A Search Algorithm for Global Optimisation

S. Chen[1], X.X. Wang[2], and C.J. Harris[1]

[1] School of Electronics and Computer Science,
University of Southampton, Southampton SO17 1BJ, U.K
[2] Neural Computing Research Group, Aston University,
Birmingham B4 7ET, U.K

Abstract. This paper investigates a global search optimisation technique, referred to as the repeated weighted boosting search. The proposed optimisation algorithm is extremely simple and easy to implement. Heuristic explanation is given for the global search capability of this technique. Comparison is made with the two better known and widely used global search techniques, known as the genetic algorithm and adaptive simulated annealing. The effectiveness of the proposed algorithm as a global optimiser is investigated through several examples.

1 Introduction

Evolutionary and natural computation has always provided inspirations for global search optimisation techniques. Indeed, two of the best-known global optimisation algorithms are the genetic algorithm (GA) [1]-[3] and adaptive simulated annealing (ASA) [4]-[6]. The GA and ASA belong to a class of guided random search methods. The underlying mechanisms for guiding optimisation 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 ASA by contrast evolves a single solution in the parameter space with certain guiding principles that imitate the random behaviour of molecules during the annealing process. It adopts a re-annealing scheme to speed up the search process and to make the optimisation process robust.

We experiment with a guided random search algorithm, which we refer to as the repeated weighted boosting search (RWBS). This algorithm is remarkably simple, requiring a minimum software programming effort and algorithmic tuning, in comparison with the GA or ASA. The basic process evolves a population of initially randomly chosen solutions by performing a convex combination of the potential solutions and replacing the worst member of the population with it until the process converges. The weightings used in the convex combination are adapted to reflect the “goodness” of corresponding potential solutions using the idea from boosting [7]-[9]. The process is repeated a number of “generations” to improve the probability of finding a global optimal solution. An elitist strategy is adopted by retaining the best solution found in the current generation in the initial population of the next generation. Several examples are included to demonstrate the effectiveness of this RWBS algorithm as a global optimisation tool and to compare it with the GA and ASA in terms of convergence speed.

L. Wang, K. Chen, and Y.S. Ong (Eds.): ICNC 2005, LNCS 3611, pp. 1122–1130, 2005.

The generic optimisation problem considered is defined by

$$\min_{\mathbf{u}\in\mathcal{U}} J(\mathbf{u}) \tag{1}$$

where $\mathbf{u} = [u_1 \cdots u_n]^T$ is the n-dimensional parameter vector to be optimised, and $\mathcal{U}$ defines the feasible set. The cost function $J(\mathbf{u})$ can be multimodal and nonsmooth.

2 The Proposed Guided Random Search Method

A simple and effective strategy for forming a global optimiser is called the multistart [10]. A local optimiser is first defined. By repeating the local optimiser multiple times with some random sampling initialisation, a global search algorithm is formed. We adopt this strategy in deriving the RWBS algorithm.

2.1 Weighted Boosting Search as a Local Optimiser

Consider a population of P_S points, $\mathbf{u}_i \in \mathcal{U}$ for $1 \le i \le P_S$. Let $\mathbf{u}_{\text{best}} = \arg\min J(\mathbf{u})$ and $\mathbf{u}_{\text{worst}} = \arg\max J(\mathbf{u})$, where $\mathbf{u} \in \{\mathbf{u}_i,\ 1 \le i \le P_S\}$. Now a (P_S+1)th point is generated by performing a convex combination of $\mathbf{u}_i$, $1 \le i \le P_S$, as

$$\mathbf{u}_{P_S+1} = \sum_{i=1}^{P_S} \delta_i \mathbf{u}_i \tag{2}$$

where the weightings satisfy $\delta_i \ge 0$ and $\sum_{i=1}^{P_S} \delta_i = 1$. The point $\mathbf{u}_{P_S+1}$ is always within the convex hull defined by $\mathbf{u}_i$, $1 \le i \le P_S$. A mirror image of $\mathbf{u}_{P_S+1}$ is then generated with respect to $\mathbf{u}_{\text{best}}$ and along the direction defined by $\mathbf{u}_{\text{best}} - \mathbf{u}_{P_S+1}$ as

$$\mathbf{u}_{P_S+2} = \mathbf{u}_{\text{best}} + (\mathbf{u}_{\text{best}} - \mathbf{u}_{P_S+1}) \tag{3}$$

According to their cost function values, the best of $\mathbf{u}_{P_S+1}$ and $\mathbf{u}_{P_S+2}$ then replaces $\mathbf{u}_{\text{worst}}$. The process is iterated until the population converges. The convergence is assumed if $\|\mathbf{u}_{P_S+1} - \mathbf{u}_{P_S+2}\| < \xi_B$, where the small $\xi_B > 0$ defines search accuracy.

