A Search Algorithm for Global Optimisation

Sheng Chen†, Xunxian Wang‡ and Chris J. Harris†

† School of Electronics and Computer Science
University of Southampton, Southampton SO17 1BJ, U.K.
E-mails: {sqc,cjh}@ecs.soton.ac.uk

‡ Neural Computing Research Group
Aston University, Birmingham B4 7ET, U.K.
E-mail: x.wang@aston.ac.uk

Presented at 2005 International Conference on Natural Computation
August 27-29, 2005, Changsha, China

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

Motivations

Optimisation methods capable of achieving global optimal solution are effective tools for solving variety of machine learning and engineering problems

◯ Genetic algorithms and adaptive simulated annealing generally perform well in very different problems and have similarly good convergence speeds

GA is population based and evolves solution population according to principles of the evolution of species in nature

ASA evolves a single solution in parameter space by imitating random behaviour of molecules during the annealing process

◯ It is highly desirable to have alternative global search algorithm

Simpler in design, programming effort and tuning - Yet has similar convergence performance as GA and ASA

Basic Strategy

○ Multistart

- **Local optimiser**: Finds a (local) optimal solution with given initial point
- **Repeat**: Re-start local optimiser with some given sampling strategy

○ Proposed global search algorithm: **Repeated Weighted Boosting Search**

- Evolve solution population by performing a convex combination of potential solutions and replacing worst member with it until process converges

  Weightings in convex combination are adapted by "boosting" to reflect "goodness" of corresponding potential solutions

- The process is repeated a number of generations

  Elitist sampling strategy retains best solution found in population initialisation

Repeated Weighted Boosting Search

Consider task of minimising $J(\mathbf{u})$

Outer Loop: N_G number of generations

Initialisation: keep best solution found in previous generation as $\mathbf{u}_1$ and randomly choose rest of population $\mathbf{u}_2, \cdots, \mathbf{u}_{P_S}$

Inner Loop: N_I iterations

- Perform a convex combination

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

- Weightings

$$\delta_i \geq 0 \quad \text{and} \quad \sum_{i=1}^{P_S} \delta_i = 1$$

  are adopted (boosting) to reflect goodness of $\mathbf{u}_i$

- $\mathbf{u}_{P_S+1}$ or its mirror image $\mathbf{u}_{P_S+2}$ replaces worst member in population $\mathbf{u}_i$, $1 \leq i \leq P_S$

