# On Random Minimization of Functions

**G. P. Tecchiolli**[1], **R. Brunelli**[1]
[1]Istituto per la Ricerca Scientifica e Tecnologica
I38050 Povo, Trento, ITALY

*Abstract*— **A non-deterministic minimization algorithm recently proposed is analyzed. Some characteristics are analytically derived from the analysis of positive definite quadratic forms. An improvement is proposed and compared with the basic algorithm. Different variants of the basic algorithm are finally compared to a standard Conjugate Gradient minimization algorithm in the computation of the Rayleigh coefficient of a positive definite symmetric matrix.**

## 1. Introduction

Function minimization is a widespread need in the scientific community. The major shortcoming of the most used minimization algorithms is the sensitivity to local minima. Deterministic methods, which are guaranteed to get the global minima (like those based on interval analysis, see [7]), have a complexity which is exponential in the dimension of the domain, making them often unusable. Methods like *simulated annealing* promise a reduced sensitivity (see [2], [3]) to local minima but their success depends on the choice of an appropriate annealing schedule. The adaptive non-deterministic algorithm recently proposed by Caprile and Girosi([5]) couples good immunity to local minima to simplicity.

The first section of the paper gives a brief review of the Caprile-Girosi algorithm. The mathematical analysis of algorithm performance for quadratic forms is then discussed. The next section introduces an improvement and the corresponding analysis. Finally, several variants of the basic algorithm are compared in the computation of the Rayleigh coefficient of symmetric positive definite quadratic forms.

## 2. The Caprile-Girosi algorithm

Let us briefly review the non-deterministic algorithm recently proposed by Girosi and Caprile ([5]). The algorithm is characterized by a random search whose scope is limited by an adaptive hyper-ellipsoid. The domain of the function is decomposed as the cartesian product of $n$ linear subspaces and the number of independently adaptive ellipsoid axes corresponds to the cardinality of the partition. The current minimum estimate is considered the sum of $n$ vectors, one for each of the partitions. Each of these components is modified in turn by the addition of a random vector whose components (in the standard orthonormal base of the given subspace) have a magnitude limited by the corresponding hyper-ellipsoid semi-axis. If the function value at the new minimum candidate is lower than the value at the current point, the searching point (the probe) is moved and the corresponding ellipsoid axis is doubled. Should this not be the case the probe is not moved and the corresponding hyper-ellipsoid axis is halved. The algorithm restarts itself automatically when the ellipsoid axes become smaller than a given threshold by moving the probe in a new randomly chosen point. The algorithm has two remarkable properties:

- it is not necessary for the function to be minimized to have a known analytical form;
- straightforward implementation;
- convergence to the absolute minimum on a compact set with probability 1[1]. While the convergence to the absolute minimum is mainly of theoretical interest (it is assured in the limit of the number of moves going to infinity) it makes the algorithm more insensitive to the presence of local minima.

## 3. Mathematical analysis

While it cannot be assumed that the function to be minimized is a quadratic form, it is a reasonable assumption that its shape near a minimum resembles an elliptical paraboloid. The efficiency with which the algorithm converges to the minimum of a quadratic form is therefore relevant as the higher the efficiency the sooner the algorithm can explore another area of the search space. What's more, the more accurate the convergence to the minimum (given the restarting thresholds) the more reliable the comparison between the values of the local minima to get the global minimum.

Let us consider a positive definite, quadratic form $A$ so that

$$\mathrm{Ker}\,\{(\cdot, A\cdot\,)\} = \{O\} \tag{1}$$

$$(\mathbf{x}_0, A\mathbf{x}_0) > 0,\ \forall \mathbf{x}_0 \in \mathbf{R}^n,\ \mathbf{x}_0 \neq \mathbf{O} \tag{2}$$

where $(\cdot,\cdot\ )$ represents the usual inner product in $\mathbf{R}^n$ and $\mathbf{O}$ is the null element of $\mathbf{R}^n$. Let $\mathbf{x} \in \mathbf{R}^n$ be a random variable, uniformly distributed in the $n$-dimensional sphere of radius $\eta$ ($S_\eta$). Let $\mathbf{x}_0$ be the starting point. We want to compute the probability of moving from the starting point into a point at which the quadratic form assumes a lower value:

