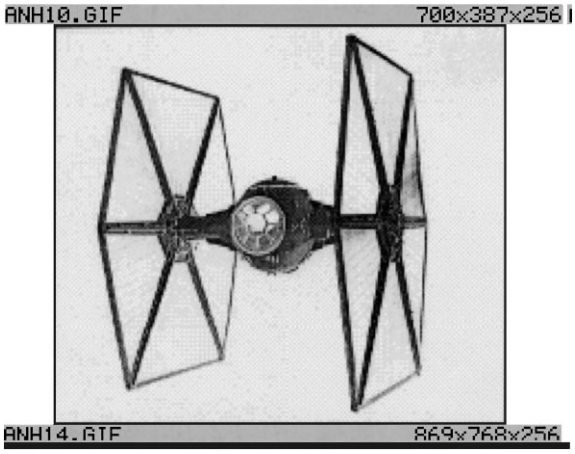

Figure 11. Gray-scale image *Spaceship*.

mutation operations per generation of the SSGA are a fraction of those needed per generation of the CGA resulting thereby in a lower overall execution time for the SSGA. Figures 14–16 show the result of applying the SSGA with random encoding of edgels in the chromosome on the three synthetic edgel maps *Synthetic-1*, *Synthetic-2* and *Synthetic-3* respectively. Figures 17 and 18 show the result of applying the SSGA to the edgel maps of gray-scale images *Saturn* and *Spaceship* respectively.

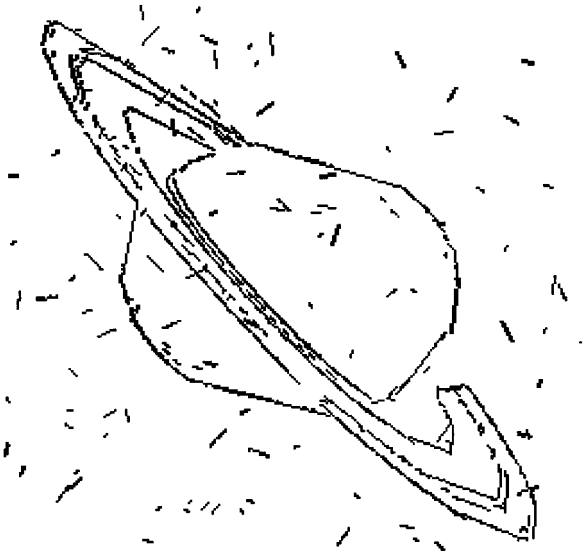

Figure 12. Edgel map of *Saturn.*

6.2. *Performance of the Hybrid Stochastic Optimization Algorithms*

The performance of the hybrid stochastic optimization algorithms i.e., CGA-SA, CGA-MCA, SSGA-SA and SSGA-MCA was evaluated on the synthetic edgel

Table 2. Performance of SSGA on the synthetic edgel maps. $SSGA_{ras}$: edgels encoded in chromosome in raster scan order, $SSGA_{ran}$: edgels encoded in chromosome in random order, NG: no. of generations, T: execution time in ms, $\gamma = \delta = 1$.

		$SSGA_{ras}$					$SSGA_{ran}$				
Edgel map	α	F_d	N_r	M	NG	T	F_d	N_r	M	NG	T
Synthetic-1	$1/N$	0	94	0.63	1235	1043	0	99	0.62	1422	1211
$F = 96$, $N = 160$	$5/N$	0	23	0.87	1242	1050	0	28	0.85	1437	1218
	$10/N$	3	18	0.87	1222	1035	4	20	0.87	1417	1197
	$15/N$	5	8	0.90	1225	1033	7	12	0.87	1420	1201
	$20/N$	7	8	0.89	1220	1032	9	12	0.86	1418	1198
Synthetic-2	$1/N$	0	38	0.72	1213	1020	0	42	0.70	1395	1180
$F = 96$, $N = 96$	$5/N$	0	15	0.86	1170	987	0	18	0.84	1332	1127
	$10/N$	3	8	0.90	1188	1005	3	10	0.88	1370	1156
	$15/N$	4	5	0.91	1173	990	5	6	0.90	1345	1138
	$20/N$	5	4	0.91	1180	998	7	5	0.89	1353	1145
Synthetic-3	$1/N$	0	39	0.75	999	845	0	45	0.73	1153	973
$F = 100$, $N = 120$	$5/N$	0	15	0.89	1015	860	0	19	0.86	1165	988
	$10/N$	0	12	0.91	1025	867	2	14	0.88	1180	997
	$15/N$	2	8	0.92	1005	851	3	10	0.90	1155	975
	$20/N$	4	5	0.92	1010	852	5	8	0.90	1160	980

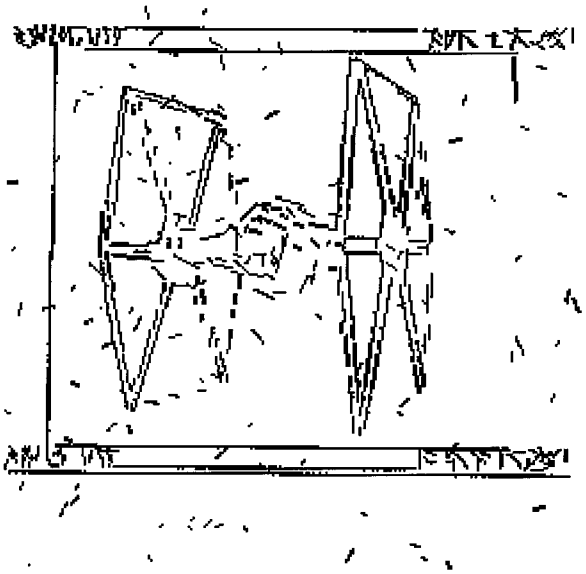

Figure 13. Edgel map of *Spaceship*.

