Signal Processing Applications Using Adaptive Simulated Annealing

S. Chen †, R.H. Istepanian ‡ and B.L. Luk †

† Department of Electrical and Electronic Engineering
University of Portsmouth, Anglesea Building
Portsmouth PO1 3DJ, U.K.

‡ Department of Electrical and Computer Engineering
Ryerson Polytechnic University, 350 Victoria Street
Toronto, Ontario, Canada, M5B 2K3

Abstract– Many signal processing applications pose optimization problems with multimodal and nonsmooth cost functions. Gradient methods are ineffective in these situations. The adaptive simulated annealing (ASA) offers a viable optimization tool for tackling these difficult nonlinear problems. We demonstrate the effectiveness of the ASA using three applications, infinite-impulse-response (IIR) filter design, maximum likelihood (ML) joint channel and data estimation and evaluation of minimum symbol-error-rate (MSER) decision feedback equalizer (DFE).

1 Introduction

Optimization problems with multimodal and nonsmooth cost functions are commonly encountered in signal processing applications. Gradient-based algorithms are ineffective in these applications due to the problem of local minima or the difficulty in calculating gradients. Optimization methods that require no gradient and can achieve a global optimal solution offer considerable advantages in solving these difficult problems. Two best-known global optimization methods are the genetic algorithm (GA) [1]–[3] and simulated annealing (SA) [4]–[6].

The GA and SA belong to a class of so-called guided random search methods. The underlying mechanisms for guiding search process are, however, very different for the two methods. The GA is population based, and evolves a solution population according to the principles of the evolution of species in nature. The SA, on the other hand, evolves a single solution in the parameter space with certain guiding principles that imitate the random behaviour of molecules during the annealing process.

While the GA seems to have attracted considerable attention in signal processing applications (e.g. [7]–[9]), the SA by contrast has not received similar interests. The SA is an optimization technique with some strikingly positive and negative features. An attractive feature of SA is that it is very easy to program and the algorithm typically has few parameters that require tuning. A serious drawback of SA is that the standard SA can be very slow, often requiring much more number of cost-function evaluations to converge, compared with a carefully designed and tuned GA.

The ASA, an improved version of SA also known as the very fast simulated reannealing [10]–[13], provides significant improvement in convergence speed over standard versions of SA and maintains all the advantages of standard SA algorithms. To illustrate its simplicity and versatility, we apply the ASA to three signal processing applications, IIR filter design, ML joint channel and data estimation and evaluation of the MSER DFE. Our study demonstrates that the ASA offers a viable approach for solving diverse signal processing problems.

2 Adaptive simulated annealing

Many signal processing applications pose the following optimization problem:

$$\min_{\mathbf{w}\in\mathcal{W}} J(\mathbf{w}), \tag{1}$$

where $\mathbf{w} = [w_1 \cdots w_n]^T$ is the n-dimensional parameter vector to be optimized, $\mathcal{W}$ is defined by

$$\mathcal{W} \triangleq \{\mathbf{w} : (L_i \le w_i \le U_i, 1 \le i \le n) \cap$$
$$(\alpha_j \le g_j(\mathbf{w}) \le \beta_j, 1 \le j \le m)\} \ , \tag{2}$$

L_i and U_i are the lower and upper bounds of w_i, and $\alpha_j \le g_j(\mathbf{w}) \le \beta_j$ are inequality constraints. The cost function $J(\mathbf{w})$ can be multimodal and nonsmooth. The ASA is an efficient global optimization scheme for solving this kind of constrained optimization problems.

0-7803-5536-9/99/$10.00 ©1999 IEEE

2.1 Search guiding mechanisms

The ASA evolves a single point $\mathbf{w}$ in the parameter or state space $\mathcal{W}$. The seemingly random search is guided by certain underlying probability distributions. An elegant discussion on how the general SA algorithm works can be found in [13]. Specifically, the general SA algorithm is described by three functions.

1. Generating probability density function

$$G(w_i^{\text{old}}, w_i^{\text{new}}, T_{i,\text{gen}}; 1 \leq i \leq n) .$$

This determines how a new state $\mathbf{w}^{\text{new}}$ is created, and from what neighbourhood and probability distributions it is generated, given the current state $\mathbf{w}^{\text{old}}$. The generating "temperatures" $T_{i,\text{gen}}$ describe the widths or scales of the generating distribution along each dimension w_i of the state space.

Often a cost function has different sensitivities along different dimensions of the state space. Ideally, the generating distribution used to search a steeper and more sensitive dimension should have a narrower width than that of the distribution used in searching a dimension less sensitive to change. The ASA adopts a so-called re-annealing scheme to periodically re-scale $T_{i,\text{gen}}$, so that they optimally adapt to the current status of the cost function. This is an important mechanism, which not only speeds up the search process but also makes the optimization process robust to different problems.

2. Acceptance function

$$P_{\text{accept}}(J(\mathbf{w}^{\text{old}}), J(\mathbf{w}^{\text{new}}), T_{\text{accept}}) .$$

This gives the probability of $\mathbf{w}^{\text{new}}$ being accepted. The acceptance temperature determines the frequency of accepting new states of poorer quality.