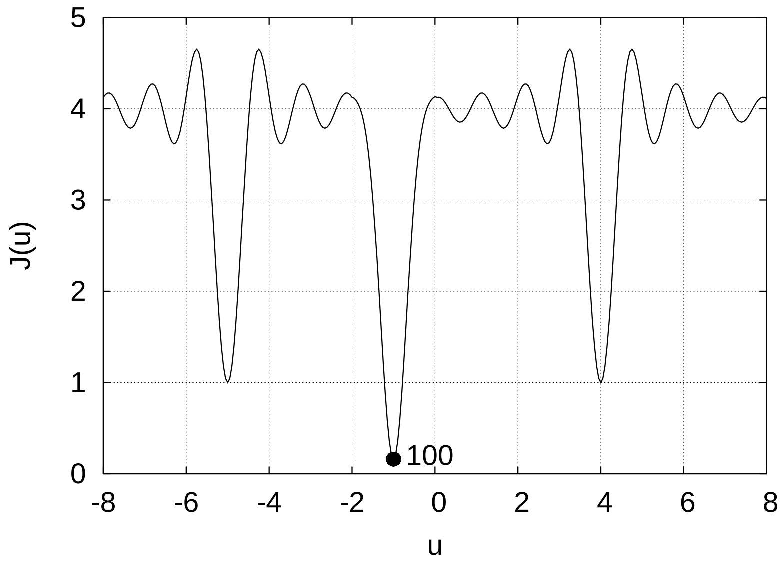
End of *Inner Loop*

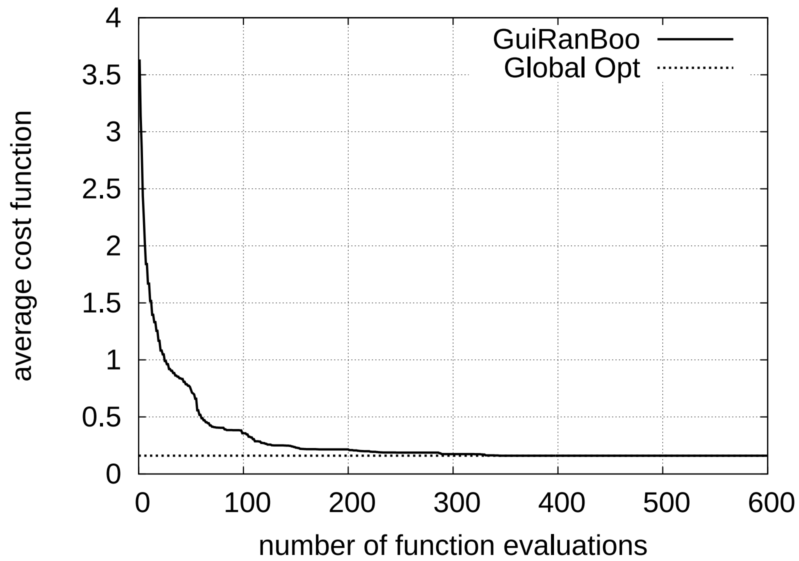
End of *Outer Loop*

One-Dimensional Optimisation Example

◯ Population size $P_S = 6$, number of Inner iterations $N_I = 20$ and number of generations $N_G = 12$

◯ 100 random experiments, populations of all 100 runs converge to global minimum



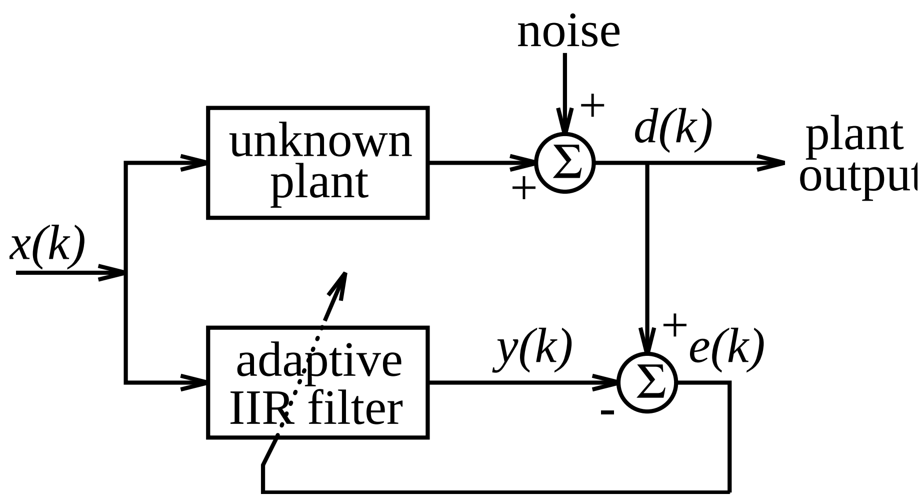


IIR Filter Design

- IIR filter with transfer function

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

used to model unknown plant with system transfer function $H_S(z)$



- Design filter coefficient vector $\mathbf{u} = [a_0 \ a_1 \cdots a_L \ b_1 \cdots b_M]^T$ by minimising

$$J(\mathbf{u}) = \mathrm{E}[e^2(k)] = \mathrm{E}[(d(k) - y(k))^2]$$

$d(k)$: desired response, $y(k)$: filter's output, $e(k) = d(k) - y(k)$: error signal