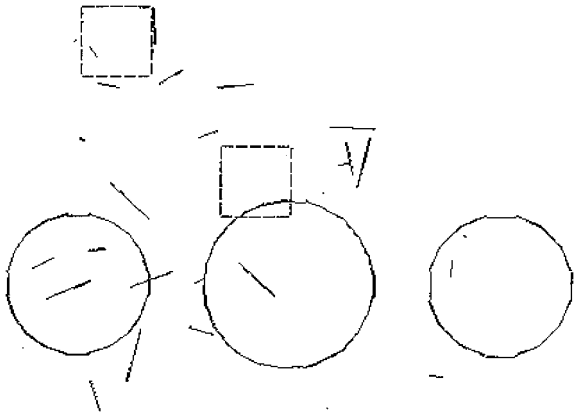

Figure 14. SSGA results on edgel map of *Synthetic-1*.

maps *Synthetic-1*, *Synthetic-2* and *Synthetic-3*. The results are summarized in Tables 3–6. The stopping criterion used in the case of each of the algorithms was identical and had an evolutionary flavor rather than an annealing-based one. The algorithm was considered to have converged if the best solution in the population had not changed in the past five consecutive generations. The annealing schedule for the temperature parameter T was chosen to be a geometric series of the form $T_{\text{next}} = \beta T_{\text{prev}}$ where $\beta = 0.95$ and T_{next} and T_{prev} are respectively the new and previous values for the temperature parameter at the end of each annealing step. The same annealing schedule was used for

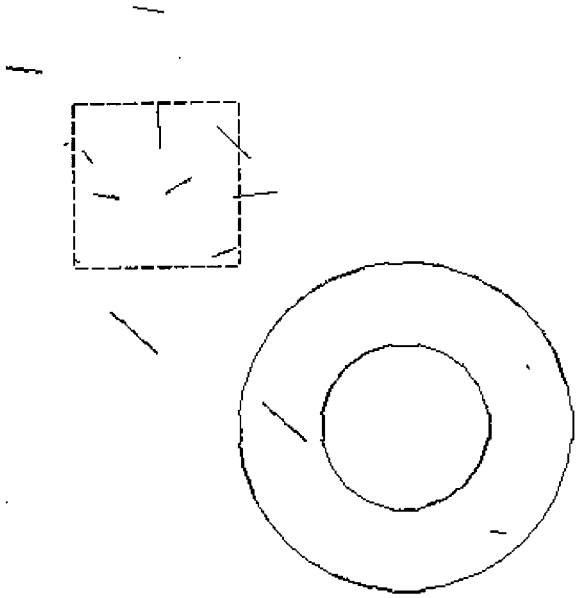

Figure 15. SSGA results on edgel map of *Synthetic-2*.

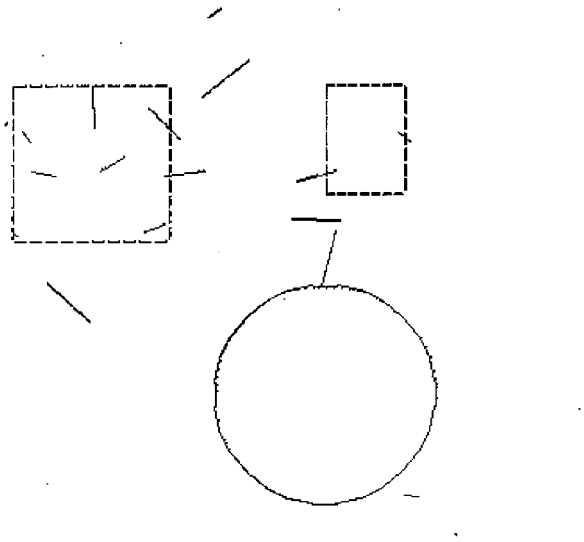

Figure 16. SSGA results on edgel map of *Synthetic-3*.

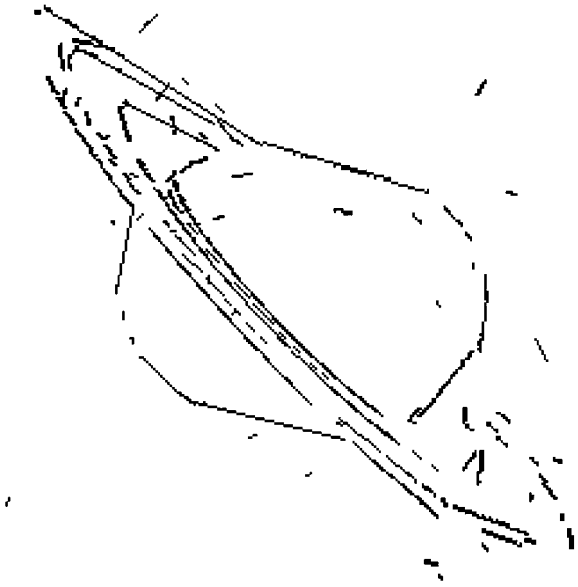

Figure 17. SSGA results on edgel map of *Saturn*.

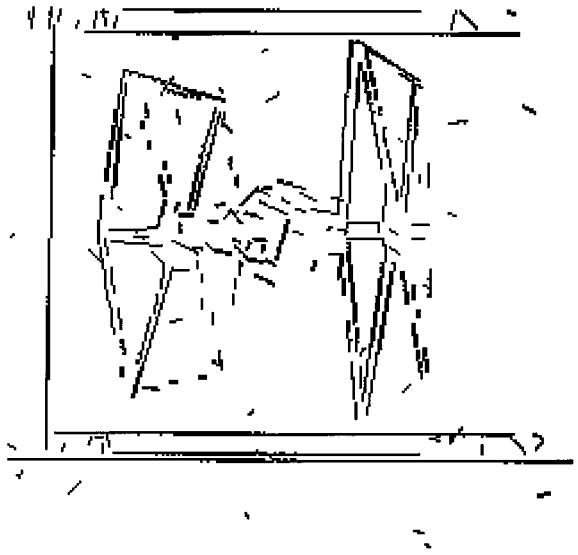

Figure 18. SSGA results on edgel map of *Spaceship*.

the kinetic energy demon parameter E_k. Although this geometric annealing schedule does not guarantee strict asymptotic convergence, as the logarithmic annealing schedule does, it has been known to give good results in practice [49].