Probability of acceptance is very high at very high temperature T_{accept}, and it becomes smaller as T_{accept} is reduced. At every acceptance temperature, there is a finite probability of accepting the new state. This produces occasionally uphill move, enables the algorithm to escape from local minima, and allows a more effective search of the state space to find a global minimum. The ASA also periodically adapts T_{accept} to best suit the status of the cost function. This helps to improve convergence speed and robustness.

3. Reduce temperatures or annealing schedule

$$\left.\begin{array}{l} T_{\text{accept}}(k_a) \longrightarrow T_{\text{accept}}(k_a+1) \\ T_{i,\text{gen}}(k_i) \longrightarrow T_{i,\text{gen}}(k_i+1), \; 1 \leq i \leq n \end{array}\right\} ,$$

where k_a and k_i are some annealing time indexes. The reduction of temperatures should be sufficiently gradual in order to ensure that the algorithm finds a global minimum.

This mechanism is based on the observations of the physical annealing process. When the metal is cooled from a high temperature, if the cooling is sufficiently slow, the atoms line themselves up and form a crystal, which is the state of minimum energy in the system. The slow convergence of many SA algorithms is rooted at this slow annealing process. The ASA, however, can employ a very fast annealing schedule, as it has self adaptation ability to re-scale temperatures.

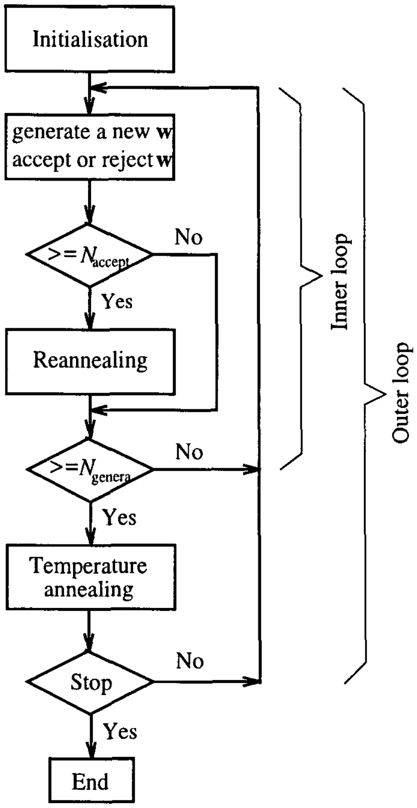


Figure 1: Flow chart of ASA.

2.2 Algorithm implementation

Although there are many realizations of the ASA, an implementation is illustrated in Fig. 1, and this algorithm is detailed here. How the ASA realizes the above three functions will also become clear during the description.

(i) In the initialization, an initial $\mathbf{w} \in \mathcal{W}$ is randomly generated, the initial temperature of the acceptance probability function, $T_{\text{accept}}(0)$, is set to $J(\mathbf{w})$, and the initial temperatures of the parameter generating probability functions $T_{i,\text{gen}}(0)$ are set to 1.0. An annealing control parameter c is given, and the annealing times k_i and k_a, are set to 0.

(ii) The algorithm generates a new point in the parameter space with:

$$w_i^{\text{new}} = w_i^{\text{old}} + q_i\left(U_i - L_i\right),\ 1 \le i \le n, \text{ and}$$

$$\mathbf{w}^{\text{new}} \in \mathcal{W}, \tag{3}$$

where q_i is calculated as

$$q_i = \text{sgn}\left(v_i - \frac{1}{2}\right) T_{i,\text{gen}}(k_i) \times \left(\left(1 + \frac{1}{T_{i,\text{gen}}(k_i)}\right)^{|2v_i - 1|} - 1\right), \tag{4}$$

and v_i a uniformly distributed random variable in [0, 1]. If a generated $\mathbf{w}^{\text{new}}$ is not in $\mathcal{W}$, it is discarded and a new point is tried again until $\mathbf{w}^{\text{new}} \in \mathcal{W}$. The value of the cost function $J(\mathbf{w}^{\text{new}})$ is then evaluated and the acceptance probability function of $\mathbf{w}^{\text{new}}$ is given by

$$P_{\text{accept}} = \frac{1}{1 + \exp\left(\frac{J(\mathbf{w}^{\text{new}}) - J(\mathbf{w}^{\text{old}})}{T_{\text{accept}}(k_a)}\right)}. \tag{5}$$

A uniform random variable P_{unif} is generated in [0, 1]. If $P_{\text{unif}} \le P_{\text{accept}}$, $\mathbf{w}^{\text{new}}$ is accepted; otherwise it is rejected.