$$\mathcal{P}[(\mathbf{x}_0 + \mathbf{x}, A(\mathbf{x}_0 + \mathbf{x})) \leq (\mathbf{x}_0, A\mathbf{x}_0)] = \frac{1}{V^n(\eta)} \int_D dV_x \tag{3}$$

where the domain $D$ is defined by the following equations:

$$\begin{aligned} (\mathbf{x}_0 + \mathbf{x}, A(\mathbf{x}_0 + \mathbf{x})) &\leq (\mathbf{x}_0, A\mathbf{x}_0) \\ (\mathbf{x}, \mathbf{x}) &\leq \eta^2 \end{aligned}$$

and $dV_x$ is short-hand for the volume element $dx_1\, dx_2\, \cdots\, dx_n$. $V^n(\eta)$ represents the measure of the $n$-dimensional sphere and is given by the following expression:

$$V^n(\eta) = \frac{\pi^{(n-1)/2}\Gamma(1/2)}{\Gamma(n/2+1)}\eta^n \tag{4}$$

From the hypothesis on $A$ we can find an orthonormal coordinate transformation taking $A$ to its *metric normal form* (sum of squares)

$$\exists U : U^T U = I,\ U^T A U = D,\ D_{ij} = d_i \delta_{ij},\ d_i > 0 \tag{5}$$

$\delta$ being the Kronecker symbol. The probability $\mathcal{P}$ can be rewritten as:

$$\mathcal{P}[(\mathbf{x}_0 + \mathbf{x}, A(\mathbf{x}_0 + \mathbf{x})) \leq (\mathbf{x}_0, A\mathbf{x}_0)] = \alpha \int_{v \in F} dV_v \tag{6}$$

[1] There is always a non null probability to effect arbitrarly long jumps: it is possible to have a sufficiently long sequence of successful moves making the ellipsoid grow to encapsulate the compact domain of the function and enabling the probe to explore it completely

where:

$$
\begin{aligned}
D &= U^T A U && (7)\\
\mathbf{v}_0 &= \frac{\sqrt{D} U^T \mathbf{x}_0}{\sqrt{(\mathbf{x}_0, A\mathbf{x}_0)}} && (8)\\
(\mathbf{v}_0, \mathbf{v}_0) &= 1 && (9)\\
D_0 &= \left( \frac{\eta}{\sqrt{(\mathbf{x}_0, A\mathbf{x}_0)}} \right)^2 D && (10)\\
\alpha &= \frac{(\mathbf{x}_0, A\mathbf{x}_0)^{n/2}}{V^n(\eta)\sqrt{\det(A)}} && (11)\\
F &= \{\mathbf{v} \in \mathbf{R}^n | (\mathbf{v}, \mathbf{v}) \leq 1; && (12)\\
&\quad (\mathbf{v} - \mathbf{v_0}, D_0^{-1}(\mathbf{v} - \mathbf{v_0})) \leq 1\}
\end{aligned}
$$

The probability $\mathcal{P}$ is then given by the quantity $\alpha$ times the volume of the intersection of the $n$-dimensional sphere of radius 1 and the hyper-ellipsoid whose semi-axes are $\frac{\eta\sqrt{d_j}}{\sqrt{(\mathbf{x}_0, A\mathbf{x}_0)}}$ and center lying on the boundary of the sphere (see Fig.1). The semi-axes of the ellipsoid are proportional to $\eta$. This implies, in the limit $\eta \to 0$, that the intersection volume approaches half the ellipsoid volume from below and then:

$$
\lim_{\eta \to 0} \mathcal{P}[(\mathbf{x}_0 + \mathbf{x}, A(\mathbf{x}_0 + \mathbf{x})) \leq (\mathbf{x}_0, A\mathbf{x}_0)] = \frac{1}{2} \qquad (13)
$$

From the following inequality

$$
\int_{\mathbf{v} \in F} dV_v \leq \int_{\mathbf{v} \in S^n(1)} dV_v \qquad (14)
$$

$\mathcal{P}$ can bound:

$$
\mathcal{P} \leq \min \left( \frac{1}{2}, \frac{(\mathbf{x}_0, A\mathbf{x}_0)^{n/2}}{\eta^n \sqrt{\det(A)}} \right) \qquad (15)
$$

Equation 15 points out a possible weakness of the algorithm: $\mathcal{P}$ can be dramatically affected by even small variations of $\eta$ if the domain dimensionality is high (e.g. if $\eta \to 2\eta$ ,as proposed in [5], $\mathcal{P}$ may be reduced by a factor as high as $2^n$).