Based on the results in Tables 3–6, the general observation is that the hybrid evolutionary stochastic optimization algorithms, by and large, performed better than their purely evolutionary counterparts. For comparison between the various algorithms, the following criteria were used:

- Figure of merit associated with the final edgel map,
- The number of generations needed for convergence,
- The execution time needed for convergence, and
- Sensitivity to the chromosome encoding scheme.

It was observed that the CGA-SA and CGA-MCA algorithms performed better than the CGA and that the

Table 3. Performance of CGA-SA on synthetic edgel maps. CGA-SA$_{ras}$: edgels encoded in chromosome in raster scan order, CGA-SA$_{ran}$: edgels encoded in chromosome in random order, NG: no. of generations, T: execution time in ms, $\gamma = \delta = 1$.

		CGA-SA$_{ras}$					CGA-SA$_{ran}$				
Edgel map	α	F_d	N_r	M	NG	T	F_d	N_r	M	NG	T
Synthetic-1	$1/N$	0	72	0.69	103	1046	0	75	0.68	115	1170
$F = 96$, $N = 160$	$5/N$	0	34	0.82	108	1095	0	37	0.81	119	1205
	$10/N$	0	10	0.94	110	1115	2	13	0.91	116	1179
	$15/N$	5	8	0.91	106	1077	5	18	0.86	121	1226
	$20/N$	12	9	0.85	111	1130	13	27	0.77	115	1172
Synthetic-2	$1/N$	0	27	0.78	98	993	0	34	0.74	107	1086
$F = 96$, $N = 96$	$5/N$	0	18	0.84	95	965	0	23	0.81	102	1035
	$10/N$	0	11	0.90	98	995	3	12	0.86	103	1047
	$15/N$	4	12	0.86	97	987	5	13	0.84	105	1070
	$20/N$	8	10	0.84	99	1010	10	11	0.82	109	1110
Synthetic-3	$1/N$	0	37	0.76	105	1071	0	41	0.75	108	1095
$F = 100$, $N = 120$	$5/N$	0	23	0.84	102	1040	0	32	0.79	105	1075
	$10/N$	0	13	0.88	109	1115	3	15	0.87	113	1157
	$15/N$	2	10	0.91	110	1125	5	14	0.86	114	1165
	$20/N$	5	9	0.89	108	1103	7	13	0.85	111	1136

Table 4. Performance of CGA-MCA on synthetic edgel maps. CGA-MCA$_{ras}$: edgels encoded in chromosome in raster scan order, CGA-MCA$_{ran}$: edgels encoded in chromosome in random order, NG: no. of generations, T: execution time in ms, $\gamma = \delta = 1$.

		CGA-MCA$_{ras}$					CGA-MCA$_{ran}$				
Edgel map	α	F_d	N_r	M	NG	T	F_d	N_r	M	NG	T
Synthetic-1	$1/N$	0	70	0.70	105	905	0	73	0.69	114	995
$F = 96$, $N = 160$	$5/N$	0	32	0.83	106	923	0	34	0.82	118	1029
	$10/N$	0	11	0.94	109	950	2	12	0.91	114	992
	$15/N$	4	8	0.92	105	906	5	15	0.87	116	1009
	$20/N$	10	9	0.86	110	958	12	25	0.78	115	1002
Synthetic-2	$1/N$	0	25	0.79	97	845	0	32	0.75	105	917
$F = 96$, $N = 96$	$5/N$	0	17	0.85	93	810	0	22	0.81	101	882
	$10/N$	0	10	0.91	97	845	3	11	0.87	105	915
	$15/N$	3	11	0.87	95	830	5	12	0.85	106	923
	$20/N$	8	8	0.86	98	856	9	10	0.83	108	942
Synthetic-3	$1/N$	0	32	0.79	103	897	0	38	0.76	105	915
$F = 100$, $N = 120$	$5/N$	0	22	0.85	103	900	0	30	0.80	107	930
	$10/N$	0	10	0.92	108	841	3	13	0.88	112	975
	$15/N$	2	9	0.91	111	966	5	12	0.87	114	991
	$20/N$	4	9	0.90	107	932	7	11	0.86	110	960

Table 5. Performance of SSGA-SA on synthetic edgel maps. SSGA-SA$_{ras}$: edgels encoded in chromosome in raster scan order, SSGA-SA$_{ran}$: edgels encoded in chromosome in random order, NG: no. of generations, T: execution time in ms, $\gamma = \delta = 1$.

		SSGA-SA$_{ras}$					SSGA-SA$_{ran}$				
Edgel map	α	F_d	N_r	M	NG	T	F_d	N_r	M	NG	T
Synthetic-1	$1/N$	0	54	0.75	1032	1003	0	58	0.73	1077	1041
$F = 96$, $N = 160$	$5/N$	0	19	0.89	1021	984	0	23	0.87	1043	1005
	$10/N$	3	14	0.89	1025	990	4	16	0.88	1031	1001
	$15/N$	5	9	0.90	1025	986	6	11	0.87	1063	1035
	$20/N$	7	7	0.90	1020	981	8	10	0.87	1087	1046
Synthetic-2	$1/N$	0	32	0.75	1019	981	0	37	0.72	1095	1053
$F = 96$, $N = 96$	$5/N$	0	12	0.89	1017	978	0	16	0.86	1062	1020
	$10/N$	2	7	0.91	1018	980	3	9	0.89	1070	1027
	$15/N$	3	4	0.93	1013	972	5	6	0.90	1045	1003
	$20/N$	4	4	0.92	1018	979	6	5	0.90	1053	1011
Synthetic-3	$1/N$	0	21	0.85	806	775	0	25	0.83	852	815
$F = 100$, $N = 120$	$5/N$	0	12	0.91	811	778	0	17	0.88	870	835
	$10/N$	0	9	0.93	813	782	2	11	0.90	872	837
	$15/N$	2	7	0.93	821	795	4	8	0.90	855	827
	$20/N$	4	5	0.92	818	787	5	7	0.90	863	835

Table 6. Performance of SSGA-MCA on synthetic edgel maps. SSGA-MCA$_{ras}$: edgels encoded in chromosome in raster scan order, SSGA-MCA$_{ran}$: edgels encoded in chromosome in random order, NG: no. of generations, T: execution time in ms, $\gamma = \delta = 1$.