(iii) After every N_{accept} acceptance points, reannealing takes place by first calculating the sensitivities

$$s_i = \left|\frac{J(\mathbf{w}^{\text{best}} + \mathbf{e}_i\delta) - J(\mathbf{w}^{\text{best}})}{\delta}\right|,\ 1 \le i \le n, \tag{6}$$

where $\mathbf{w}^{\text{best}}$ is the best point found so far, δ is a small step size, the n-dimensional vector $\mathbf{e}_i$ has unit ith element and the rest of elements of $\mathbf{e}_i$ are all zeros. Let $s_{\max} = \max\{s_i,\ 1 \le i \le n\}$. Each $T_{i,\text{gen}}$ is scaled by a factor $s_{\max}/s_i$ and the annealing time k_i is reset

$$\left.\begin{array}{l} T_{i,\text{gen}}(k_i) = \frac{s_{\max}}{s_i} T_{i,\text{gen}}(k_i) \\ k_i = \left(-\frac{1}{c}\log\left(\frac{T_{i,\text{gen}}(k_i)}{T_{i,\text{gen}}(0)}\right)\right)^n \end{array}\right\}. \tag{7}$$

Similarly, $T_{\text{accept}}(0)$ is reset to the value of the last accepted cost function, $T_{\text{accept}}(k_a)$ is reset to $J(\mathbf{w}^{\text{best}})$ and the annealing time k_a is rescaled accordingly

$$k_a = \left(-\frac{1}{c}\log\left(\frac{T_{\text{accept}}(k_a)}{T_{\text{accept}}(0)}\right)\right)^n. \tag{8}$$

(iv) After every N_{genera} generated points, annealing takes place for $1 \le i \le n$

$$\left.\begin{array}{l} k_i = k_i + 1 \\ T_{i,\text{gen}}(k_i) = T_{i,\text{gen}}(0)\exp\left(-ck_i^{\frac{1}{n}}\right) \end{array}\right\} \tag{9}$$

and

$$\left.\begin{array}{l} k_a = k_a + 1 \\ T_{\text{accept}}(k_a) = T_{\text{accept}}(0)\exp\left(-ck_a^{\frac{1}{n}}\right) \end{array}\right\}; \tag{10}$$

Otherwise, goto step (ii).

(v) The algorithm is terminated if the parameters have remained unchanged for a few successive reannealings or a preset maximum number of cost function calls has been reached; Otherwise, goto step (ii).

As in a standard SA algorithm, this ASA contains two loops. The inner loop ensures that the parameter space is searched sufficiently at a given temperature, which is necessary to guarantee that the algorithm finds a global optimum. The differences with standard SA algorithms are that the ASA uses a much faster annealing schedule and employs a reannealing scheme to adapt itself.

2.3 Algorithm parameter tuning

The ASA uses only the value of the cost function in the optimization process and is very simple to program, just as in a standard SA algorithm. For the above ASA algorithm, most of the algorithm parameters are automatically set and "tuned", and the user only needs to assign a control parameter c and set two values N_{accept} and N_{genera}.

Obviously, the optimal values of N_{accept} and N_{genera} are problem dependent, but our experience suggests that an adequate choice for N_{accept} is in the range of tens to hundreds and an appropriate value for N_{genera} is in the range of hundreds to thousands. The annealing rate control parameter c can be determined form the chosen initial temperature, final temperature and predetermined number of annealing steps [11],[12]. We have found out that a choice of c in the range 1.0 to 10.0 is often adequate.

It should be emphasized that, as the ASA has excellent self adaptation ability, the performance of the algorithm is not critically influenced by the specific chosen values of c, N_{accept} and N_{genera}. This has been observed in a variety of applications.

3 Optimization applications

The versatility of the ASA as a global optimization tool is demonstrated on the three very different problems.

3.1 IIR filter design

Consider the IIR filter with the transfer function:

$$H_M(z) = \frac{A(z)}{B(z)} = \frac{\sum_{i=0}^{L} a_i z^{-i}}{1 + \sum_{i=1}^{M} b_i z^{-i}}, \tag{11}$$

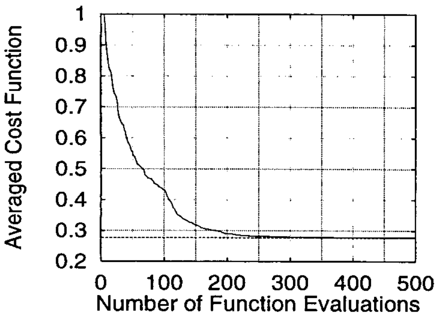


Figure 2: Normalized cost function versus number of cost function evaluations averaged over 100 random runs using the ASA for Example 1. The dashed line indicates the global minimum.

where $M(\geq L)$ is the filter order. The IIR filter design can be formulated as an optimization problem with the mean square error (MSE) as the cost function

$$J(\mathbf{w}_H) = \mathrm{E}[(d(k) - y(k))^2], \tag{12}$$

where $\mathbf{w}_H = [a_0\ a_1 \cdots a_L\ b_1 \cdots b_M]^T$ denotes the filter coefficient vector, $y(k)$ and $d(k)$ are the filter's output and desired response, respectively. In practice, ensemble operation is difficult to realize, and the cost function (12) is usually substituted by the time-averaged one

$$J_N(\mathbf{w}_H) = \frac{1}{N}\sum_{k=1}^{N}(d(k) - y(k))^2. \tag{13}$$

A major concern is that the cost function is generally multimodal, and a gradient algorithm can easily be stuck at local minima. The GA has been applied to IIR filter design (e.g. [8],[14],[15]) to overcome this difficulty. We show that the ASA offers an alternative. To maintain the stability during optimization, the direct-form coefficients b_i, $1 \leq i \leq M$, are converted into the lattice-form reflection coefficients κ_i, $0 \leq i \leq M-1$. Thus the filter coefficient vector used in optimization is $\mathbf{w} = [a_0\ a_1 \cdots a_L\ \kappa_0 \cdots \kappa_{M-1}]^T$, with the constraints $|\kappa_i| < 1$. Converting the reflection coefficients back to the direct-form coefficients is straightforward [16].