When the minimum semi-axis of the ellipsoid is equal to or greater than the sphere diameter, the whole sphere is inside the ellipsoid. This implies that the intersection volume is given by the sphere volume and the probability $\mathcal{P}$ is given by:

$$
\mathcal{P}[(\mathbf{x}_0 + \mathbf{x}, A(\mathbf{x}_0 + \mathbf{x})) \leq (\mathbf{x}_0, A\mathbf{x}_0)] = \frac{(\mathbf{x}_0, A\mathbf{x}_0)^{n/2}}{\eta^n \sqrt{\det(A)}} \qquad (16)
$$

when

$$
\eta \geq 2\sqrt{\frac{(\mathbf{x}_0, A\mathbf{x}_0)}{d}} \qquad (17)
$$

where $d$ represents the minimum eigenvalue of $A$:

$$
d = \min_{j=1,\ldots,n} \{d_j\}
$$

We also have

$$
\mathcal{P}(\eta = 2\sqrt{\frac{(\mathbf{x}_0, A\mathbf{x}_0)}{d}}) = \frac{1}{2^n \sqrt{\prod_j \left(\frac{d_j}{d}\right)}} \leq \frac{1}{2^n} \qquad (18)
$$

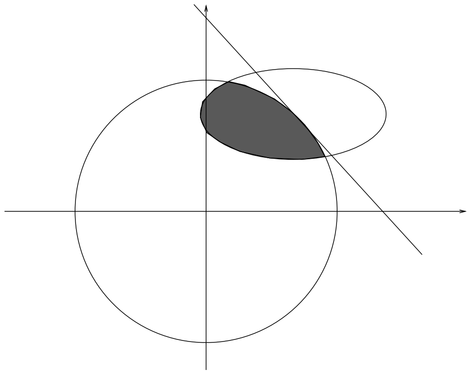

Fig. 1. The shaded area corresponds to a region of lower values of the quadratic form

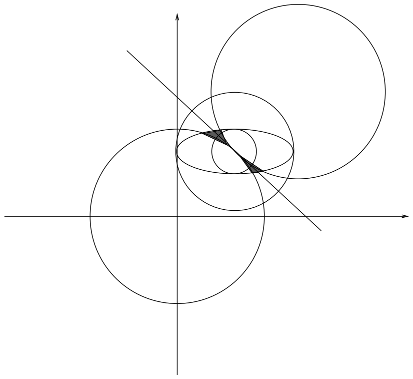

Fig. 2. The shaded area represent the region where the double shot strategy is unsuccessful

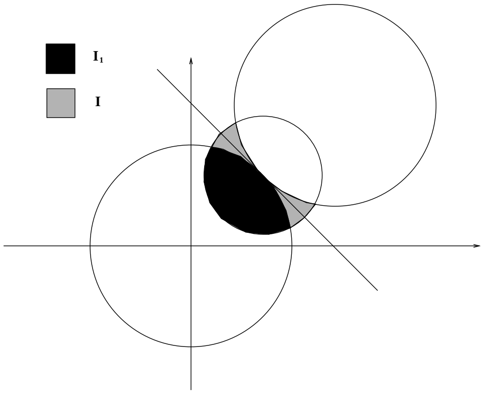

Fig. 3. The black area represents a region of lower function values while the dot filled one represents the region where the double shot strategy is unsuccessful

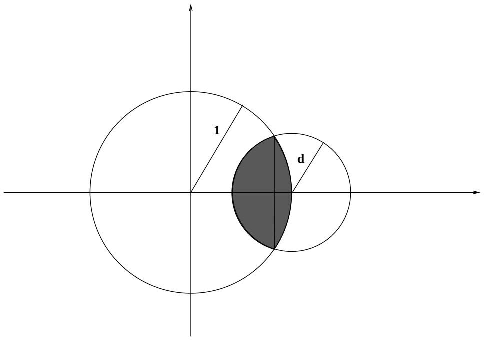

Fig. 4. The integration domain $I_1$ representing the points where the probe can move and lower the function value