		SSGA-MCA$_{ras}$					SSGA-MCA$_{ran}$				
Edgel map	α	F_d	N_r	M	NG	T	F_d	N_r	M	NG	T
Synthetic-1	$1/N$	0	55	0.74	1035	910	0	58	0.73	1075	945
$F = 96$, $N = 160$	$5/N$	0	20	0.89	1019	898	0	22	0.88	1047	923
	$10/N$	4	15	0.88	1020	900	6	16	0.86	1035	920
	$15/N$	4	10	0.91	1027	903	6	12	0.88	1058	933
	$20/N$	7	8	0.89	1018	895	8	10	0.87	1075	956
Synthetic-2	$1/N$	0	31	0.76	1022	901	0	36	0.73	1090	960
$F = 96$, $N = 96$	$5/N$	0	14	0.87	1015	893	0	17	0.85	1055	928
	$10/N$	3	7	0.91	1015	894	5	8	0.88	1075	945
	$15/N$	4	4	0.92	1005	886	5	5	0.91	1044	919
	$20/N$	5	4	0.91	1012	892	6	5	0.90	1051	927
Synthetic-3	$1/N$	0	23	0.84	809	713	0	27	0.82	843	742
$F = 100$, $N = 120$	$5/N$	0	13	0.90	807	710	0	19	0.86	852	750
	$10/N$	0	8	0.94	812	714	2	10	0.91	863	761
	$15/N$	3	7	0.92	817	720	4	9	0.90	851	747
	$20/N$	5	4	0.92	812	717	5	7	0.90	854	753

SSGA-SA and SSGA-MCA algorithms performed better than the SSGA. When comparing the CGA and the CGA-SA algorithm, it was found that the CGA-SA algorithm needed 30–35% fewer generations for convergence. However, due to the overhead of having to compute the Boltzmann function during population replacement (Fig. 3), the decrease in overall execution time was only 15–18% compared to the CGA. The CGA-SA algorithm, however, did exhibit greater insensitivity to the value of α and to the chromosome encoding scheme used and also converged to a solution with a higher figure of merit. The CGA-MCA algorithm, also needed 30–35% fewer generations for convergence but the overall decrease in computation time was approximately 25–30% compared to the CGA. Although, the CGA-MCA algorithm does entail updating the demon energy during population replacement (Fig. 5), this can be accomplished with a simple addition/subtraction operation as opposed to having to compute the exponential term in the Boltzmann function (Eq. (16)) as in the case of the CGA-SA algorithm. In terms of the figure of merit of the final solution, and the insensitivity to the value of α and to the chromosome encoding scheme used, the performance of the CGA-MCA algorithm was comparable to that of the CGA-SA algorithm. The principal advantage of the CGA-MCA algorithm over the CGA-SA algorithm is the shorter execution time of the former.

When comparing the SSGA and the SSGA-SA algorithm, it was noticed that the SSGA-SA algorithm required approximately 15–20% fewer generations for convergence as compared to the SSGA. However due to the overhead of having to compute the Boltzmann function, the overall reduction in execution time of the SSGA-SA algorithm was only 5–8% compared to the SSGA. When comparing the SSGA-MCA algorithm with the SSGA, the SSGA-MCA algorithm was seen to require 15–20% fewer generations for convergence and exhibit a 12–15% reduction in execution time compared to the SSGA. Both, the SSGA-MCA algorithm and the SSGA-SA algorithm showed an improvement over the SSGA algorithm in terms of the execution time, figure of merit of the final result, and the insensitivity to the value of α and to the chromosome encoding scheme used. The performance of the SSGA-MCA algorithm was comparable to that of the SSGA-SA algorithm in terms of the figure of merit of the final solution and the insensitivity to the value of α and to the chromosome encoding scheme used. The SSGA-MCA algorithm however showed an overall lower execution time compared to the SSGA-SA algorithm. This could be again attributed to the fact that updating demon energy in the case of the former is computationally far less expensive compared to the computation of the Boltzmann function in the case of the latter.

For the sake of completeness, the hybrid evolutionary stochastic optimization algorithms were compared with the classical SA and MCA algorithms (Tables 7 and 8). The annealing parameters used in the SA and MCA algorithms were identical to those of their hybrid evolutionary counterparts. In the case of MCA, the kinetic energy was represented by an ensemble of demons, one demon for each edgel pair. At the end of each annealing step, the demons were shuffled and randomly reassigned to the edgel pairs as suggested in [15]. The SA and MCA algorithms were observed to converge faster than their hybrid evolutionary counterparts but the solutions that they converged to had a lower figure of merit. The slower speed of the hybrid evolutionary stochastic optimization algorithms (i.e., CGA-SA, CGA-MCA, SSGA-SA and SSGA-MCA) could be attributed to the fact that, unlike SA or MCA, they incur the overhead of having to maintain and update a population of solution states. The fact that the hybrid evolutionary stochastic optimization algorithms converged to better solutions can be attributed to their ability to use good partial solutions as building blocks in the process of constructing globally optimal solutions. The quality of solutions obtained by SA and MCA was comparable. As expected, both SA and MCA showed insensitivity to the chromosome encoding scheme employed, with the MCA exhibiting a faster rate of convergence than SA.

In an overall comparison of all the evolutionary stochastic optimization algorithms for figure-ground separation that were examined in this paper, the SSGA-MCA performed the best in terms of execution time, number of generations needed for convergence, the quality of the final solution (in terms of a predefined figure of merit) and the robustness of the algorithm to change in the threshold parameter α. The SSGA-MCA was followed by the SSGA-SA algorithm which matched the performance of the former in terms of all the aforementioned criteria except for an degradation in the execution time for reasons already mentioned. Figures 19 and 20 show the results of the SSGA-MCA algorithm on the edgel maps of the gray-scale images *Saturn* and *Spaceship* respectively. Figures 21 and 22 show the results of the SSGA-SA algorithm on the

Table 7. Performance of SA on synthetic edgel maps. SA_{ras}: edgels encoded in chromosome in raster scan order, SA_{ran}: edgels encoded in chromosome in random order, N: no. of iterations, T: execution time in ms, $\gamma = \delta = 1$.