System identification application is used in the experiment. The unknown plant has a transfer function $H_S(z)$, and the ASA is employed to adjust the IIR filter that is used to model the system. When the filter order M is smaller than the system order, local minima problems can be encountered [17], and this is used to simulate a multimodal environment. Two examples were tested.

Example 1. This example is taken from [17]. The system and filter transfer functions are respectively

$$\left.\begin{array}{l} H_S(z) = \frac{0.05-0.4z^{-1}}{1-1.1314z^{-1}+0.25z^{-2}} \\ H_M(z) = \frac{a_0}{1+b_1 z^{-1}} \end{array}\right\} . \tag{14}$$

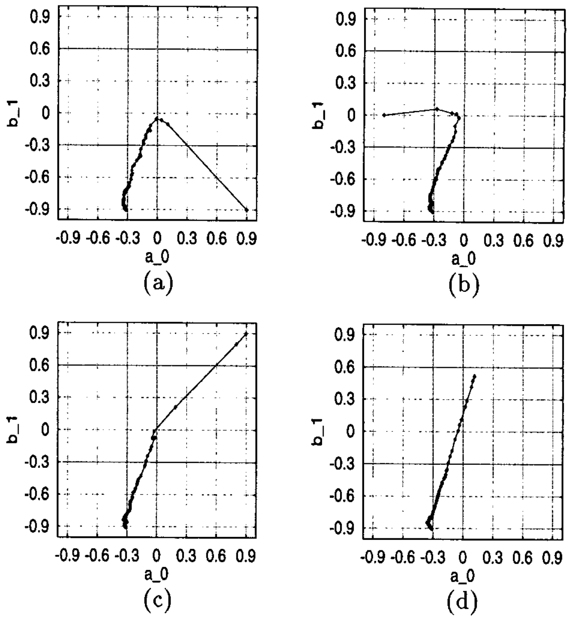


Figure 3: Filter parameter trajectories averaged over 100 runs using the ASA for Example 1, started from the fixed initial positions: (a) $[0.9\ -0.9]^T$, (b) $[-0.8\ 0.0]^T$, (c) $[0.9\ 0.9]^T$ and (d) $[0.114\ 0.519]^T$.

The analytical MSE (12) in this case is known when the input is a white sequence and the noise is absent. The MSE has a global minimum at $\mathbf{w}^{\text{global}} = [-0.311\ -0.906]^T$ with the normalized MSE value 0.2772, and a local minimum at $\mathbf{w}^{\text{local}} = [0.114\ 0.519]^T$. Fig. 2 depicts the evolution of the normalized MSE averaged over 100 random runs. Fig. 3 shows the trajectories of the filter parameter vector averaged over 100 runs, started from four fixed initial positions. It can be seen that the ASA consistently found the global optimal solution.

Example 2. This is a $3rd$ order system with the transfer function given by

$$H_S(z) = \frac{-0.3 + 0.4z^{-1} - 0.5z^{-2}}{1 - 1.2z^{-1} + 0.5z^{-2} - 0.1z^{-3}}. \tag{15}$$

The system input was a uniform white sequence taking values in $(-0.5, 0.5)$, and the signal to noise ration (SNR) was 30 dB. The data length used in calculating the MSE (13) was $N = 2000$. When a reduced-order filter with $M = 2$ and $L = 1$ was used, the MSE was multimodal. Extensive simulation showed that the MSE had a global minimum of 0.059. The ASA consistently reached this global minimum, as shown in Fig. 4. To illustrate the multimodal nature of the cost function, Fig. 5 shows the behaviours of a standard gradient algorithm.

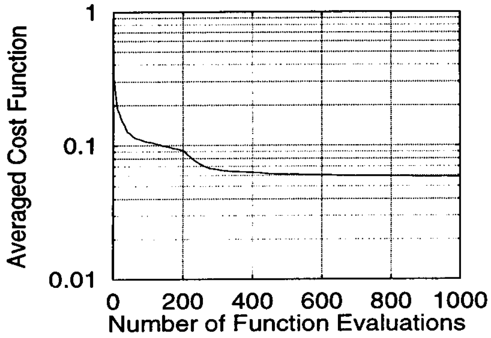


Figure 4: Cost function versus number of cost function evaluations averaged over 100 random runs using the ASA for Example 2.