The weightings δ_i, $1 \le i \le P_S$, should reflect the "goodness" of $\mathbf{u}_i$, and the process should be capable of self-learning these weightings. We modify the AdaBoost algorithm [8] to adapt the weightings δ_i, $1 \le i \le P_S$. Let t denote the iteration index, and give the initial weightings $\delta_i(0) = \frac{1}{P_S}$, $1 \le i \le P_S$. Further denote $J_i = J(\mathbf{u}_i)$ and $\bar{J}_i = J_i / \sum_{j=1}^{P_S} J_j$, $1 \le i \le P_S$. Then the weightings are updated according to

$$\tilde{\delta}_i(t) = \begin{cases} \delta_i(t-1)\beta_t^{\bar{J}_i}, & \text{for } \beta_t \le 1 \\ \delta_i(t-1)\beta_t^{1-\bar{J}_i}, & \text{for } \beta_t > 1 \end{cases} \tag{4}$$

$$\delta_i(t) = \frac{\tilde{\delta}_i(t)}{\sum_{j=1}^{P_S} \tilde{\delta}_j(t)},\ 1 \le i \le P_S \tag{5}$$

where

$$\beta_t = \frac{\eta_t}{1-\eta_t},\ \eta_t = \sum_{i=1}^{P_S} \delta_i(t-1)\bar{J}_i \tag{6}$$

The weighted boosting search (WBS) is a local optimiser that finds an optimal solution within the convex region defined by the initial population. This capability can be explained heuristically using the theory of weak learnability [7],[8]. The members of the population $\mathbf{u}_i$, $1 \le i \le P_S$, can be seen to be produced by a "weak learner", as they are generated "cheaply" and do not guarantee certain optimal property. Schapire [7] showed that any weak learning procedure can be efficiently transformed (boosted) into a strong learning procedure with certain optimal property. In our case, this optimal property is the ability of finding an optimal point within the defined search region.

2.2 Repeated Weighted Boosting Search as a Global Optimiser

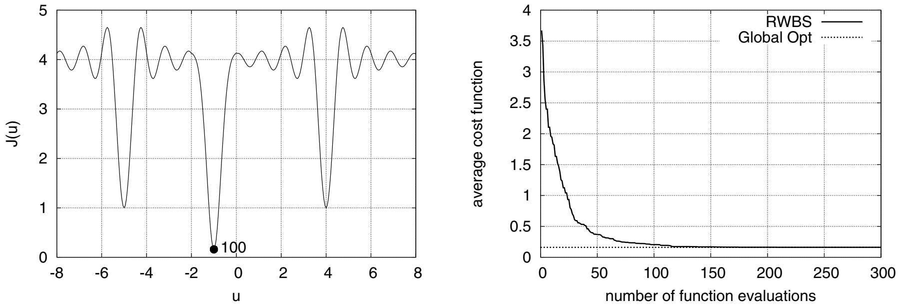
The WBS is a local optimiser, as the solution obtained depends on the initial choice of population. We "convert" it to a global search algorithm by repeating it N_G times or "generations" with a random sampling initialization equipping with an elitist mechanism. The resulting global optimiser, the RWBS algorithm, is summarised as follows.

- **Loop: generations** For $g = 1 : N_G$
- – Initialise the population by setting $\mathbf{u}_1^{(g)} = \mathbf{u}_{\text{best}}^{(g-1)}$ and randomly generating rest of the population members $\mathbf{u}_i^{(g)}$, $2 \le i \le P_S$, where $\mathbf{u}_{\text{best}}^{(g-1)}$ denotes the solution found in the previous generation. If $g = 1$, $\mathbf{u}_1^{(g)}$ is also randomly chosen
- – Call the WBS to find a solution $\mathbf{u}_{\text{best}}^{(g)}$
- **End of generation loop**