		SA_{ras}					SA_{ran}				
Edgel map	α	F_d	N_r	M	N	T	F_d	N_r	M	N	T
Synthetic-1	$1/N$	0	85	0.65	2173	981	0	88	0.65	2180	984
$F = 96$, $N = 160$	$5/N$	0	42	0.79	2145	968	0	43	0.79	2151	971
	$10/N$	3	26	0.84	2148	969	4	27	0.83	2155	972
	$15/N$	8	12	0.86	2149	969	8	14	0.85	2154	971
	$20/N$	17	10	0.81	2153	970	18	11	0.80	2160	973
Synthetic-2	$1/N$	0	37	0.72	2105	950	0	39	0.71	2110	952
$F = 96$, $N = 96$	$5/N$	1	24	0.79	2110	952	1	25	0.79	2114	954
	$10/N$	3	16	0.83	2112	953	4	15	0.83	2115	955
	$15/N$	4	14	0.84	2110	952	5	15	0.83	2112	953
	$20/N$	8	12	0.83	2107	951	9	12	0.82	2110	952
Synthetic-3	$1/N$	0	47	0.72	1670	754	0	49	0.71	1676	756
$F = 100$, $N = 120$	$5/N$	1	34	0.77	1681	759	1	35	0.77	1685	761
	$10/N$	2	18	0.85	1680	759	2	20	0.84	1686	761
	$15/N$	4	12	0.88	1678	758	4	14	0.86	1684	760
	$20/N$	7	10	0.87	1675	756	7	11	0.86	1679	753

Table 8. Performance of MCA on synthetic edgel maps. MCA_{ras}: edgels encoded in chromosome in raster scan order, MCA_{ran}: edgels encoded in chromosome in random order, N: no. of iterations, T: execution time in ms, $\gamma = \delta = 1$.

		MCA_{ras}					MCA_{ran}				
Edgel map	α	F_d	N_r	M	N	T	F_d	N_r	M	N	T
Synthetic-1	$1/N$	0	82	0.66	2317	870	0	84	0.66	2328	874
$F = 96$, $N = 160$	$5/N$	0	36	0.82	2335	877	0	37	0.81	2338	879
	$10/N$	4	24	0.84	2300	863	4	25	0.83	2310	867
	$15/N$	7	12	0.87	2305	865	7	13	0.87	2310	867
	$20/N$	15	11	0.82	2301	863	16	11	0.81	2305	865
Synthetic-2	$1/N$	0	36	0.73	2190	823	0	35	0.73	2057	773
$F = 96$, $N = 96$	$5/N$	1	22	0.81	2180	819	2	21	0.81	2183	820
	$10/N$	4	14	0.84	2185	821	5	15	0.83	2188	822
	$15/N$	6	12	0.84	2188	822	7	12	0.83	2191	823
	$20/N$	9	11	0.83	2173	816	9	12	0.82	2178	818
Synthetic-3	$1/N$	0	43	0.74	1825	685	0	45	0.73	1827	686
$F = 100$, $N = 120$	$5/N$	0	33	0.78	1821	683	1	34	0.78	1824	684
	$10/N$	2	17	0.86	1818	682	2	19	0.85	1820	683
	$15/N$	4	13	0.87	1817	682	4	14	0.86	1820	683
	$20/N$	8	11	0.85	1814	681	8	12	0.85	1818	682

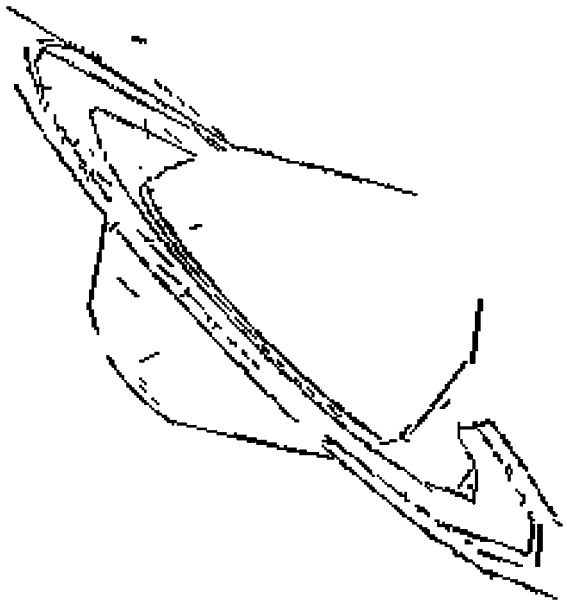

Figure 19. SSGA-MCA algorithm results on the edgel map of *Saturn*.

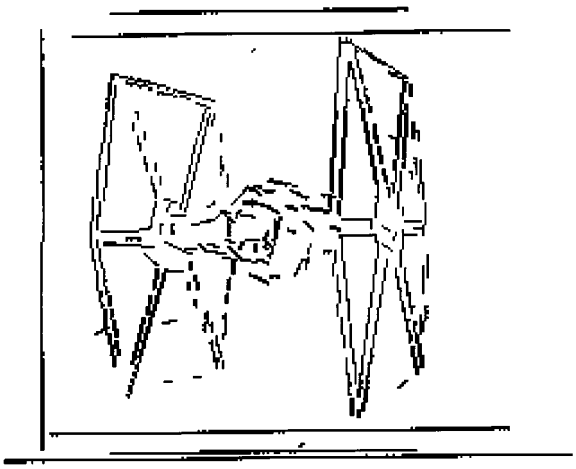

Figure 20. SSGA-MCA algorithm results on the edgel map of *Spaceship*.