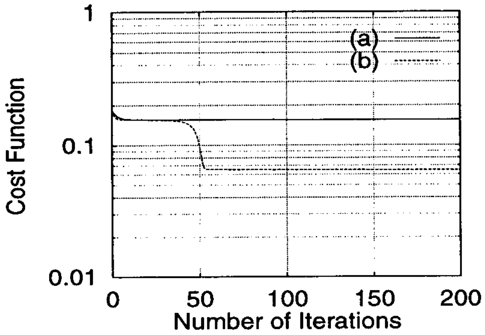


Figure 5: Convergence behaviours of the gradient algorithm, started from the two initial conditions: (a) $[0.0\ 0.0\ 0.3\ 0.1]^T$ and (b) $[0.0\ 0.0\ 0.3\ 0.0]^T$, for Example 2.

3.2 ML joint channel and data estimation

Consider the digital communication channel modelled as a finite impulse response filter with an additive noise source. The received signal at sample k is given by

$$r(k) = \sum_{i=0}^{n_a-1} a_i s(k-i) + e(k), \qquad (16)$$

where n_a is the channel length, a_i are the channel taps, the symbol sequence $\{s(k)\}$ is independently identically distributed with an M-PAM symbol constellation, and $e(k)$ is a Gaussian white noise. Let

$$\left.\begin{array}{l} \mathbf{r} = [r(1)\ r(2) \cdots r(N)]^T \\ \mathbf{s} = [s(-n_a+2) \cdots s(0)\ s(1) \cdots s(N)]^T \\ \mathbf{a} = [a_0\ a_1 \cdots a_{n_a-1}]^T \end{array}\right\} \qquad (17)$$

be the vector of N received data samples, the transmitted data sequence and the channel tap vector, respectively.

The joint ML estimate of $\mathbf{a}$ and $\mathbf{s}$ is obtained by maximizing the conditional probability density function of $\mathbf{r}$ given $\mathbf{a}$ and $\mathbf{s}$. Equivalently, the ML solution is the minimum of the cost function

$$J_c(\mathbf{a}, \mathbf{s}) = \sum_{k=1}^{N} \left(r(k) - \sum_{i=0}^{n_a-1} a_i s(k-i) \right)^2, \qquad (18)$$

that is,

$$(\mathbf{a}^*, \mathbf{s}^*) = \arg \left[\min_{\mathbf{a},\, \mathbf{s}} J_c(\mathbf{a}, \mathbf{s}) \right]. \qquad (19)$$

This joint ML estimate, however, is too expensive to compute except for the simplest case. In practice, suboptimal solutions are adopted for computational purpose. The algorithm based on a blind trellis search technique [18] is such an example.

The joint minimization process (19) can also be performed using an iterative loop first over the data sequences $\mathbf{s}$ and then over all the possible channels $\mathbf{a}$

$$(\mathbf{a}^*, \mathbf{s}^*) = \arg \left[\min_{\mathbf{a}} \left(\min_{\mathbf{s}} J_c(\mathbf{a}, \mathbf{s}) \right) \right]. \qquad (20)$$

The inner optimization can be carried out using the Viterbi algorithm (VA). The previous research has used the quantized channel algorithm [19] and the GA [9] to perform the outer optimization. In this study, we apply the ASA to perform the outer optimization. Specifically, given the channel estimate $\hat{\mathbf{a}}$, let the data sequence decoded by the VA be $\hat{\mathbf{s}}^*$. The cost function used by the ASA is the MSE:

$$J(\hat{\mathbf{a}}) = \frac{1}{N} J_c(\hat{\mathbf{a}}, \hat{\mathbf{s}}^*). \qquad (21)$$

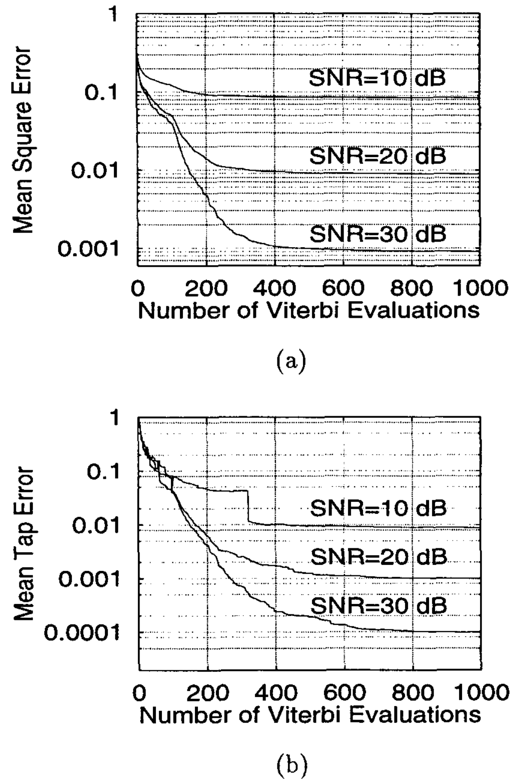
The search range for each channel tap is $-1.0 \le a_i \le 1.0$, since the channel can always be normalized.