The appropriate values for P_S, N_G and ξ_B depends on the dimension of $\mathbf{u}$ and how hard the objective function to be optimised. Generally, these algorithmic parameters have to be found empirically, just as in any other global search algorithm. The elitist initialisation is useful, as it keeps the information obtained by the previous search generation, which otherwise would be lost due to the randomly sampling initialisation. Note that for the iterative procedure of the WBS, there is no need for every members of the population to converge to a (local) minimum, and it is sufficient to locate where the minimum lies. Thus, ξ_B can be set to a relatively large value. This makes the search efficient, achieving convergence with a small number of the cost function evaluations. It should be obvious, although the formal proof is still required, that with sufficient number of generations, the algorithm will guarantee to find a global optimal solution, since the parameter space will be searched sufficiently. In a variety of optimisation applications, we have found that the RWBS is efficient in finding global optimal solutions and achieve a similar convergence speed as the GA and ASA, in terms of the required total number of the cost function evaluations. The RWBS algorithm has additional advantage of being very simple, needing a minimum programming effort and having few algorithmic parameters that require tuning, in comparison with the GA and ASA.

3 Optimisation Applications

Example 1. The cost function to be optimised is depicted in Fig. 1 (a). Uniformly random sampling in $[-8,\ 8]$ was adopted for population initialisation. With $P_S = 4$,



(a) (b)

Fig. 1. One-dimensional multimodal function minimisation using the RWBS: (a) cost function, where number 100 beside the point in the graph indicates convergence to the global minimum in all the 100 experiments, and (b) convergence performance averaged over 100 experiments.

$\xi_B = 0.02$ as well as $N_G > 6$, the RWBS algorithm consistently converged to the global minimum point at $u = -1$ in all the 100 experiments conducted, as can be seen from the convergence performance shown in Fig. 1 (b). The averaged number of cost function evaluations required for the algorithm to converge to the global optimal solution is around 100, which is consistent with what can be achieved using GA and ASA for this type of one-dimensional optimisation.

Example 2. The IIR filter with transfer function $H_M(z)$ was used to identify the system with transfer function $H_S(z)$ by minimising the mean square error (MSE) $J(\mathbf{u})$, where

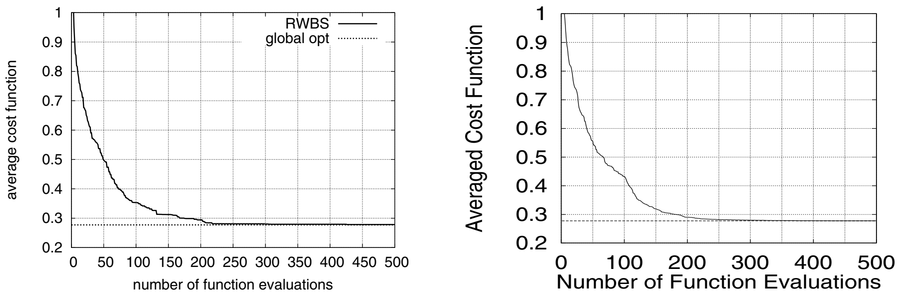
$$H_S(z) = \frac{0.05 - 0.4z^{-1}}{1 - 1.1314z^{-1} + 0.25z^{-2}}, \; H_M(z) = \frac{a_0}{1 + b_1 z^{-1}} \tag{7}$$

and $\mathbf{u} = [a_0 \; b_1]^T$. When the system input is white and the noise is absent, the MSE cost function has a global minimum at $\mathbf{u}_{\text{global}} = [-0.311 \; -0.906]^T$ with the value of the normalised MSE 0.2772 and a local minimum at $\mathbf{u}_{\text{local}} = [0.114 \; 0.519]^T$ with the normalised MSE value 0.9762 [11]. In the population initialisation, the parameters were uniformly randomly chosen as $(a_0, \; b_1) \in (-1.0, \; 1.0) \times (-0.999, \; 0.999)$. It was found empirically that $P_S = 4$, $\xi_B = 0.05$ $N_G > 15$ were appropriate, and Fig. 2 (a) depicts convergence performance of the RWBS algorithm averaged over 100 experiments. The previous study [6] applied the ASA to this example. The result of using the ASA is reproduced in Fig. 2 (b) for comparison. The distribution of the solutions obtained in 100 experiments by the RWBS algorithm is shown in Fig. 3.

