# A SIMULATED ANNEALING GENETIC ALGORITHM FOR BLIND DECONVOLUTION OF NONLINEARLY DEGRADED IMAGES

*Kaaren May, Tania Stathaki, and Anthony Constantinides*

Signal Processing and Digital Systems Section
Imperial College of Science, Technology and Medicine, London SW7 2BT, U. K.
e-mail: k.may@ic.ac.uk

## ABSTRACT

A Simulated Annealing Genetic Algorithm (SAGA) is presented for blind restoration of a nonlinearly-degraded image with additive Gaussian noise. The degradation is modelled by a quadratic Volterra filter. Deconvolution of the original image and the unknown Volterra filter is formulated as a constrained optimisation problem, the cost function of which is minimised by the SAGA.

## 1. INTRODUCTION

Image degradation involves both linear and nonlinear processes, which include relative motion between the object and camera, wrong focus, nonlinearity of the electro-optical sensor, and defects of optical lenses. A general model for image degradation is

$$g(n_1, n_2) = T\left(\sum_{i_1,i_2} d(n_1, n_2, i_1, i_2) f(n_1, n_2)\right) \odot v(n_1, n_2), \tag{1}$$

where $T(\cdot)$ defines a memoryless nonlinearity, $d(n_1, n_2, i_1, i_2)$ is the 2D impulse response of the blurring system at pixel $(n_1, n_2)$, $\odot$ represents a pointwise operation, and $v(n_1, n_2)$ is the corruptive noise process [1]. It is usually assumed that the blurring function is space-invariant and the noise $v(n_1, n_2)$ is an additive white Gaussian process. In general, the function $T$ is ignored. However, in some applications it is necessary to take the sensor nonlinearity into account. Tekalp and Pavlović [1, Chapter 8] conclude that if the logarithmic relationship between the optical density and the incident exposure of scanned photographic images is ignored, then this can result in unacceptable restorations. In this paper, a Volterra filter with an additive noise term is used as a general model for image degradation.

The Volterra filter is suitable for modelling equation (1), since the overall effect of a memoryless nonlinear transformation of a linear filter is a nonlinear mapping with memory. Other applications of Volterra filters include restoration of images degraded by quantum-limited imaging conditions. This type of degradation occurs in remote sensing and medical imaging, in which low levels of radiation are necessary to protect the patient.

The degraded image $g(n_1, n_2)$ is modelled as the output of a Volterra filter whose input is the original image $f(n_1, n_2)$, plus additive noise $v(n_1, n_2)$:

$$\begin{aligned} g(n_1, n_2) &= H[f(n_1, n_2)] + v(n_1, n_2) \\ &= h_0 + H_1[f(n_1, n_2)] + H_2[f(n_1, n_2)] \\ &\quad + v(n_1, n_2) \end{aligned} \tag{2}$$

where $h_0$ is a constant,

$$H_1[f(n_1, n_2)] = \sum_{(i_1,i_2)\in S} h_1(i_1, i_2) f(n_1 - i_1, n_2 - i_2) \tag{3}$$

is the linear Volterra operator,

$$\begin{aligned} &H_2[f(n_1, n_2)] = \\ &\sum_{(i_1,i_2,j_1,j_2)\in S} h_2(i_1, i_2, j_1, j_2) f(n_1 - i_1, n_2 - i_2) \cdot \\ &f(n_1 - j_1, n_2 - j_2) \end{aligned} \tag{4}$$

is the quadratic Volterra operator and $S$ is the support region of the filter.

As in the 1D case, the 2D Volterra kernels are symmetric in that any permutation of the variable couples $(i_1, i_2)$ and $(j_1, j_2)$ leaves $h_2(i_1, i_2, j_1, j_2)$ unchanged, i.e. $h_2(i_1, i_2, j_1, j_2) = h_2(j_1, j_2, i_1, i_2)$.

If the nonlinearity is pointwise, as in equation (1), then several observations can be made. Firstly, the particular symmetry of the linear blur $d(i_1, i_2)$ is preserved in the linear Volterra kernels. Furthermore, the symmetry can be extended to the quadratic kernels by considering the input products $d(i_1, i_2)d(j_1, j_2)$, $\forall i_1, i_2, j_1, j_2$. This provides a means of reducing the number of independent variables needed to model the Volterra system. Secondly, since it is generally assumed that $d(i_1, i_2) \geq 0$, the sign of the quadratic kernels is determined by the sign of $T^{(2)}(\overline{f})$, the second-derivative of $T$ evaluated at the image mean $\overline{f}$.