The following numerical example was used to illustrate the combined ASA and VA approach for ML joint channel and data estimation. The channel was given by

$$\mathbf{a} = [0.407\ 0.815\ 0.407]^T. \qquad (22)$$

Because the true channel length $n_a = 3$ is unknown, an estimated length $\hat{n}_a = 4$ was assumed in the simulation. In practice, the performance of the algorithm is observed through the MSE. In simulation, the performance of the algorithm can also be assessed by the mean tap error (MTE), defined as $\text{MTE} = \|\hat{\mathbf{a}} - \mathbf{a}\|^2$.

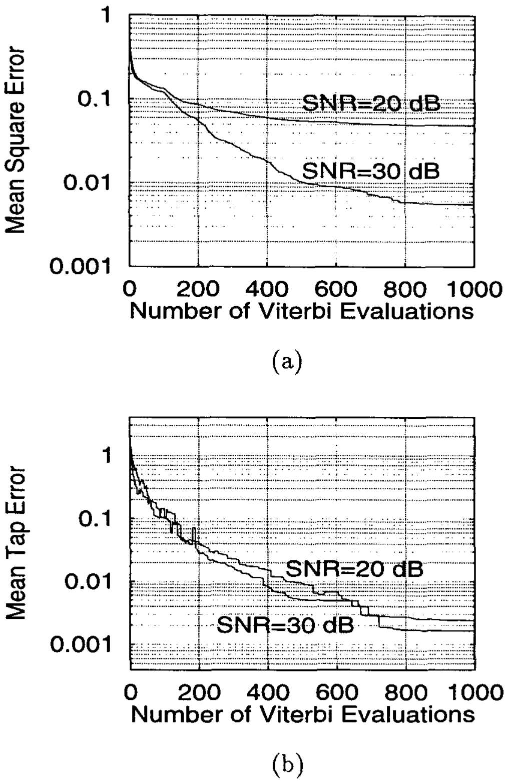
Figs. 6 and 7 show the evolutions of the MSE and MTE with 2-PAM and 4-PAM symbols and different SNRs, respectively. All the results were averaged over 100 different runs. Each run had a different noisy received data sequence and a different random initialization of the algorithm. No divergence was observed for any run. It can be seen from Figs. 6 and 7 that the MSE converged to the noise floor.



(a)

(b)

Figure 6: MSE (a) and MTE (b) against number of VA evaluations averaged over 100 different runs. 2-PAM and data samples $N = 50$.



(a)

(b)

Figure 7: MSE (a) and MTE (b) against number of VA evaluations averaged over 100 different runs. 4-PAM and data samples $N = 100$.

3.3 MSER decision feedback equalizer

DFE, shown in Fig. 8, is a powerful technique for combating distortion and interference in communication links [20],[21] and high-density data storage systems [22],[23], and is widely used in practice, as it provides a good balance between performance and complexity. For the channel defined in (16), the DFE produces an estimate $\hat{s}(k-d)$ of $s(k-d)$ by quantizing the filter output

$$f(\mathbf{r}(k), \hat{\mathbf{s}}_b(k)) = \mathbf{w}^T\mathbf{r}(k), +\mathbf{b}^T\hat{\mathbf{s}}_b(k) \qquad (23)$$

where $\mathbf{w} = [w_0 \cdots w_{m-1}]^T$ and $\mathbf{b} = [b_1 \cdots b_n]^T$ are the coefficients of the feedforward and feedback filters respectively, $\mathbf{r}(k) = [r(k) \cdots r(k-m+1)]^T$ is the channel output vector and $\hat{\mathbf{s}}_b(k) = [\hat{s}(k-d-1) \cdots \hat{s}(k-d-n)]^T$ is the past detected symbol vector. It is sufficient to choose $d = n_a - 1$, $m = n_a$ and $n = n_a - 1$ (see [24],[25]).

The minimum MSE (MMSE) solution ($\mathbf{w}_{\mathrm{MMSE}}$, $\mathbf{b}_{\mathrm{MMSE}}$) is often said to be the optimal solution for the coefficients of the DFE [26]. However, the MMSE solution does not correspond to the MSER solution, the symbol error rate (SER) being the ultimate performance criterion of equalization. It can be shown that the decision feedback $\mathbf{b}^T_{\mathrm{MMSE}}\hat{\mathbf{s}}_b(k)$ in a DFE performs a space translation which mapps the DFE onto a "linear" equalizer in the translated observation space [24],[25]:

$$f'(\mathbf{r}'(k)) = \mathbf{w}^T\mathbf{r}'(k), \qquad (24)$$

where $\mathbf{r}'(k)$ is the translated observation vector. This equivalent DFE is depicted in Fig. 9. In principle given the channel model, the analytic expression of the SER, $P_E(\mathbf{w})$, for the DFE with the weight vector $\mathbf{w}$ can be derived and the MSER solution $\mathbf{w}_{\mathrm{MSER}}$ can be obtained by minimizing $P_E(\mathbf{w})$. Furthermore, it becomes clear that the MMSE solution is not the MSER solution.