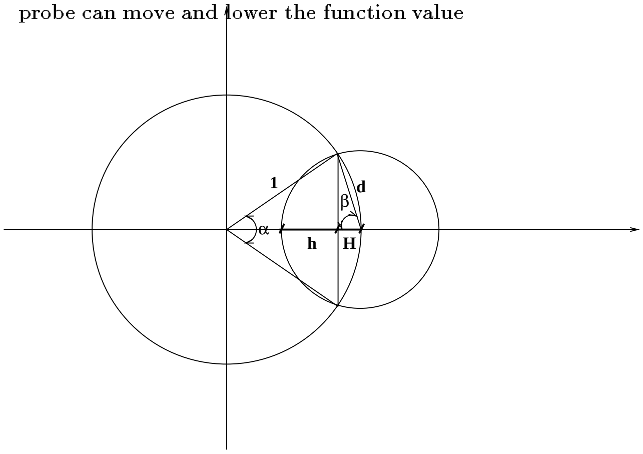

Fig. 5. The geometrical elements used in the computation of the spherical segment heights

## 4. Double shot: an improvement

Can we derive any information from the failure of a guess? We will demonstrate this to be the case. In fact, if $\mathbf{x}+\mathbf{x}_0$ is an unsuccessful guess, we shall show that the probability of failing a new guess, given as $\mathbf{x}-\mathbf{x}_0$, goes to 0 when $\eta \to 0$. The probability of such an event is given by:

$$\mathcal{P}[(\mathbf{x}_0 \pm \mathbf{x}, A(\mathbf{x}_0 \pm \mathbf{x})) \geq (\mathbf{x}_0, A\mathbf{x}_0)] = \frac{(\mathbf{x}_0, A\mathbf{x}_0)^{n/2}}{V^n(\eta)\sqrt{\det(A)}} \int_I dV_v \tag{19}$$

where the integration domain $I$ (see Fig.2) is defined by:

$$\begin{aligned} (\mathbf{v}, \mathbf{v}) &\geq 1 && (20)\\ (\mathbf{v}-2\mathbf{v}_0, \mathbf{v}-2\mathbf{v}_0) &\geq 1 && (21)\\ (\mathbf{v}-\mathbf{v}_0, D_0^{-1}(\mathbf{v}-\mathbf{v}_0)) &\leq 1 && (22) \end{aligned}$$

We will show that:

$$\lim_{\eta\to 0} \frac{(\mathbf{x}_0, A\mathbf{x}_0)^{n/2}}{V^n(\eta)\sqrt{\det(A)}} \int_I dV_v = 0 \tag{23}$$

The starting point is the observation that the ellipsoid volume is bounded from above and below by the spheres whose radii equal the maximum and minimum semi-axis respectively and then the limit in Eq. 23 is upper and lower bounded by the same limit when $I$ refers to the $n$-sphere with radius equal to the biggest (smallest) semi-axis (which we recall to be proportional to $\eta$).

We can then assume the ellipsoid to be a sphere of radius $d$ (see Fig.3). The symmetry of the integration allows us to express the integral we are interested in as:

$$\int_I dV_v = V^{(n)}(d) - 2\int_{I_1} dV_v \tag{24}$$

The computation of volume $I_1$ reduces to the computation of the volume of two spherical segments (see Fig.4) whose heights are given by (see Fig.5):

$$\begin{aligned} H &= \frac{d^2}{2} && (25)\\ h &= d - H && (26) \end{aligned}$$

The volume of an $n$-dimensional spherical segment is:

$$C^{(n)}(h,r) = \frac{\pi^{(n-1)/2}}{\Gamma(\frac{n+1}{2})} \int_{r-h}^{r} (r^2 - t^2)^{(n-1)/2} dt \tag{27}$$

It can be shown [2] that the following expansions hold in the limit of $d \to 0$:

$$\begin{aligned} \int_{d-h}^{d} (d^2 - t^2)^{(n-1)/2} dt &= \frac{\Gamma(\frac{1}{2})\Gamma(\frac{n+1}{2})}{2\Gamma(\frac{n}{2}+1)} d^n - && (28)\\ &- \frac{d^{(n+1)}}{2} + O(d^{n+3}) \\ \int_{1-H}^{1} (1-t^2)^{(n-1)/2} dt &= \frac{d^{n+1}}{n+1} + O(d^{n+3}) && (29) \end{aligned}$$