The following constraints are applied to the Volterra filter:

1. $h_0 = 0$;
2. $h_1(i_1, i_2) \geq 0$;
3. $\sum h_1(i_1, i_2) = 1$;
4. $h_2(i_1, i_2, j_1, j_2) \leq 0$.

The first three constraints follow from standard assumptions on linear degradations, namely, that the image is formed from radiant energy (which is unsigned), and that the energy is conserved in the blurring process when $h_2(\cdot) = 0$. The fourth constraint results from the particular case when $T$ is convex, i.e. the sensor compresses the intensity range of the image. It should be pointed out that the Volterra filter can model many types of image degradations, and the constraints depend on the particular application. The sensor nonlinearity is used as an example to illustrate one application of the Volterra filter to image restoration.

Blind deconvolution of linearly-degraded images by simulated annealing (SA) was first proposed by McCallum [2]. In this method, the following multi-modal cost function is minimised:

$$J(\hat{f}, \hat{h}_1) = \sum_{n_1,n_2} (g(n_1,n_2) - \hat{h}_1(n_1,n_2) * \hat{f}(n_1,n_2))^2, \quad (5)$$

where $*$ denotes convolution. The image and the blurring function are assumed to be positive with known finite support. The algorithm provides reasonable results in the presence of noise, but convergence to the global minimum is slow. For realistically-sized images, SA is too computationally intensive to produce good results.

A genetic algorithm (GA) has recently been applied to minimisation of equation (5) [3]. The original image was assumed to be binary with known finite support, and a blurring function of the following form was considered:

$$h_1(n_1,n_2) = \begin{cases} 1/\pi a^2, & n_1^2 + n_2^2 \leq a^2 \\ 0, & \text{otherwise} \end{cases}, \quad (6)$$

with unknown parameter $a$. However, GA's lack an efficient local search mechanism; while they are able to locate the neighbourhood of the optimum very quickly, they are not suited to fine-tuning of solutions. Chen et al [3] compensate for this by using an extremely large population (300 for the image and 10 for the blurring function). The large memory requirements make the basic GA unsuitable for processing realistically sized images.

In this paper, a hybrid SAGA is applied to minimisation of

$$\begin{aligned} J(\hat{f}, \hat{H}, \lambda) = & \\ & \sum_{n_1,n_2} (g(n_1,n_2) - \hat{H}[\hat{f}(n_1,n_2)])^2 \\ & + \frac{\lambda}{2} \sum_{n_1,n_2} W(n_1,n_2)(c(n_1,n_2) * \hat{f}(n_1,n_2))^2. \quad (7) \end{aligned}$$

The second term of (7) follows from a piecewise smoothness constraint on the original image [4]. The Laplace filter

$$c = \begin{bmatrix} 0 & -0.25 & 0 \\ -0.25 & 1 & -0.25 \\ 0 & -0.25 & 0 \end{bmatrix}$$

functions as the regularization operator, and $\lambda$ is known as the regularization parameter. The weight matrix $W(n_1,n_2)$ allows the amount of smoothing to be adjusted locally according to the variance of the degraded image. It is defined as

$$W(n_1,n_2) = \frac{1}{1 + \alpha\sigma^2(n_1,n_2)},$$

where $\sigma^2(n_1,n_2)$ is the local variance of the blurred image $g$ [4]. The constant $\alpha$ is chosen such that

$$\max(W(n_1,n_2))/\min(W(n_1,n_2)) = 2000.$$

The image is assumed to be positive, with known finite support.

## 2. SIMULATED ANNEALING AND GENETIC ALGORITHMS

Genetic algorithms are a stochastic search technique based on natural selection and genetics. GA's differ from conventional optimisation techniques in that they are parallel, probabilistic, and use only the objective function. They are superior to gradient-descent techniques, which are biased toward local optima. Although no formal proof of convergence exists for GA's, they are usually able to locate the neighbourhood of the optima quickly. However, it is known that GA's, in their basic form, are not well-suited to fine-tuning of solutions. Furthermore, GA's are difficult to apply to large-scale optimisation problems because of the large memory requirements.