For the 2-PAM case, the gradient algorithm has been used to optimize $P_E(\mathbf{w})$ to obtain the MSER solution [24],[25]. For high order PAM channels, however, the optimization based on gradient method becomes highly complex and very costly. We propose a Monte Carlo approach based on the ASA to achieve the MSER solution for the general M-PAM channel. The SER can be approximated as

$$\hat{P}_E(\mathbf{w}) = \frac{1}{N}\sum_{k=1}^{N}\delta(\hat{s}(k-d) - s(k-d)), \qquad (25)$$

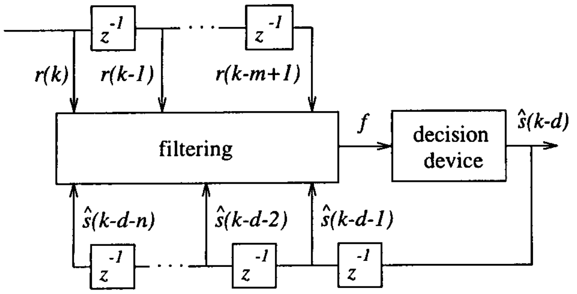


Figure 8: Schematic of a generic DFE.

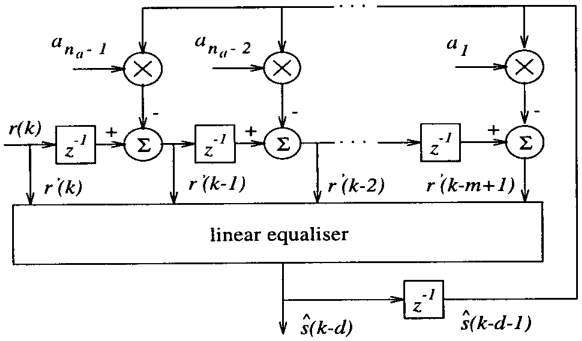


Figure 9: Schematic of the translated DFE.

where the indicator function

$$\delta(\hat{s}(k-d)-s(k-d)) = \begin{cases} 0, & \hat{s}(k-d) = s(k-d) \\ 1, & \hat{s}(k-d) \neq s(k-d) \end{cases} \tag{26}$$

and N is the number of training data. The MSER solution $\mathbf{w}_{\text{MSER}}$ is obtained by minimizing $\hat{P}_E(\mathbf{w})$ using the ASA algorithm.

The proposed Monte Carlo approach was tested using the following example:

$$\left.\begin{array}{l} \text{Channel } \mathbf{a} = [0.3482\ 0.8704\ 0.3482]^T \\ 4-\text{PAM symbols} \end{array}\right\} . \tag{27}$$

For the case of SNR=20 dB and $N = 2000$ with the initial $\mathbf{w}$ set to the $\mathbf{w}_{\text{MMSE}}$, Fig. 10 depicts the evolution of the training SER. The SERs obtained by the MSER DFE and the MMSE DFE with detected symbols being fed back are compared in Fig. 11. The number of training data N for evaluating $\hat{P}_E(\mathbf{w})$ ranged from 100 to 100,000, depending on the SNR. The number of symbols used to estimate the SERs of the trained equalizers, shown in Figs. 11, was sufficient large to produce 400 error counts for each SNR.

As expected, the MSER solution is superior over the MMSE solution. This Monte Carlo algorithm is block-data based, and is particularly suited for data storage systems, as in many commercial disk drives, the equalizers are trained at the factory floor and then are "frozen" before shipping. For communication links, the approach is suitable for the initial set up of the DFE.

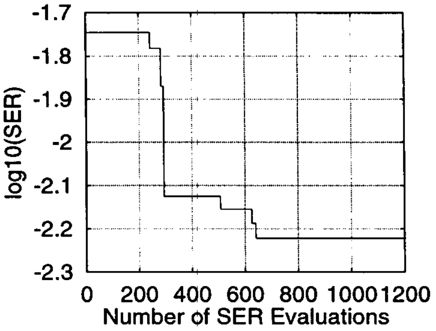


Figure 10: Evolution of the training SER versus number of SER evaluations. SNR=20 dB and the initial **w** is the MMSE solution.

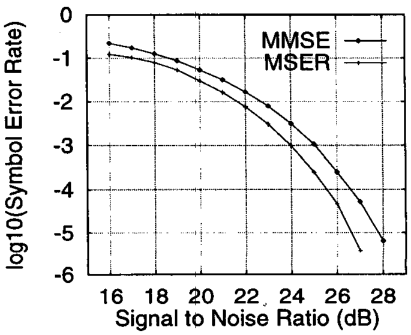


Figure 11: Performance comparison for the MSER and MMSE DSFEs with detected symbols being fed back.

4 Conclusions

The ASA is a global optimization technique having certain advantages. The algorithm is versatile and very easy to program, and has very few parameters that require tuning. In this study, we have applied the ASA to three very different signal processing applications, IIR filter design, ML joint channel and data estimation and evaluation of MSER DFE. Our results shows that, for the first two applications, the efficiency of the ASA appears to be in the same order as the GA, which are available in the literature. This study has demonstrated that the ASA provides a viable alternative to the better known GA for solving diverse signal processing applications with multimodal and nonsmooth cost functions.