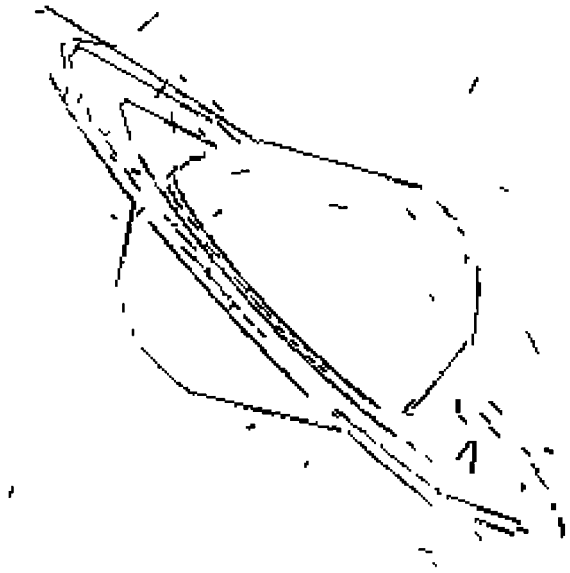

Figure 21. SSGA-SA algorithm results on the edgel map of *Saturn*.

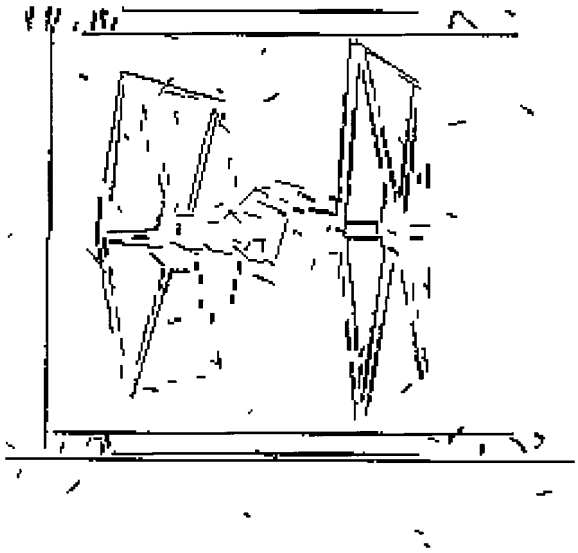

Figure 22. SSGA-SA algorithm results on the edgel map of *Spaceship*.

edgel maps of the gray-scale images *Saturn* and *Spaceship* respectively. Figures 23 and 24 compare the convergence rates of the SSGA, SSGA-SA and SSGA-MCA on the *Saturn* and *Spaceship* images respectively. As can be seen, the incorporation of stochastic hill-climbing greatly improves the convergence rate of the SSGA and also enables it to converge to a better solution (i.e., one with a lower energy value).

7. Conclusions and Future Directions

In this paper, the problem of figure-ground separation was tackled from the viewpoint of combinatorial optimization. Previous attempts have used deterministic optimization techniques based on relaxation and gradient descent-based search, and stochastic optimization techniques based on simulated annealing (SA)

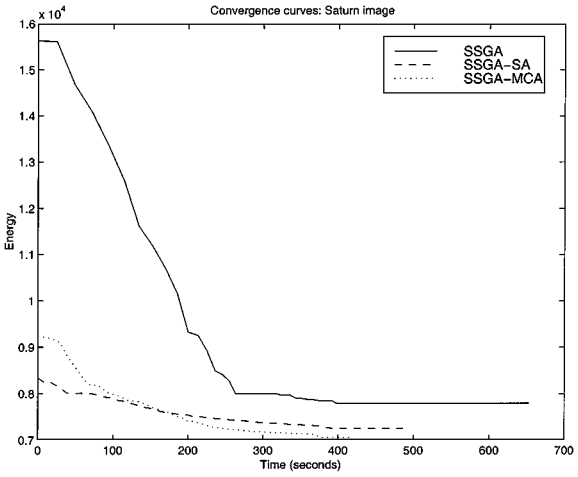

Figure 23. Convergence curves for the *Saturn* image.

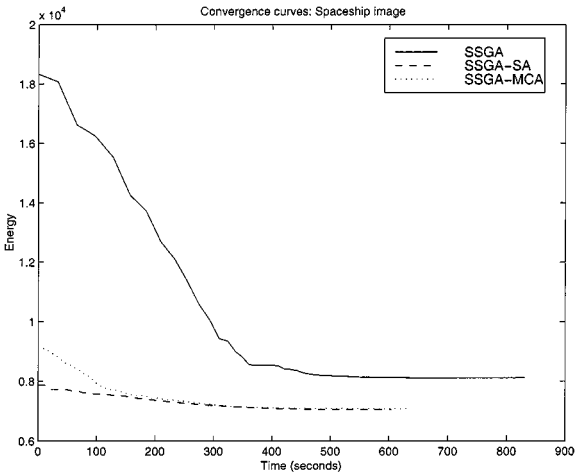

Figure 24. Convergence curves for the *Spaceship* image.

and microcanonical annealing (MCA). This paper explored the use of evolutionary algorithms, in particular, the canonical genetic algorithm (CGA) and the steady-state genetic algorithm (SSGA), in the context of figure-ground separation.

A mathematical model encoding the figure-ground separation problem that makes explicit the definition of shape in terms of attributes such as cocircularity, smoothness, proximity and contrast was described. The model was based on the formulation of an energy function that incorporates pairwise interactions between local image features in the form of edgels and was shown to be isomorphic to the interacting spin (Ising) system from quantum physics. The desired edgel map was deemed to be the one that corresponded to a global minimum of the energy function. Determining a global minimum via exhaustive search would have resulted in an algorithm with an exponential run-time complexity whereas a deterministic optimization technique based on local search would have been trapped in one of the several local minima in the energy function landscape thereby forgoing a desired global minimum. A stochastic optimization technique that could avoid local minima and asymptotically converge to a global minimum with unit probability was clearly called for.

In spite of the computational advantage derived from the *Schema Theorem* and their inherent parallelism, evolutionary algorithms like the CGA and the SSGA were seen to suffer from certain inherent drawbacks in the context of the figure-ground separation problem. The CGA and SSGA showed a great degree of sensitivity to the value of the threshold α as well as to the manner in which the edgel map was encoded in the form of a chromosome. The absence of a hill climbing mechanism resulted in a large number of generations (and a correspondingly high execution time) for the convergence of the CGA and SSGA. The incorporation of a deterministic hill-climbing mechanism drove the CGA and SSGA towards premature convergence to a local minimum, i.e., a suboptimal solution. With a view towards alleviating these shortcomings, the paper considered the use of hybrid evolutionary stochastic optimization algorithms that combined the stochastic hill-climbing and asymptotic convergence properties of stochastic annealing algorithms with the *building blocks* property of evolutionary algorithms.