GA's work with a population of individuals (or solutions), each of which is assigned a certain fitness. The particular traits of an individual are encoded in the chromosome, which consists of a string of parameters (genes). All GA's consist of the three basic operations - selection, crossover, and mutation. From the parent generation, individuals are selected for reproduction such that high-fitness individuals have a higher probability of passing on genetic material to the next generation. Recombination occurs through crossover and mutation. In crossover, individuals are randomly paired, and genetic information from each of the parents is combined with a certain probability to produce two new offspring. If no crossover occurs, then the parents are simply duplicated. Mutation provides a means of introducing new genes into the population, by allowing each gene of an individual to mutate with a certain probability.

Simulated annealing is a stochastic descent technique derived from statistical mechanics. If a system is in thermal equilibrium at temperature $T$, then the probability that the system is in a particular atomic configuration is $P(E) = \exp(-E/k_BT)$, where $E$ is the energy of the configuration, $k_B$ is Boltzmann's constant, and $T$ is the temperature. Simulated annealing attempts to reach the minimum-energy (cost) state through a series of atomic reconfigurations (local perturbations) which are accepted if the energy is decreased, and accepted with probability $P(\Delta E) = \exp(-\Delta E/T)$ if the energy is increased. For $T > 0$, there is always some probability that a detrimental step will be accepted, thus allowing the algorithm to escape from local minima.

Analysis of the algorithm as a Markov chain shows that SA will converge in distribution to the minimum energy states, provided that certain conditions on the local transitions and cooling schedule are met [5]. However, since the required cooling schedule is too slow to implement, faster approximations are used at the expense of a guarantee of optimality. In practise, SA is able to locate near-optimal solutions for many combinatorial optimisation and nonlinear programming problems.

The local search capability of a GA can be improved by implementing SA-type mutation and crossover. Adler [6] has developed a hybrid algorithm in which crossovers and mutations are accepted according to the SA criterion. This is equivalent to the standard GA when $T = \infty$, i.e. all crossovers and mutations are accepted. If the temperature is lowered gradually, the SAGA is able to maintain population diversity, while reducing disruption to the solutions through crossover and mutation as the population fitness increases. Furthermore, the improved local search mechanism enables the use of a much smaller population.

The convergence properties of the SAGA can be analysed by considering the population members individually and viewing crossover as a special form of local transition. Provided that the crossover operator meets the conditions specified in [5], the convergence properties of SA should be upheld in the SAGA. Furthermore, implementation of fitness selection does not interfere with the individual Markov chains.

## 3. THE SAGA ALGORITHM FOR IMAGE RESTORATION

The pixels of the original image are treated as continuous variables, on the assumption that the number of possible input levels is large ($> 20$), as in the case of an 8-bit image. This allows the implementation of efficient heuristic operators. Each chromosome is formed by mapping the image and filter estimates to a vector. The population is randomly initialised within the bounds for each variable, and the linear kernels are normalised so that $\sum h_1(i_1, i_2) = 1$.

The fitness is defined to be $f = -J(\hat{f}, \hat{H}, \lambda)$, as given by equation (7). The algorithm can be implemented with or without fitness selection. At lower temperatures, the implementation of a selection mechanism does not noticeably affect the algorithm's performance, since SA-type operations are sufficient to move the population toward maximal fitness. At higher temperatures, the algorithm relies more on the GA search.

If fitness selection is implemented, then individuals are chosen for reproduction by means of probabilistic binary tournament selection. Pairs of individuals are randomly chosen from the population, and the individual with higher fitness wins the tournament with probability $p_w \leq 1$. The winner is then copied into the next generation. This method of selection lessens the sensitivity of the algorithm to the fitness function, since the selection probability is not directly proportional to the fitness.

During crossover, chromosomes are randomly paired. For each pair of chromosomes, a mask $M$ of random numbers is generated in the range $[0, 1]$. The first child is produced by multiplying each gene $x(i)$ of the first parent by the mask value $M(i)$, and the genes of the second parent by the complementary mask $1 - M(i)$. This gives a weighted average of the genes from both parents. Before the cost is recalculated, the linear kernels are renormalised. The cost of the child is compared to that of the first parent, and acceptance is determined by the SA criterion, with the crossover temperature $T_c$ being set to the standard deviation of the fitness of the previous generation. The parents are then exchanged to produce the second child.

A mutation operator is applied to each gene with fixed probability $p_m$. The gene is mutated by an amount $\Delta x(i) = \alpha(i) r$, where $\alpha(i)$ is the step size and $r$ is a random number uniformly distributed in the range $[-1, 1]$. If the new value falls outside the feasible range, then the process is repeated until a suitable value is found. After mutation of a linear kernel, the kernels are renormalised before the cost is recalculated. For the image genes, it is not necessary to recalculate the entire cost function after each mutation, since only output values in a region of size $S$ are affected.