Example 3. For this 2nd-order IIR filter design, the system and filter transfer functions are given by

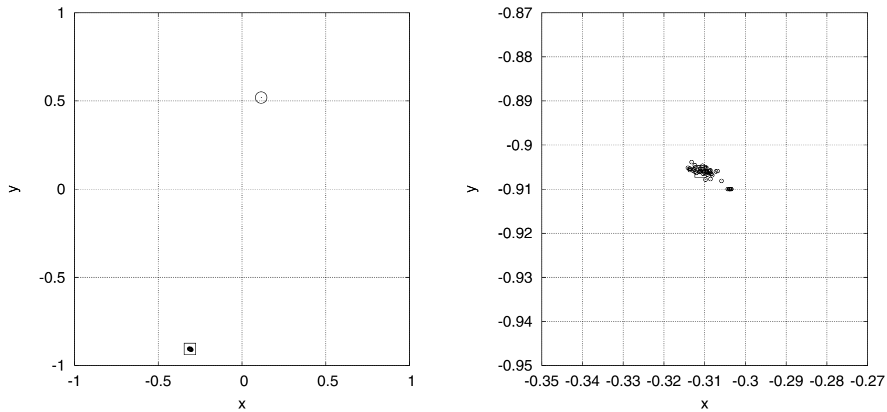
$$H_S(z) = \frac{-0.3 + 0.4z^{-1} - 0.5z^{-2}}{1 - 1.2z^{-1} + 0.5z^{-2} - 0.1z^{-3}}, \; H_M(z) = \frac{a_0 + a_1 z^{-1}}{1 + b_1 z^{-1} + b_2 z^{-2}} \tag{8}$$

respectively. In the simulation, the system input was a uniformly distributed white sequence, taking values from $(-1, \; 1)$, and the signal to noise ratio was SNR=30 dB.



(a) (b)

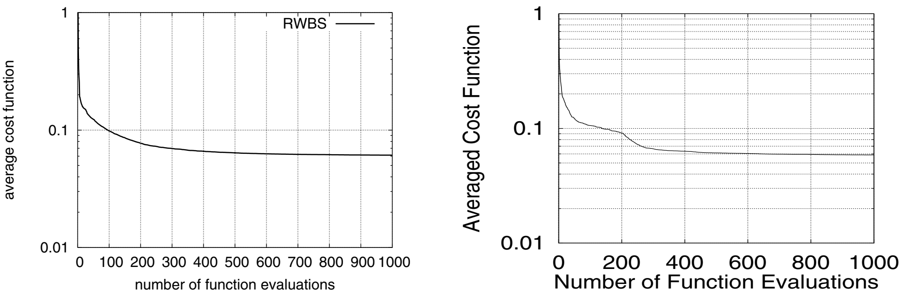
Fig. 2. Convergence performance averaged over 100 experiments for the 1st-order IIR filter design: (a) using the RWBS, and (b) using the ASA.



(a) (b)

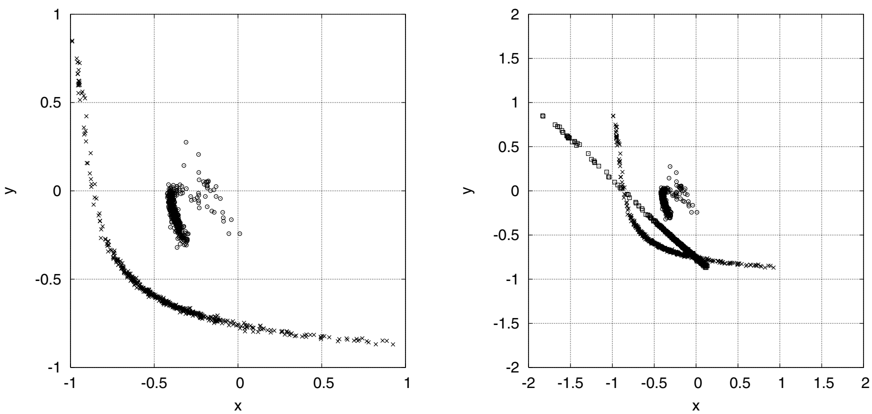
Fig. 3. Distribution of solutions (a_0, b_1) (small circles) obtained in 100 experiments for the 1st-order IIR filter design by the RWBS: (a) showing the entire search space, and (b) zooming in the global minimum, where large square indicate the global minimum and large circle the local minimum.