- Time-averaging cost function $J_N(\mathbf{u})$ is used in practice

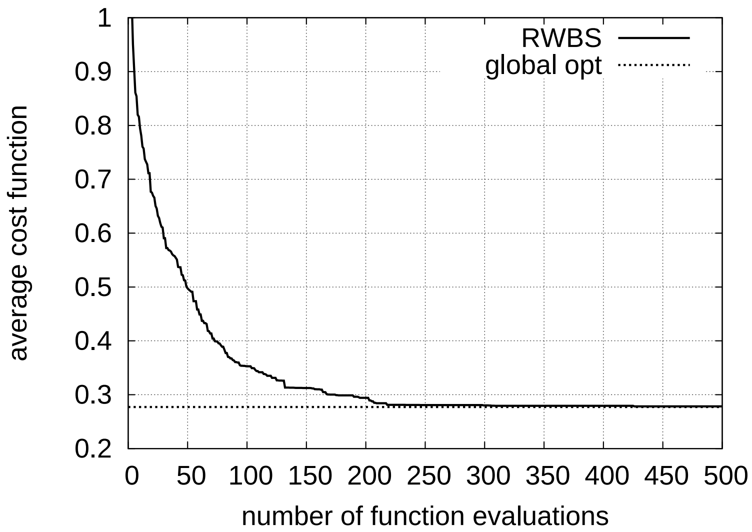
$$J_N(\mathbf{u}) = \frac{1}{N} \sum_{k=1}^{N} (d(k) - y(k))^2$$

IIR Filter Design - Example One

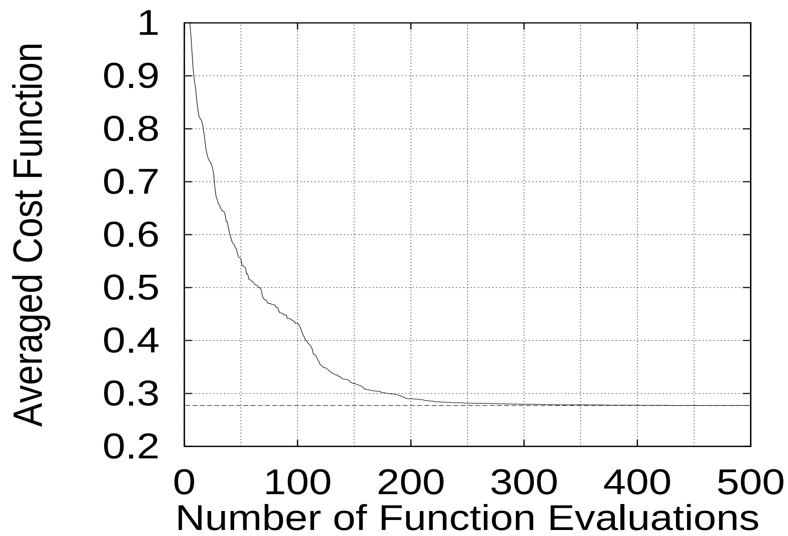
- System and IIR filter transfer functions are respectively