The volume of $I$ in the limit of small $d$ is then:

$$\int_I dV_v = \frac{\pi^{(n-1)/2}}{\Gamma(\frac{n}{2}+1)} \left(1 - \frac{2}{n+1}\right) d^{n+1} + O(d^{n+3}) \tag{30}$$

and this implies that:

$$\lim_{\eta\to 0} \frac{(\mathbf{x}_0, A\mathbf{x}_0)^{n/2}}{V^n(\eta)\sqrt{\det(A)}} \int_I dV_v = 0 \tag{31}$$

This is an improvement because the probability of two sequential unsuccessful tests goes to zero for small $\eta$ while in the original algorithm it is lower bounded by 1/4 regardless of the value of $\eta$.

[2] Using standard results on the hypergeometric function we have:

$$\int_0^1 (1-x^2)^{(n-1)/2} dx = \frac{\Gamma(1/2)\Gamma((n+1)/2)}{2\Gamma(n/2+1)}$$

while the binomial series allows us to write:

$$\int_0^{H/d} (1-x^2)^{(n-1)/2} dx = \frac{H}{d} + O(\left(\frac{H}{d}\right)^3) = \frac{d}{2} + O(d^3)$$

and

$$\begin{aligned} \int_{1-H}^{1} (1-t^2)^{(n-1)/2} dt &= 2^{(n-1)/2} \int_0^H [x(1-\frac{x}{2})]^{(n-1)/2} \\ &= \frac{2^{(n+1)/2}}{(n+1)} H^{(n+1)/2} + O(H^{(n+3)/2}) \\ &= \frac{d^{(n+1)}}{(n+1)} + O(d^{(n+3)}) \end{aligned}$$

The structure of the algorithm suggests several variations on the basic theme:

- fixed or telescoping partitions: the simplest form is to consider a sphere whose radius is adaptively modified (BATCH mode). Another possibility is to consider each variable separately (the cardinality of the partition corresponds to the dimensionality of the domain: SINGLE mode). A more sophisticated scheme (see [5]) uses a variable partition whose cardinality is increased at each successive restart.
- asymmetric growing/shrinking of the ellipsoid;
- random perturbations in a spherical shell (the probe is then biased to move near the boundary of the noise ellipsoid thereby improving the convergence rate).

All of these variations will be compared experimentally in the next section.

## 5. Experimental results

It is natural to ask how well random minimization can perform whenever a not strictly quadratic form must be minimized. A typical benchmark for minimization routines is the computation of the Rayleigh coefficient $\rho$ of a positive definite quadratic form $\mathbf{A}$ (corresponding to its minimum eigenvalue):

$$\rho = \min_{\mathbf{x}} \mathcal{R}(\mathbf{x}) \qquad (32)$$

where

$$\mathcal{R}(\mathbf{x}) = \frac{\mathbf{x}^T \mathbf{A} \mathbf{x}}{\mathbf{x}^T \mathbf{x}} \qquad (33)$$

The experiments we report compare the performance of Conjugate Gradient (see [6]) minimization with several variations of the random minimization scheme (with and without double shot). The different algorithms are compared varying the dimension of the space and the ratio of the maximum and minimum eigenvalues which affects the speed of convergence of the conjugate gradient algorithm (see [1]). All of the experiments were done using C-language code on a SunSPARCstation1. The *clock ticks* used in the comparison of the algorithms are the computations of $\mathcal{R}$ and of $\nabla\mathcal{R}$, both of complexity $O(n^2)$ with $n$ the dimensionality of the domain. A single evaluation of $\mathcal{R}$ in $R^{100}$ took about 17 msec resulting in a maximum execution time of approximately 90 secs.

As we could expect from the previous analysis, the performance of the random minimization routine is better at low dimensionality and with a small ratio of the extremal eigenvalues.

The most remarkable effect is that of the double shot: trying the reverse of the unsuccessful perturbation dramatically increases the performance of all of the compared schemes at all dimensionalities and eigenvalue ratios.