The data length used in calculating the MSE cost function was 2000. The MSE for this example was multi-modal and the gradient-based algorithm performed poorly as was demonstrated clearly in [6]. In the actual optimisation, the lattice form of the IIR filter was used, and the filter coefficient vector used in optimisation was $\mathbf{u} = [a_0 \; a_1 \; \kappa_0 \; \kappa_1]^T$, where κ_0 and κ_1 are the lattice-form reflection coefficients. In the population initialisation, the parameters were uniformly randomly chosen as $a_i \in (-1.0, \; 1.0)$ and $\kappa_i \in (-0.999, \; 0.999)$ for $i = 0, 1$. It was found out that $N_B = 10$, $\xi_B = 0.05$ and $N_G > 20$ were appropriate for the RWBS algorithm, and Fig. 4 (a) depicts convergence performance of the RWBS algorithm averaged over 500 experiments. In [6], convergence performance using the ASA was obtained by averaging over 100 experiments, and this result is also re-plotted in Fig. 4 (b) as a comparison. The distribution



(a) (b)

Fig. 4. Convergence performance for the 2nd-order IIR filter design: (a) using the RWBS averaged over 500 experiments, and (b) using the ASA averaged over 100 experiments.



(a) (b)

Fig. 5. Distribution of the solutions obtained in 500 experiments for the 2nd-order IIR filter design by the RWBS: (a) showing (a_0, a_1) as circles and (κ_0, κ_1) as crosses, and (b) showing (a_0, a_1) as circles, (b_1, b_2) as squares, and (κ_0, κ_1) as crosses.

of the solutions obtained in 500 experiments by the RWBS is illustrated in Fig. 5. It is clear that for this example there are infinitely many global minima, and the global minimum solutions for (b_1, b_2) form a one-dimensional space.

Example 4. Consider a blind joint maximum likelihood (ML) channel estimation and data detection for the single-input multiple-output (SIMO) system that employs a single transmitter antenna and L (> 1) receiver antennas. In a SIMO system, the symbol-rate sampled antennas' outputs $x_l(k)$, $1 \leq l \leq L$, are given by

$$x_l(k) = \sum_{i=0}^{n_c - 1} c_{i,l} s(k - i) + n_l(k) \tag{9}$$

where $n_l(k)$ is the complex-valued Gaussian white noise associated with the lth channel and $E[|n_l(k)|^2] = 2\sigma_n^2$, $\{s(k)\}$ is the transmitted symbol sequence taking values from the quadrature phase shift keying (QPSK) symbol set $\{\pm 1 \pm j\}$, and $c_{i,l}$ are the channel impulse response (CIR) taps associated with the lth receive antenna. Let

$$\begin{aligned} \mathbf{x} &= [x_1(1)\, x_1(2) \cdots x_1(N)\, x_2(1) \cdots x_L(1)\, x_L(2) \cdots x_L(N)]^T \\ \mathbf{s} &= [s(-n_c+2) \cdots s(0)\, s(1) \cdots s(N)]^T \\ \mathbf{c} &= [c_{0,1}\, c_{1,1} \cdots c_{n_c-1,1}\, c_{0,2} \cdots c_{0,L}\, c_{1,L} \cdots c_{n_c-1,L}]^T \end{aligned} \quad (10)$$

be the vector of $N \times L$ received signal samples, the corresponding transmitted data sequence and the vector of the SIMO CIRs, respectively. The probability density function of the received data vector $\mathbf{x}$ conditioned on $\mathbf{c}$ and $\mathbf{s}$ is

$$p(\mathbf{x}|\mathbf{c},\mathbf{s}) = \frac{1}{(2\pi\sigma_n^2)^{NL}}\, e^{-\frac{1}{2\sigma_n^2}\sum_{k=1}^{N}\sum_{l=1}^{L}\left|x_l(k)-\sum_{i=0}^{n_c-1} c_{i,l}s(k-i)\right|^2} \quad (11)$$

The joint ML estimate of $\mathbf{c}$ and $\mathbf{s}$ is obtained by maximising $p(\mathbf{x}|\mathbf{c},\mathbf{s})$ over $\mathbf{c}$ and $\mathbf{s}$ jointly. Equivalently, the joint ML estimate is the minimum of the cost function

$$J_{\mathrm{ML}}(\hat{\mathbf{c}},\hat{\mathbf{s}}) = \frac{1}{N}\sum_{k=1}^{N}\sum_{l=1}^{L}\left|x_l(k) - \sum_{i=0}^{n_c-1}\hat{c}_{i,l}\hat{s}(k-i)\right|^2 \quad (12)$$