The paper examined four hybrid evolutionary stochastic optimization algorithms in the context of the figure-ground separation problem:

- CGA-SA, a combination of the CGA and SA originally proposed by Mahfoud and Goldberg [46],
- CGA-MCA, a combination of the CGA and MCA proposed in this paper,
- SSGA-SA, a combination of the SSGA and SA proposed in this paper, and
- SSGA-MCA, a combination of the SSGA and MCA proposed in this paper.

The hybrid evolutionary stochastic optimization algorithms were found to outperform their purely evolutionary counterparts when compared using synthetic edgel maps. The SSGA-MCA and SSGA-SA algorithms were shown to perform the best and second best respectively. Results of the CGA, SSGA, SSGA-MCA and SSGA-SA on edgel maps derived from gray-scale

images underscore the advantages of the hybrid evolutionary stochastic optimization algorithms. When compared to the classical SA and MCA algorithms, the hybrid evolutionary algorithms were observed to be slower due to the overhead of having to maintain and update a population of solution states. But the hybrid evolutionary algorithms were seen to converge to better solutions (i.e., with lower energy or higher figure of merit) compared to the solutions obtained using the classical SA or MCA algorithms.

The intent of this paper was to show how the incorporation of classical stochastic hill-climbing techniques such as those employed by classical SA and MCA could improve the performance of classical GA-based algorithms such as the CGA and SSGA. It must be noted however, that we have not used the most recent SA or MCA algorithms such as those based on Adaptive SA (ASA) and fat-tailed SA with Cauchy annealing. Also, we have not used the most recent GA-based algorithms such as those based on messy GAs. Hybrid evolutionary stochastic optimization algorithms that combine the modern developments in SA, MCA and the GA were beyond the scope of this paper but will be pursued in our future research on hybrid evolutionary stochastic optimization algorithms. Future research will consider issues related to the parallelization of the aforementioned hybrid evolutionary stochastic optimization algorithms. Neural network structures that are capable of hybrid evolutionary optimization will be investigated. Other problems in computer vision such as shape from stereo, motion analysis and image segmentation will also be investigated.

Acknowledgments

The support of the University of Georgia Research Foundation Inc., Athens, Georgia in the form of a Faculty Research Grant to Dr. Suchendra M. Bhandarkar is gratefully acknowledged. The authors wish to thank the anonymous reviewers for their insightful comments and constructive suggestions which greatly improved the paper.

References

1. L. Herault and R. Horaud, "Figure-ground discrimination: A combinatorial optimization approach," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 15, no. 9, pp. 899–914, 1993.
2. S. Kirkpatrick, C. Gelatt, Jr., and M. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.
3. M. Creutz, "Microcanonical Monte Carlo Simulation," *Physics Review Letters*, vol. 50, no. 19, pp. 1411–1414, 1983.
4. P. Banerjee, M.H. Jones, and J.S. Sargent, "Parallel simulated annealing algorithms for cell placement on the hypercube multiprocessor," *IEEE Trans. Parallel and Distributed Systems*, vol. 1, pp. 91–106, 1990.
5. A. Casotto, F. Romeo, and A. Sangiovanni-Vincentelli, "A parallel simulated annealing algorithm for the placement of macro cells," *IEEE Trans. Computer-Aided Design*, pp. 838–847, 1987.
6. R. Jayaraman and R. Rutenbar, "Floor planning by annealing on a hypercube multiprocessor," in *Proc. IEEE Intl. Conf. Computer Aided Design*, Nov. 1987, pp. 346–349.
7. S. Nahar, S. Sahni, and E. Shragowitz, "Simulated annealing and combinatorial optimization," *International Journal of Computer Aided VLSI Design*, vol. 1, pp. 1–23, 1989.
8. C.P. Wong and R.D. Fiebrich, "Simulated annealing-based circuit placement on the connection machine system," in *Proc. Intl. Conf. Computer Design*, Oct. 1987, pp. 78–82.
9. E. Bonomy and J.L. Lutton, "N-city traveling salesman problem and metropolis algorithm," *SIAM Rev.*, vol. 26, no. 84, pp. 551–568, 1984.
10. E. Felten, S. Karlin, and S.W. Otto, "The traveling salesman problem on a hypercubic MIMD computer," in *Proc. IEEE International Conference Parallel Processing*, 1985, pp. 6–10.
11. S.M. Bhandarkar and S. Chirravuri, "A study of massively parallel simulated annealing algorithms for chromosome reconstruction via clone ordering," *Parallel Algorithms and Applications*, vol. 9, pp. 67–89, 1996.
12. A.J. Cuticchia, J. Arnold, and W.E. Timberlake, "The use of simulated annealing in chromosome reconstruction experiments based on binary scoring," *Genetics*, vol. 132, pp. 591–601, 1992.
13. S. Lin and B. Kernighan, "An effective heuristic for the traveling salesman problem," *Operations Research*, vol. 21, pp. 498–516, 1973.
14. E.H.L. Aarts and K. Korst, *Simulated Annealing and Boltzman Machines: A Stochastic Approach to Combinatorial Optimization and Neural Computing*, Wiley: New York, 1989.
15. S.T. Barnard, "Stochastic matching of stereo over scale," *Intl. Jour. Computer Vision*, vol. 3, no. 1, pp. 17–32, 1989.
16. S. Geman and D. Geman, "Stochastic relaxation, Gibbs distribution and the Bayesian restoration of images," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 6, pp. 721–741, 1984.
17. H. Orland, "Mean field theory for optimization problems," *Jour. Physics Letters*, vol. 46, no. 17, pp. L763–L770, 1985.
18. L. Herault and J.J. Niez, "Neural networks and graph K-partioning," *Complex Systems*, vol. 3, no. 6, pp. 531–576, 1989.
19. C. Peterson and J.R. Anderson, "A mean field learning algorithm for neural networks," *Complex Systems*, vol. 1, pp. 995–1019, 1987.
20. C. Peterson, "A new method for mapping optimization problems onto neural networks," *Intl. Jour. Neural Syst.*, vol. 1, no. 1, pp. 3–22, 1989.
21. D. Geiger and A. Yuille, "A common framework for image segmentation," *Intl. Jour. Computer Vision*, vol. 6, no. 3, pp. 227–243, 1991.
22. D. Geiger and F. Girosi, "Parallel and deterministic algorithms from MRF's: Surface reconstruction," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 13, no. 5, pp. 410–412, 1991.