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

- Convergence performance averaged over 100 experiments: (a) RWBS, and (b) ASA

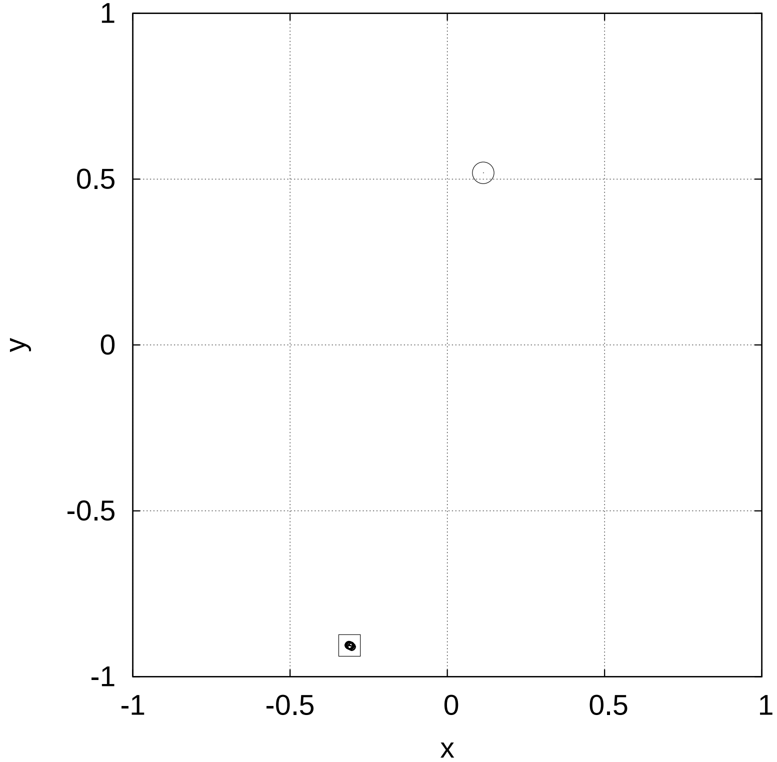


(a)

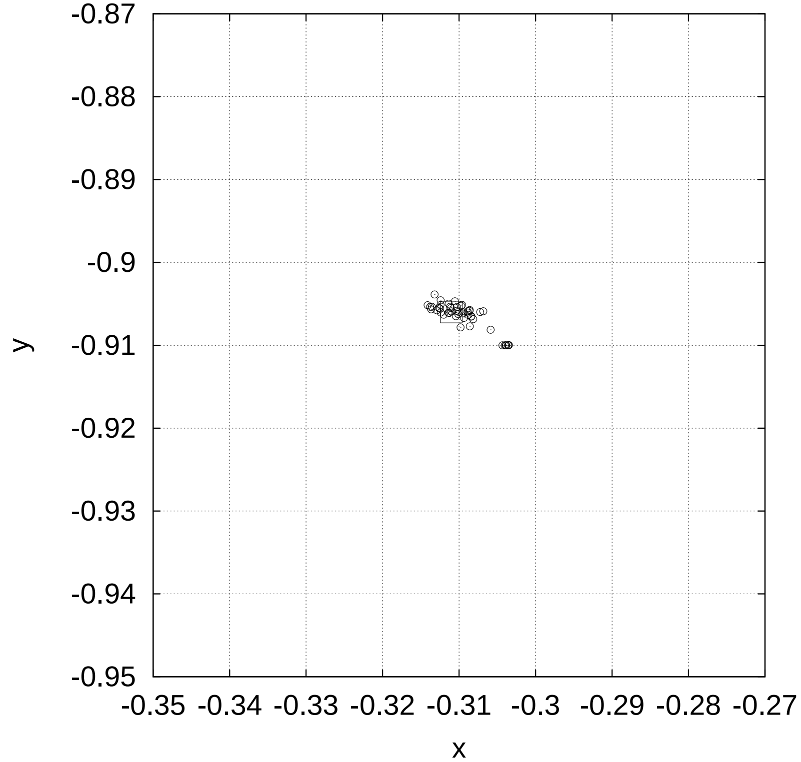


(b)

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



(a)