The joint minimisation process $(\hat{\mathbf{c}}^*,\hat{\mathbf{s}}^*) = \arg\left[\min_{\hat{\mathbf{c}},\hat{\mathbf{s}}} J_{\mathrm{ML}}(\hat{\mathbf{c}},\hat{\mathbf{s}})\right]$ can be solved iteratively first over the data sequences $\hat{\mathbf{s}}$ and then over all the possible channels $\hat{\mathbf{c}}$:

$$(\hat{\mathbf{c}}^*,\hat{\mathbf{s}}^*) = \arg\left[\min_{\hat{\mathbf{c}}}\left(\min_{\hat{\mathbf{s}}} J_{\mathrm{ML}}(\hat{\mathbf{c}},\hat{\mathbf{s}})\right)\right] \quad (13)$$

The inner optimisation can readily be carried out using the Viterbi algorithm (VA). We employ the RWBS algorithm to perform the outer optimisation task, and the proposed blind joint ML optimisation scheme can be summarised as follows.

Outer level Optimisation. The RWBS searches the SIMO channel parameter space to find a global optimal estimate $\hat{\mathbf{c}}^*$ by minimising the MSE $J_{\mathrm{MSE}}(\hat{\mathbf{c}}) = J_{\mathrm{ML}}(\hat{\mathbf{c}},\tilde{\mathbf{s}}^*)$.

Inner level optimisation. Given the channel estimate $\hat{\mathbf{c}}$, the VA provides the ML decoded data sequence $\tilde{\mathbf{s}}^*$, and feeds back the corresponding value of the likelihood metric $J_{\mathrm{ML}}(\hat{\mathbf{c}},\tilde{\mathbf{s}}^*)$ to the upper level.

The SIMO CIRs, listed in Table 1, were simulated with the data length $N = 50$. In practice, the value of $J_{\mathrm{MSE}}(\hat{\mathbf{c}})$ is all that the upper level optimiser can see, and the convergence of the algorithm can only be observed through this MSE. In simulation, the performance of the algorithm can also be assessed by the mean tap error defined as

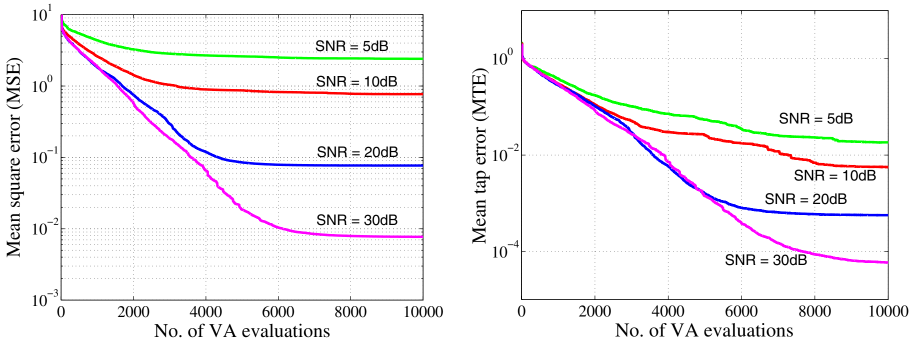
$$\mathrm{MTE} = \|\mathbf{c} - a\cdot\hat{\mathbf{c}}\|^2 \quad (14)$$

where

$$a = \begin{cases} \pm 1, & \text{if } \hat{\mathbf{c}} \to \pm\mathbf{c} \\ \mp j, & \text{if } \hat{\mathbf{c}} \to \pm j\mathbf{c} \end{cases} \quad (15)$$