23. J. Zerubia and R. Chellappa, "Mean field approximation using compound Gauss–Markov random field for edge detection and image restoration," in *Proc. IEEE Intl. Conf. Acoustics Speech Signal Processing*, Alburquerque, NM, April 1990, pp. 2193–2196.
24. W. Kohler, *Gestalt Psychology*, Meridian Press: New York, NY, 1980.
25. P. Parent and S.W. Zucker, "Trace inference, curvature consistency and curve detection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 11, no. 8, pp. 823–839, 1989.
26. A. Sha'ashua and S. Ullman, "Structural saliency: The detection of globally salient features using a locally connected network," in *Proc. IEEE Intl. Conf. Computer Vision*, Tampa, FL, Dec. 1988, pp. 321–327.
27. D. Gutfinger and J. Sklansky, "Robust classifiers by mixed adaptation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 13, no. 6, pp. 552–567, 1991.
28. T.J. Sejnowski and G.E. Hinton, "Separating figure from ground with a Boltzmann machine," in *Vision, Brain, and Cooperative Computation*, edited by M. Arbib and A. Hanson, MIT Press: Cambridge, MA, pp. 703–724, 1988.
29. P. Carnevali, L. Coletti, and S. Patarnello, "Image processing by simulated annealing," *IBM Jour. Res. and Dev.*, vol. 29, no. 6, pp. 569–579, 1985.
30. C. Peterson, "Track finding with neural networks," *Nuclear Instrum. Methods Physics Res.*, vol. A279, pp. 537–545, 1989.
31. A. Blake and A. Zisserman, *Visual Reconstruction*, MIT Press: Cambridge, MA, 1987.
32. A. Blake, "Comparison of the efficiency of deterministic and stochastic algorithms for visual reconstruction," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 11, no. 1, pp. 2–12, 1989.
33. H.L. Tan, S.B. Gelfand, and E.J. Delp, "A comparative cost function approach to edge detection," *IEEE Trans. Systems, Man, and Cybernetics*, vol. 19, no. 6, pp. 1337–1349, 1989.
34. H.L. Tan, S.B. Gelfand, and E.J. Delp, "A cost minimization approach to edge detection using simulated annealing," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 14, no. 1, pp. 3–18, 1991.
35. S.M. Bhandarkar, Y. Zhang, and W.D. Potter, "Edge detection using genetic algorithm-based optimization," *Pattern Recognition*, vol. 27, no. 9, pp. 1159–1180, 1994.
36. S.T. Acton and A.C. Bovik, "Anisotropic edge detection using mean field annealing," in *Proc. IEEE Intl. Conf. Acoustics, Speech and Signal Processing*, 1992, vol. II, pp. 393–396.
37. G. Roth and M.D. Levine, "Geometric primitive extraction using a genetic algorithm," in *Proc. IEEE Intl. Conf. Comp. Vis. Patt. Recog.*, 1992, pp. 640–643.
38. D.B. Fogel, "An introduction to simulated evolutionary computation," *IEEE Trans. Neural Networks*, vol. 5, no. 1, pp. 3–14, 1994.
39. D. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley Pub. Co.: Reading, MA, 1988.
40. J.H. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press: Ann Arbor, MI, 1975.
41. M. Mitchell, *An Introduction to Genetic Algorithms*, MIT Press: Cambridge, MA, 1996.
42. L. Davis, *Handbook of Genetic Algorithms*, Van Nostrand Reinhold: New York, NY, 1991.
43. N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *Jour. Chemical Physics*, vol. 21, pp. 1087–1092, 1953.
44. G. Bhanot, M. Creutz, and H. Neuberger, "Microcanonical simulation of ising systems," *Nuclear Physics*, vol. B235, no. FS11, pp. 417–434, 1984.
45. D. Adler, "Genetic algorithms and simulated annealing: A marriage proposal," in *Proc. IEEE Intl. Conf. Neural Networks*, San Francisco, CA, 1993.
46. S.W. Mahfoud and D.E. Goldberg, "Parallel recombinative simulated annealing: A genetic algorithm," *Parallel Computing*, vol. 21, no. 1, pp. 1–28, 1995.
47. J.F. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679–698, 1986.
48. V.S. Nalwa and T.O. Binford, "On detecting edges," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 699–714, 1986.
49. F. Romeo and A. Sangiovanni–Vincentelli, "A theoretical framework for simulated annealing," *Algorithmica*, vol. 6, pp. 302–345, 1991.
50. M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *Intl. Jour. Computer Vision*, vol. 1, no. 4, pp. 321–331, 1988.

Suchendra M. Bhandarkar received a B.Tech in Electrical Engineering from the Indian Institute of Technology, Bombay, India in 1983, and M.S. and Ph.D. in Computer Engineering from Syracuse University, Syracuse, New York in 1985 and 1989 respectively. He is currently an Associate Professor and Director of the Visual and Parallel Computing Laboratory (VPCL) in the Department of Computer Science at the University of Georgia, Athens, Georgia. He was a Syracuse University Fellow for the academic years 1986–1987 and 1987–1988. He is a member of the IEEE, AAAI, ACM and SPIE, and the honor societies Phi Kappa Phi and Phi Beta Delta. He is a coauthor of the book *3D Object Recognition from Range Images* (Springer-Verlag, 1992). His research interests include computer vision, pattern recognition, image processing, artificial intelligence and parallel algorithms and architectures for computer vision and pattern recognition. He has over 70 published research articles in these areas including over 30 articles in refereed archival journals.

Xia Zeng received a B.S. in Information Theory from the Department of Mathematics, Beijing University, P.R. China in 1985 and an M.S. in Computer Science from the University of Georgia, Athens, Georgia in 1995. He is currently employed as a senior software engineer at Talus Inc. in Atlanta, Georgia.