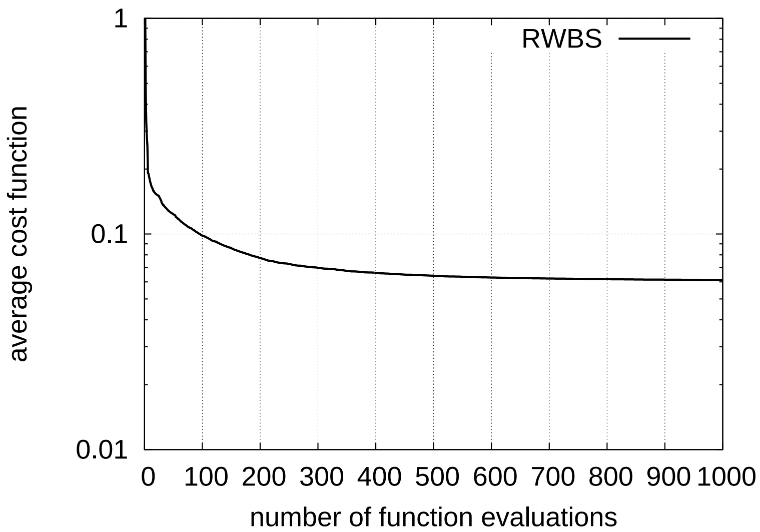
(b)

IIR Filter Design - Example Two

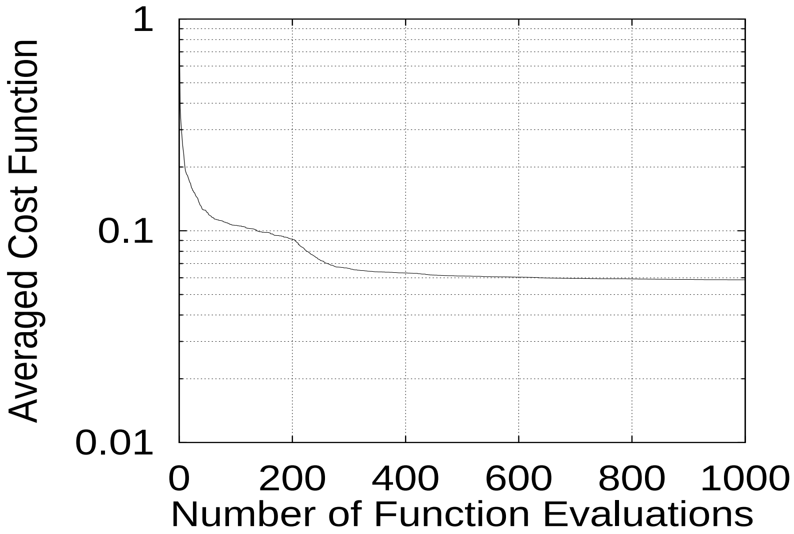
◯ System and IIR filter transfer functions are respectively

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

◯ Convergence performance: (a) RWBS averaged over 500 runs, and (b) ASA averaged over 100 runs