The use of asymmetric growing/shrinking and the delimitation of the search to a spherical shell has a negative effect on the performance of the BATCH mode. The use of asymmetric growing/shrinking has no major effect on the SINGLE mode which, however, benefits from the limitation of the search to a spherical shell. The global trends at the explored dimensionalities ($n = 10, 50, 100$) and at the different ratio of the extremal eigenvalues ($R = 10, 100, 1000$) is, in order of decreasing performance:

- Conjugate Gradient;
- symmetric, shell limited, double shot SINGLE mode;
- symmetric, double shot BATCH mode;

The SINGLE mode variants perform consistently better than the BATCH mode schemes with the single exception of the runs at the higher dimensionality with the smallest extremal eigenvalues ratio. While the performance of the Conjugate Gradient Algorithm is always superior, the improved random algorithm performs nearly as well at low dimensionalities (for some examples see Figures 6 through 9 where abscissas represent the number of function (gradient) evaluations while ordinatas represent the logarithm of the absolute difference from the minimum eigenvalue).

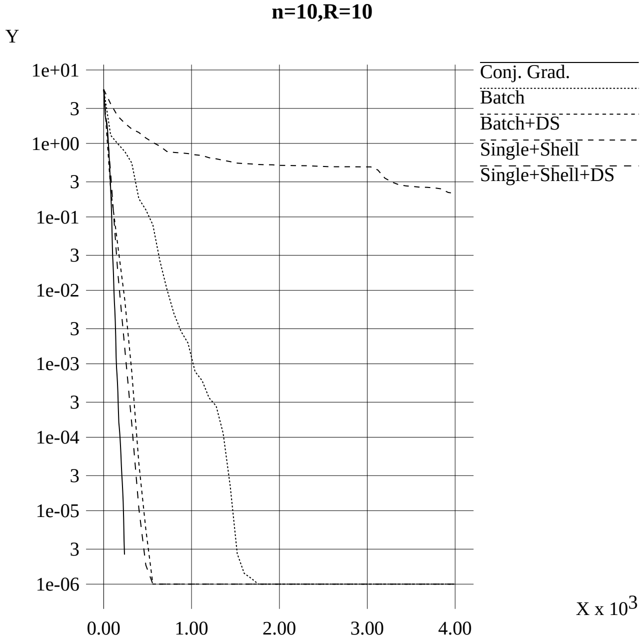

Fig. 6. Comparison between the different minimization schemes at different dimensionality $n$ and different eigenvalue ratio $R$ ($\lambda_M/\lambda_m$). (DS stands for double shot while Shell means shell limited noise. Average data on several runs with different starting points are reported.)

## 6. Conclusions

Some characteristics of a recently proposed non deterministic minimization algorithm have been analytically derived. A simple yet very effective variant has been proposed and its better performance analitically justified. Several variants of the same basic algorithm have been experimentally compared to a Conjugate Gradient approach in the computation of the Rayleigh coefficient of a positive definite quadratic form. The Caprile-Girosi algorithm has also been succesfully applied to the approximation of functions using an expansion in radial basis functions (see [8], [5], [9]) and should prove effective in training standard feedforward neural networks (for a similar approach see [4]).

## References

[1] G.Gambolati, G.Pini, and F.Sartoretto. An improved iterative optimization techniques for the leftmost eigenpars of large symmetric matrices. *Journal of Computational Physics*, 74:41–60, 1988.

[2] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

[3] R. A. Rutenbar. Simulated annealing algorithms: An overview. *IEEE Circuits and Devices Magazine*, 5(1):19–26, 1989.

[4] N. Baba. A new approach for finding the global minimum of error function of neural networks. *Neural Networks*, 2:367–373, 1989.

[5] B. Caprile and F. Girosi. A Nondeterministic Minimization Algorithm. A.I. Memo No. 1254, Massachusetts Institute of Technology, 1990.

[6] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. *Numerical Recipes in C.* Cambridge University Press, 1988.