References

[1] J.H. Holland, *Adaptation in Natural and Artificial Systems.* Ann Arbor, MI.: University of Michigan Press, 1975.

[2] D.E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning.* Reading, MA.: Addison-Wesley, 1989.

[3] L. Davis, Ed., *Handbook of Genetic Algorithms.* Van Nostrand Reinhold, 1991.

[4] S. Kirkpatrick, C.D. Gelatt Jr. and M.P. Vecchi, "Optimization by simulated annealing," *Science*, Vol.220, No.4598, pp.671–680, 1983.

[5] A. Corana, M. Marchesi, C. Martini and S. Ridella, "Minimizing multimodal functions of continuous variables with the simulated annealing algorithm," *ACM Trans. Mathematical Software*, Vol.13, No.3, pp.262–280, 1987.

[6] P.J.M. van Laarhoven and E.H.L. Aarts, *Simulated Annealing: Theory and Applications.* Dordrecht, Netherlands: D. Reidel, 1987.

[7] K.S. Tang, K.F. Man, S. Kwong and Q. He, "Genetic algorithms and their applications," *IEEE Signal Processing Magazine*, Vol.13, No.6, pp.22–37, 1996.

[8] R. Nambiar, C.K.K. Tang and P. Mars, "Genetic and learning automata algorithms for adaptive digital filters," in *Proc. IEEE ICASSP*, 1992, Vol.IV, pp.41–44.

[9] S. Chen and Y. Wu, "Maximum likelihood joint channel and data estimation using genetic algorithms," *IEEE Trans. Signal Processing*, Vol.46, No.5, pp.1469–1473, 1998.

[10] L. Ingber and B. Rosen, "Genetic algorithms and very fast simulated reannealing: a comparison," *Mathematical and Computer Modelling*, Vol.16, No.11, pp.87–100, 1992.

[11] L. Ingber, "Simulated annealing: practice versus theory," *Mathematical and Computer Modelling*, Vol.18, No.11, pp.29–57, 1993.

[12] L. Ingber, "Adaptive simulated annealing (ASA): lessons learned," *J. Control and Cybernetics*, Vol.25, No.1, pp.33-54, 1996.

[13] B.E. Rosen, "Rotationally parameterized very fast simulated reannealing," submitted to *IEEE Trans. Neural Networks*, 1997.

[14] P.B. Wilson and M.D. Macleod, "Low implementation cost IIR digital filter design using genetic algorithms," in *Workshop on Natural Algorithms in Signal Processing* (Chelmsford, Essex), Nov.14-16, 1993, pp.4/1–4/8.

[15] S.C. Ng, S.H. Leung, C.Y. Chung, A. Luk and W.H. Lau, "The genetic search approach: a new learning algorithm for adaptive IIR filtering," *IEEE Signal Processing Magazine*, Vol.13, No.6, pp.38–46, 1996.

[16] A.H. Gray Jr. and J.D. Markel, "Digital lattice and ladder filter synthesis," *IEEE Trans. Audio and Electroacoustics*, Vol.AU-21, No.6, pp.491–500, 1973.

[17] J.J. Shynk, "Adaptive IIR filtering," *IEEE ASSP Magazine*, April 1989, pp.4–21.

[18] N. Seshadri, "Joint data and channel estimation using blind trellis search techniques," *IEEE Trans. Communications*, Vol.42, No.2/3/4, pp.1000–1011, 1994.

[19] E. Zervas, J. Proakis and V. Eyuboglu, "A quantized channel approach to blind equalization," in *Proc. ICC'92* (Chicago), 1992, Vol.3, pp.351.8.1–351.8.5.

[20] S.U.H. Qureshi, "Adaptive equalization," *Proc. IEEE*, Vol.73, No.9, pp.1349–1387, 1985.

[21] J.G. Proakis, *Digital Communications.* 3rd edition, New York: McGraw-Hill, 1995.

[22] J. Moon, "The role of SP in data-storage systems," *IEEE Signal Processing Magazine*, Vol.15, No.4, pp.54–72, 1998.

[23] J.G. Proakis, "Equalization techniques for high-density magnetic recording," *IEEE Signal Processing Magazine*, Vol.15, No.4, pp.73–82, 1998.

[24] S. Chen, E.S. Chng, B. Mulgrew and G. Gibson, "Minimum-BER linear-combiner DFE," in *Proc. ICC'96* (Dallas, Texas), 1996, Vol.2, pp.1173–1177.

[25] S. Chen, B. Mulgrew, E.S. Chng and G. Gibson, "Space translation properties and the minimum-BER linear-combiner DFE," *IEE Proc. Communications*, Vol.145, No.5, pp.316–322, 1998.

[26] J.M. Cioffi, G.P. Dudevoir, M.V. Eyuboglu and G.D. Forney, "MMSE decision-feedback equalizers and coding – part I: equalization results," *IEEE Trans. Communications*, Vol.43, No.10, pp.2582–2594, 1995.