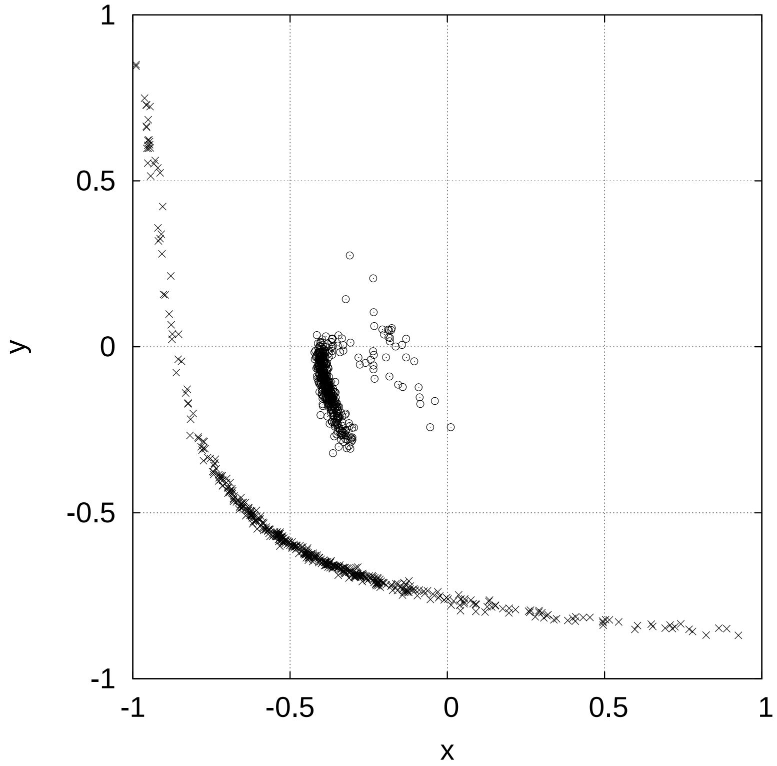


(a)

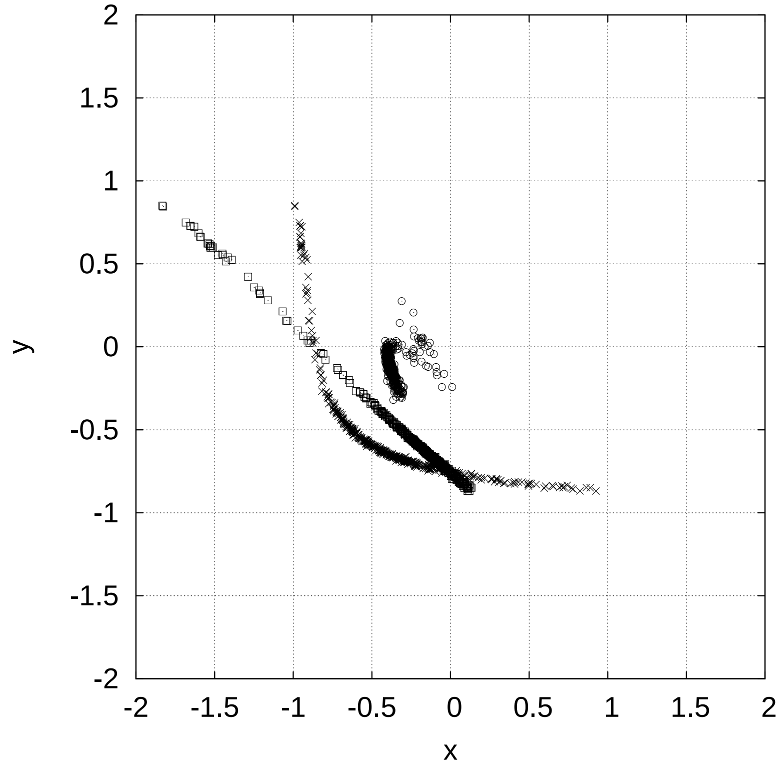


(b)

◯ Distribution of solutions obtained with RWBS in 500 runs for IIR filter design Example 2: (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, κ_i being reflection coefficients



(a)



(b)

Single-Input Multiple-Output Communication System

◯ For SIMO system employing L receiver antennas: antennas' output samples are given by

$$x_l(k) = \sum_{i=0}^{n_C-1} c_{i,l} s(k-i) + n_l(k), \ 1 \le l \le L$$

$n_l(k)$: complex-valued Gaussian white noise, $s(k) \in \{\pm 1 \pm j\}$: transmitted QPSK symbols, $c_{i,l}$: complex-valued CIR taps associated with lth receive antenna, n_c: channel length

◯ Define the vector of $N \times L$ received signal samples, the corresponding transmitted data sequence and the vector of the SIMO CIRs, respectively,

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

Knowing $\mathbf{x}$ and $\mathbf{s}$: channel estimation (LS solution), Knowing $\mathbf{x}$ and $\mathbf{c}$: data detection (Viterbi algorithm), Knowing $\mathbf{x}$ only: blind joint channel estimation and data detection