The step size for mutation is adapted according to the method developed by Corona et al [7] for SA, which attempts to maintain a 1:1 ratio between the number of accepted and rejected mutations. A single step size is used for the image, and this is updated after each generation. If the pixel lies outside of the image support, then $\alpha(i) = 0$. The step size for each Volterra kernel is updated after $N_S$ generations. The maximum step size for both the image and the kernels is defined to be the search range for that variable. The mutation temperature $T_m$ is decreased by a factor of $r_T$ every $N_T N_S$ generations.

## 4. SIMULATIONS

To reduce the number of variables (for a $3 \times 3$ linear support, there are 45 distinct second-order kernels), it was assumed that the linear part of the degradation was of the form

$$H_1 = \begin{bmatrix} l3 & l2 & l3 \\ l2 & l1 & l2 \\ l3 & l2 & l3 \end{bmatrix}, \tag{8}$$

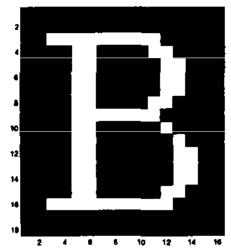
or in vector notation

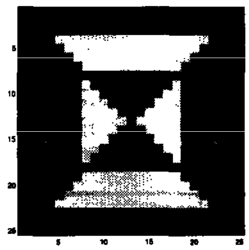
$$\mathbf{h}_1 = \begin{bmatrix} l3 & l2 & l3 & l2 & l1 & l2 & l3 & l2 & l3 \end{bmatrix}^{\mathrm{T}}. \tag{9}$$

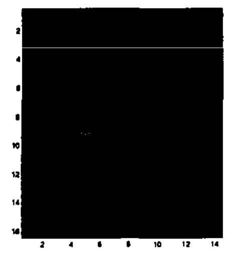
For a pointwise nonlinearity, this would yield

$$H_2 = \begin{bmatrix} q3 & q6 & q3 & q6 & q5 & q6 & q3 & q6 & q3 \\ q6 & q2 & q6 & q2 & q4 & q2 & q6 & q2 & q6 \\ q3 & q6 & q3 & q6 & q5 & q6 & q3 & q6 & q3 \\ q6 & q2 & q6 & q2 & q4 & q2 & q6 & q2 & q6 \\ q5 & q4 & q5 & q4 & q1 & q4 & q5 & q4 & q5 \\ q6 & q2 & q6 & q2 & q4 & q2 & q6 & q2 & q6 \\ q3 & q6 & q3 & q6 & q5 & q6 & q3 & q6 & q3 \\ q6 & q2 & q6 & q2 & q4 & q2 & q6 & q2 & q6 \\ q3 & q6 & q3 & q6 & q5 & q6 & q3 & q6 & q3 \end{bmatrix}, \tag{10}$$

where the $(i, j)$-th entry corresponds to the quadratic kernel $h_2(i, j)$ under the mapping in equation (9).


(a) Binary Image


(b) Geometric Image


(c) Degraded Binary Image

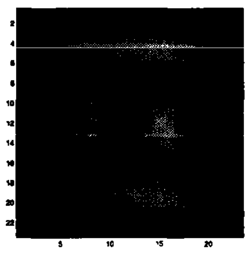
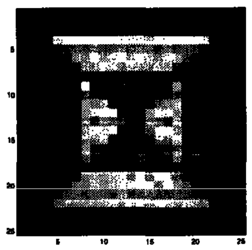
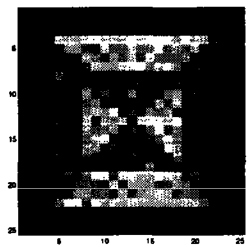
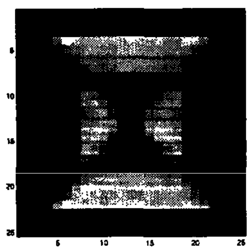

(d) Degraded Geometric Image

Figure 1: Original and Degraded Images


(a) No noise ($\lambda = 0$)


(b) 30 dB noise ($\lambda = 0$)


(c) 30 dB noise ($\lambda = 1$)

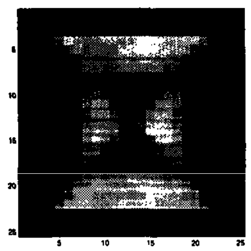
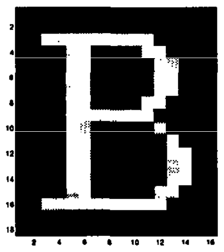