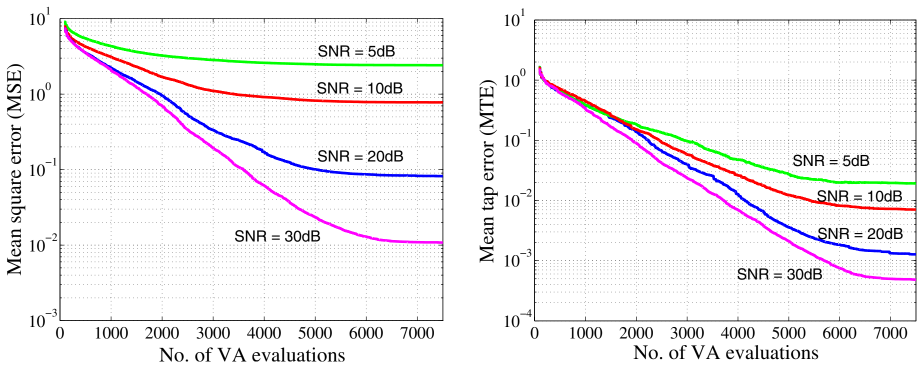
Table 1. The simulated SIMO system

l	Channel impulse response		
1	0.365-0.274j	0.730+0.183j	-0.440+0.176j
2	0.278+0.238j	-0.636+0.104j	0.667-0.074j
3	-0.639+0.249j	-0.517-0.308j	0.365+0.183j
4	-0.154+0.693j	-0.539-0.077j	0.268-0.358j



(a) (b)

Fig. 6. Convergence performance of blind joint ML estimation using the RWBS averaged over 50 runs: (a) MSE and (b) MTE against number of VA evaluations.



(a) (b)

Fig. 7. Convergence performance of blind joint ML estimation using the GA averaged over 50 runs: (a) MSE and (b) MTE against number of VA evaluations.

Note that since $(\hat{\mathbf{c}}^*, \hat{\mathbf{s}}^*)$, $(-\hat{\mathbf{c}}^*, -\hat{\mathbf{s}}^*)$, $(-j\hat{\mathbf{c}}^*, +j\hat{\mathbf{s}}^*)$ and $(+j\hat{\mathbf{c}}^*, -j\hat{\mathbf{s}}^*)$ are all the solutions of the joint ML estimation problem, the channel estimate $\hat{\mathbf{c}}$ can converges to $\mathbf{c}$, $-\mathbf{c}$, $j\mathbf{c}$ or $-j\mathbf{c}$. Fig. 6 shows the evolutions of the MSE and MTE averaged over 50 runs and for different values of signal to noise ratio (SNR), obtained by the blind joint ML optimisation scheme using the RWBS. From Fig. 6, it can be seen that the MSE converged to the noise floor. We also investigated using the GA to perform the upper-level optimisation, and the results obtained by this GA-based blind joint ML estimation scheme are presented in Fig. 7. It is worth pointing out that the dimension of the search space was $n = 24$ for this example.

4 Conclusions

A guided random search optimisation algorithm has been proposed. The local optimiser in this global search method evolves a population of the potential solutions by forming a convex combination of the solution population with boosting adaptation. A repeating loop involving a combined elitist and random sampling initialisation strategy is adopted to guarantee a fast global convergence. The proposed guided random search method, referred to as the RWBS, is remarkably simple, involving minimum software programming effort and having very few algorithmic parameters that require tuning. The versatility of the proposed method has been demonstrated using several examples, and the results obtained show that the proposed global search algorithm is as efficient as the GA and ASA in terms of global convergence speed, characterised by the total number of cost function evaluations required to attend a global optimal solution.

Acknowledgement

S. Chen wish to thank the support of the United Kingdom Royal Academy of Engineering.

References

1. J.H. Holland, *Adaptation in Natural and Artificial Systems*. University of Michigan Press: Ann Arbor, MI, 1975.
2. D.E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison Wesley: Reading, MA, 1989.
3. L. Davis, Ed., *Handbook of Genetic Algorithms*. Van Nostrand Reinhold: New York, 1991.
4. 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.
5. 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.
6. S. Chen and B.L. Luk, "Adaptive simulated annealing for optimization in signal processing applications," *Signal Processing*, Vol.79, No.1, pp.117–128, 1999.
7. R.E. Schapire, "The strength of weak learnability," *Machine Learning*, Vol.5, No.2, pp.197–227, 1990.
8. Y. Freund and R.E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *J. Computer and System Sciences*, Vol.55, No.1, pp.119–139, 1997.
9. R. Meir and G. Rätsch, "An introduction to boosting and leveraging," in: S. Mendelson and A. Smola, eds., *Advanced Lectures in Machine Learning*. Springer Verlag, 2003, pp.119–184.
10. F. Schoen, "Stochastic techniques for global optimization: a survey of recent advances," *J. Global Optimization*, Vol.1, pp.207–228, 1991.
11. J.J. Shynk, "Adaptive IIR filtering," *IEEE ASSP Magazine*, pp.4–21, April 1989.