Joint ML Channel Estimation and Data Detection

◯ PDF of $\mathbf{x}$ conditioned on $\mathbf{c}$ and $\mathbf{s}$ is

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

◯ Joint ML estimate of $\mathbf{c}$ and $\mathbf{s}$ is solution that maximises $p(\mathbf{x}|\mathbf{c},\mathbf{s})$ over $\mathbf{c}$ and $\mathbf{s}$ jointly or, equivalently, it is minimum of cost function

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

namely

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

◯ Joint minimisation can also be solved using an iterative loop first over data sequences $\hat{\mathbf{s}}$ and then over all the possible channels $\hat{\mathbf{c}}$

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

Iterative Loop for Blind Joint ML Solution

Inner optimisation can readily be carried out using standard Viterbi algorithm

Outer optimisation should be capable of finding a global optimal channel estimate efficiently and we employ RWBS algorithm

- *Outer Optimisation*. RWBS searches SIMO channel parameter space to find a global optimal estimate $\hat{\mathbf{c}}^*$ by minimising mean square error (MSE)

$$J_{\rm MSE}(\hat{\mathbf{c}}) = J_{\rm ML}(\hat{\mathbf{c}}, \tilde{\mathbf{s}}^*)$$

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

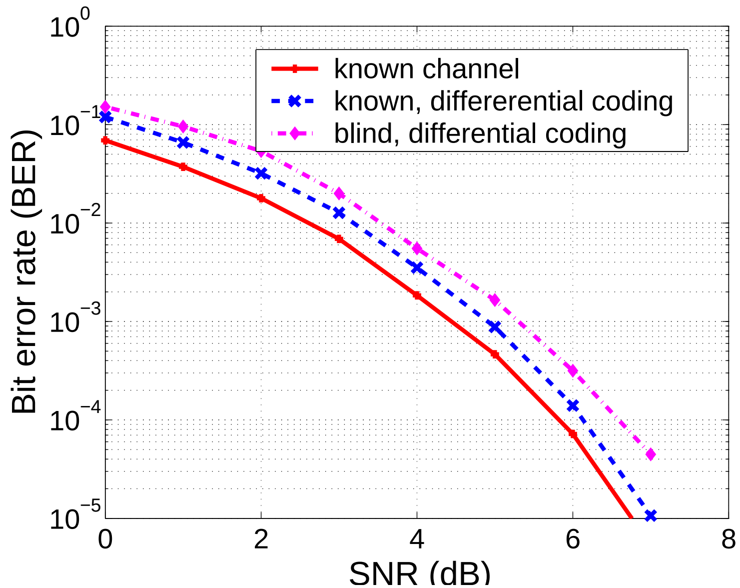
Other global optimisation search algorithms, such as GA and ASA, can also be employed for outer optimisation task