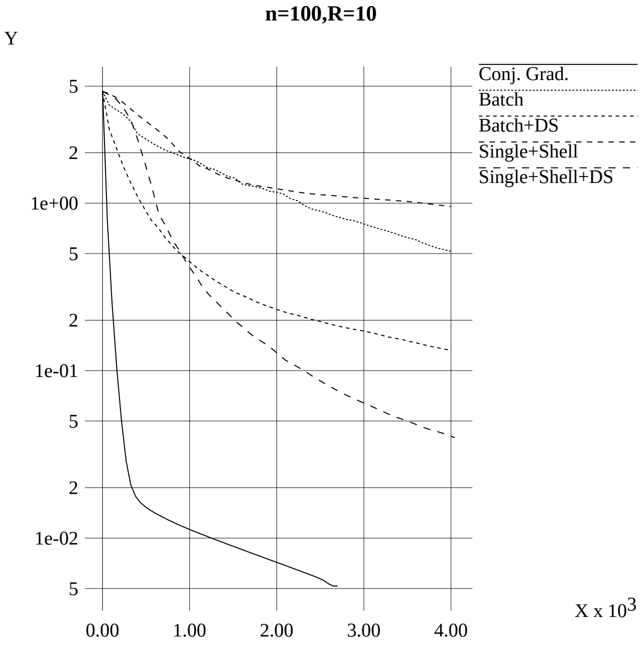

Fig. 7. $n = 100$, $\lambda_M/\lambda_m = 10$

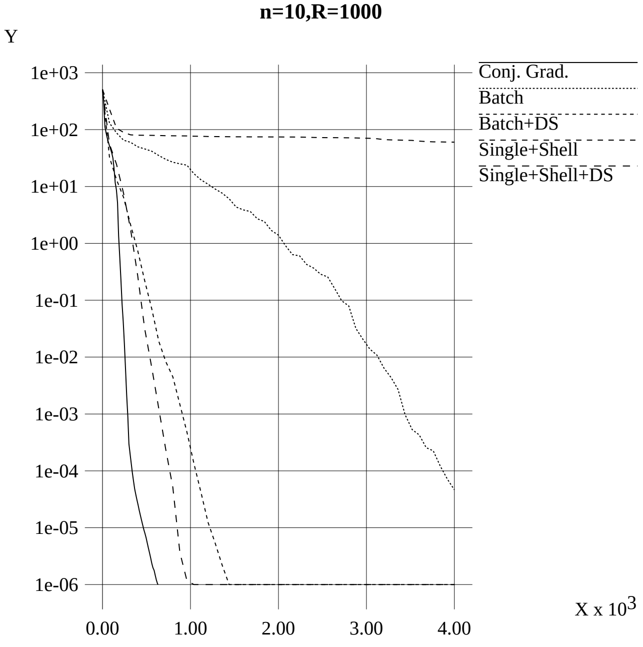

Fig. 8. $n = 10$, $\lambda_M/\lambda_m = 1000$

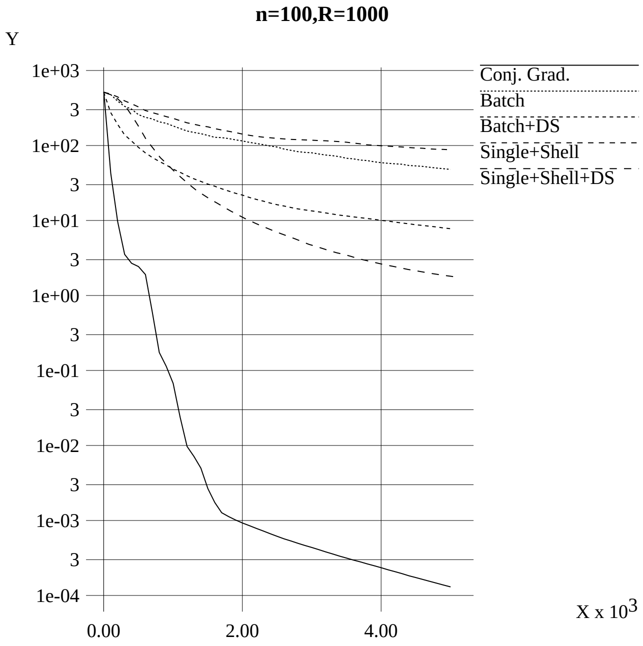

Fig. 9. $n = 100$, $\lambda_M/\lambda_m = 1000$

[7] H. Ratscheck and J. Rokne. *New Computer Methods for Global Optimization*. Ellis Harwood Limited, 1988.

[8] T. Poggio and F. Girosi. A theory of networks for approximation and learning. A.I. Memo No. 1140, Massachusetts Institute of Technology, 1989.

[9] R. Brunelli and T. Poggio. Use of RBF in Real Object Recognition. Technical Report 9011-09, I.R.S.T, 1990.