(d) 20 dB noise ($\lambda = 10$)

Figure 2: Restored Geometric Image


(a) 30 dB noise ($\lambda = 0$)

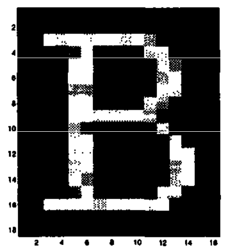

(b) 20 dB noise ($\lambda = 0$)

Figure 3: Restored Binary Image

The algorithm was tested on two different images — a binary image of the letter B, and a geometric image, as shown in Figure 1 (a) and (b). The degradation parameters for both images are listed below.

| | |
|---|---|
| $l_1$ | 0.1479 |
| $l_2$ | 0.1183 |
| $l_3$ | 0.0947 |
| $q_1$ | −0.0066 |
| $q_2$ | −0.0042 |
| $q_3$ | −0.0027 |
| $q_4$ | −0.0053 |
| $q_5$ | −0.0042 |
| $q_6$ | −0.0034 |

Figure 1 (c) and (d) show the degraded images with 30 dB noise.

The search ranges for the image, linear kernels, and quadratic kernels were defined to be $[0, 1]$, $[0, 1]$, and $[-1, 0]$, respectively. In all simulations shown, a population size $N_p = 4$ was used. The initial step size was set to one-fifth the search range for each variable. The kernel step sizes were updated every $N_s = 20/(p_m N_p)$ generations. The mutation temperature was initialised to $T_0 = 0.1$ and decreased by a factor of $r_T = 0.85$ after $N_T = 2$ updates of the step size. For the results listed in Tables 1 and 2, no fitness selection was used. However, when several of the simulations were re-run with fitness selection, there was only small improvement. At higher temperatures, the difference in performance was more significant.

The algorithm was tested on the degraded geometric image without noise, with 30 dB noise, and with 20 dB noise. In the noiseless case, the regularization parameter in equation (7) was set to 0. Although the SAGA was able to find image and kernel estimates for which the cost was very small ($\mathcal{O}(10^{-4})$), the restored image was very noisy, as shown in Figure 2 (a). In this case, the regularization term can be used to incorporate additional information about the original image $f$, based on the characteristics of $g$, into the restoration. Without this information, it seems that for multi-level images, the problem is too ill-defined to obtain a good image estimate. For lower SNR's, it was necessary to use larger regularization terms, as shown in Figure 2 (b)-(d). Even though the piecewise smoothness constraint produced better image estimates, the corresponding filter

Table 1: Restoration of a Geometric Image

| SNR (dB) | $\infty$ | 30 | 30 | 20 |
|---|---|---|---|---|
| $\lambda$ | 0 | 0 | 1 | 10 |
| $p_m$ | 0.25 | 0.25 | 0.25 | 0.25 |
| gen. | 4000 | 2000 | 2000 | 2000 |
| feval | 1476376 | 738919 | 738466 | 738918 |
| $l_1$ | 0.1634 | 0.1499 | 0.2466 | 0.3626 |
| $l_2$ | 0.1132 | 0.1174 | 0.0637 | 0.0002 |
| $l_3$ | 0.0959 | 0.0951 | 0.1246 | 0.1591 |
| $q_1$ | -0.0120 | -0.0021 | -0.0002 | -0.0002 |
| $q_2$ | -0.0034 | -0.0036 | -0.0019 | -0.0023 |
| $q_3$ | -0.0050 | -0.0022 | -0.0032 | -0.0025 |
| $q_4$ | -0.0026 | -0.0019 | -0.0028 | -0.0000 |
| $q_5$ | -0.0031 | -0.0048 | -0.0064 | -0.0012 |
| $q_6$ | -0.0006 | -0.0018 | -0.0022 | -0.0032 |
| MSE | 2.2040 | 3.6517 | 1.1671 | 1.8131 |
| SNRI | 3.3489 | 2.0151 | 6.3051 | 4.1276 |
| $J(\hat{f},\hat{H},\lambda)$ | $2.87 \times 10^{-4}$ | 0.0157 | 0.0665 | 0.3324 |
| $J(f,H,\lambda)$ | 0 | 0.0250 | 0.1738 | 0.9184 |