Blind Joint ML SIMO Example

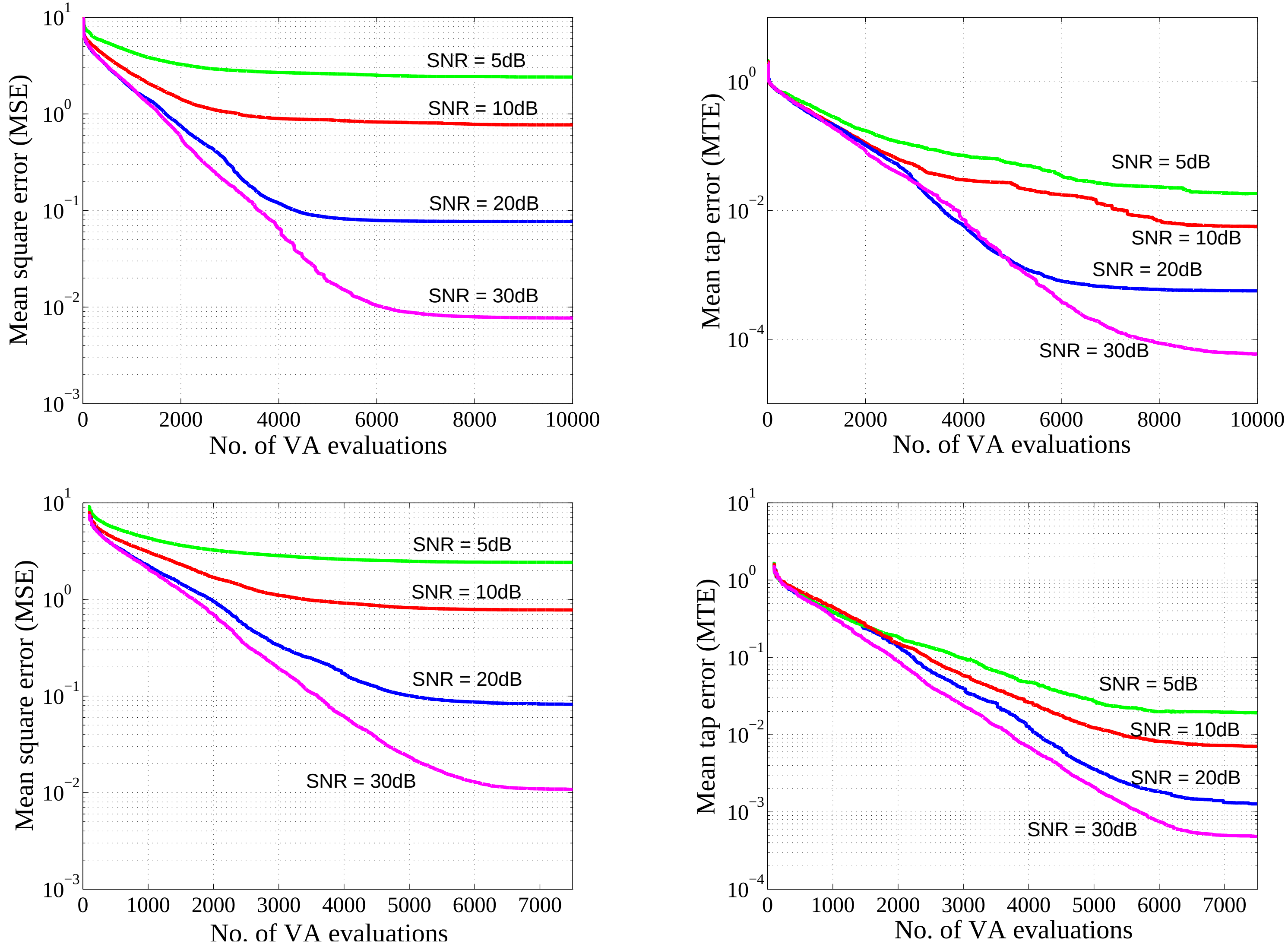
◯ Simulated SIMO

Channel	Channel impulse response		
1	0.3652-0.2739j	0.7304+0.1825j	-0.4402+0.1761j
2	0.2783+0.2376j	-0.6362+0.1039j	0.6671-0.0741j
3	-0.6393+0.2494j	-0.5169-0.3084j	0.3651+0.1826j
4	-0.1539+0.6928j	-0.5389-0.0770j	0.2683-0.3578j

◯ Bit error rate performance using maximum likelihood sequence detection for SIMO channel listed above. Length of data samples for blind scheme is $N = 50$



○ Convergence performance using RWBS and GA averaged over 50 runs



Conclusions

- A guided random search optimisation algorithm has been proposed
  - Local optimiser evolves a population of potential solutions by forming a convex combination of solution population with boosting adaptation
  - Repeating loop involving a combined elitist and random sampling initialisation strategy ensures fast global convergence
- Proposed guided random search method, referred to as RWBS, is remarkably simple, involving minimum software programming effort and having very few algorithmic parameters that require tuning
- Versatility of proposed method has been demonstrated using several examples
  - It is as efficient as GA and ASA in terms of total number of cost function evaluations required to attend a global optimal solution