Table 2: Restoration of a Binary Image

| SNR (dB) | 30 | 20 |
|---|---|---|
| $\lambda$ | 0 | 0 |
| $p_m$ | 0.125 | 0.25 |
| gen. | 4500 | 2000 |
| feval | 413255 | 351576 |
| $l_1$ | 0.1503 | 0.1403 |
| $l_2$ | 0.1189 | 0.1231 |
| $l_3$ | 0.0935 | 0.0919 |
| $q_1$ | -0.0005 | -0.0005 |
| $q_2$ | -0.0040 | -0.0035 |
| $q_3$ | -0.0016 | -0.0002 |
| $q_4$ | -0.0055 | -0.0039 |
| $q_5$ | -0.0056 | -0.0044 |
| $q_6$ | -0.0024 | -0.0022 |
| MSE | 0.1718 | 1.5741 |
| SNRI | 191.1583 | 21.0648 |
| $J(\hat{f},\hat{H},\lambda)$ | 0.0080 | 0.0703 |
| $J(f,H,\lambda)$ | 0.0089 | 0.0889 |

estimates were much worse. These are listed in Table 1. As a measure of the goodness of the restoration, the percentage mean square error (MSE) and improvement in signal-to-noise ratio (SNRI) were used. They were defined as follows:

$$\text{MSE}(\hat{f}) = 100\frac{\sum_{n_1,n_2}(a\hat{f}(n_1,n_2) - f(n_1,n_2))^2}{\sum_{n_1,n_2} f^2(n_1,n_2)}, \quad (11)$$

$$\text{SNRI} = \frac{\text{MSE(g)}}{\text{MSE}(\hat{\text{f}})}, \quad (12)$$

where $a = \sum_{n_1,n_2} f(n_1,n_2)\hat{f}(n_1,n_2) / \sum_{n_1,n_2} \hat{f}^2(n_1,n_2)$. In the last two rows of the table, the costs of the estimated and actual image-filter convolutions are compared.

For the degraded binary image, it was not necessary to use a regularization term to obtain reasonable image estimates for mild noise levels. The results of the SAGA for 30 dB and 20 dB noise are shown in Figure 3 and Table 2.

Corona's SA algorithm was tested against the SAGA by setting $N_p = 1$ and $P_m = 1$. While the number of function evaluations (feval) for both algorithms was approximately the same, the advantage of the SAGA is that most of these evaluations can be done in parallel, since they arise from SA mutation. In the case of the geometric image with 30 dB noise and $\lambda = 0$, the SA algorithm found estimates with $J(\hat{f},\hat{H},\lambda) = 0.0183$ and MSE = 4.6526 in 2000 scans (feval = 722361). This can be compared to the results in Table 1. The improvement in MSE may be due to the averaging which occurs during crossover. However, before further conclusions about the efficiency of the algorithms can be made, the effect of the population size and mutation rate must be studied more thoroughly.

## 5. CONCLUSIONS

In this paper, a method of combining simulated annealing and genetic algorithms was presented for image restoration. The degradation was modelled by a Volterra filter, and a set of constraints were developed which incorporated partial knowledge of the nonlinearity under consideration. Areas for further research include the effect of the constraints on the solution, and the influence of the SAGA parameters on the algorithm's performance. It may also be interesting to explore how modification of the SAGA for multi-objective optimisation could be used to fine-tune the regularization parameter.

## 6. REFERENCES

[1] A. K. Katsaggelos, editor. *Digital Image Restoration.* Springer-Verlag, New York, 1991.

[2] B. C. McCallum. Blind deconvolution by simulated annealing. *Optics Communications*, 75(2):101–105, 1990.

[3] Yen-Wei Chen, Zensho Nakao, and Shinchi Tamura. Blind deconvolution by genetic algorithms. In *Proceedings of SPIE — The International Society for Optical Engineering*, volume 2662, pages 192–196, 1996.

[4] Yu-Li Lou and M. Kaveh. A regularization approach to joint blur identification and image restoration. *IEEE Transactions on Image Processing*, 5(3):416–427, March 1996.

[5] Emile Aarts and Jan Korst. *Simulated Annealing and Boltzmann Machines.* John Wiley and Sons, Chichester, 1989.

[6] D. Adler. Genetic algorithms and simulated annealing: a marriage proposal. In *IEEE International Conference on Neural Networks*, pages 1104–1109, 1993.

[7] A. Corona, M. Marchesi, C. Martini, and S. Ridella. Minimizing multi-modal functions of continuous variables with the simulated annealing algorithm. *ACM Transactions on Mathematical Software*, 13(3):262–280, September 1987.