Computational Optimization and Applications 14, 75–86 (1999)

Finite Sample Behaviour of an Ergodically Fast Line-Search Algorithm*

L. PRONZATO pronzato@unice.fr
Laboratoire I3S, CNRS-UNSA, bât. 4, 250 rue A. Einstein, Sophia Antipolis, 06560 Valbonne, France

H.P. WYNN hpw@stats.warwick.ac.uk
Department of Statistics, University of Warwick, Coventry CV4 7AL, UK

A.A. ZHIGLJAVSKY ZhigljavskyAA@cardiff.ac.uk
School of Mathematics, University of Wales, Cardiff, Senghennydd Road, Cardiff, CF2 4YH, UK

Received August 14, 1996; Revised October 6, 1997; Accepted November 11, 1997

Abstract. In order to represent a line-search algorithm as a non-convergent dynamic system, we perform a renormalisation of the uncertainty interval at each iteration. Its ergodic behaviour can then be studied, and it happens that, for locally symmetric functions, the asymptotic performances of the algorithm suggested are far better than those of the well-known Golden Section algorithm. A proper tuning of internal parameters is then performed to obtain good performances for a finite number of iterations. The case of a function symmetric with respect to its optimum is considered first. An algorithm is presented, that only uses function comparisons, with a significant reduction of the number of comparisons required to reach a given precision when compared to the Golden Section algorithm. The robustness of these performances with respect to non-symmetry of the function is then checked by numerical simulations.

Keywords: line-search, golden section, Fibonacci, dynamic system, ergodic behaviour

1. Introduction

Algorithms that are simply convergent can be considered as dynamic systems, after a suitable transformation is applied to each iteration. The main idea is to renormalise the region containing the search object x^*, for instance the point where a given function $f(\cdot)$ is optimum, so that the target remains in a standard region $\mathcal{R}$. This means that x^* is no longer fixed but moves in $\mathcal{R}$ at each iteration, and follows the evolution of a dynamic system. Methods of analysis relying on ergodic theory and chaos thus apply, see [4]. It happens that the worst cases of the worst-case optimal algorithms often have a Lebesgue measure equal to zero, and algorithms with a better ergodic convergence rate can thus be constructed. The question is then: how fast do these algorithms converge for a finite number of iterations?

We consider here the minimisation of a uniextremal function $f(\cdot)$ on some interval $[A, B]$ ($f(\cdot)$ is nonincreasing for $x \leq x^*$ and increasing for $x > x^*$) using a "second-order"

*This work was supported by a French-British Alliance grant (nb. 94002), by the UK Science and Engineering Research Council grant for the second author, and by a Visiting Fellowship from CNRS for the third author.

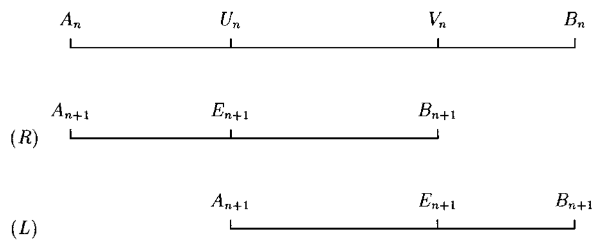

Figure 1. Deletion rule.

algorithm, as defined by Kiefer [3]. These algorithms only compare function values, which makes them especially useful when optimizing a function defined, e.g., by a convergent series, for which arbitrary precise bounds can be constructed (note that algorithms using derivative information, or simply exact function values, may have better performances).

Fix $[A_1, B_1] \supseteq [A, B]$, and an initial point $E_1 \in [A_1, B_1]$. The algorithm compares function values at two points: (1) E_n, from the previous iteration and (2) E'_n, selected by the algorithm at the current iteration.

Define $U_n = \min\{E_n, E'_n\}$, $V_n = \max\{E_n, E'_n\}$, then the deletion rule is:

$$\begin{cases} \text{(R)} \quad \text{if } f(U_n) < f(V_n) \text{ delete } (V_n, B_n] \text{ so that } [A_{n+1}, B_{n+1}] = [A_n, V_n] \\ \text{(L)} \quad \text{if } f(U_n) \geq f(V_n) \text{ delete } [A_n, U_n) \text{ so that } [A_{n+1}, B_{n+1}] = [U_n, B_n]\,. \end{cases} \tag{1}$$

Here (R) and (L) stand for right and left deletion. The deletion rule is illustrated by figure 1. Note that at next iteration the function has already been evaluated at a point in (A_{n+1}, B_{n+1}). This point corresponds to E_{n+1}.

The performance of the algorithm up to iteration n is measured by the length $L_n = (B_n - A_n)$. The reduction or convergence rate of the nth iteration is $r_n = \frac{L_{n+1}}{L_n}$, i.e., $L_n = L_0 \prod_{i=0}^{n-1} r_i$, with $L_0 = B - A$. For instance, the *Golden Section algorithm* corresponds to $[A_1, B_1] = [A, B]$, so that $r_0 = 1$, and

$$\frac{V_n - A_n}{B_n - A_n} = 1 - \frac{U_n - A_n}{B_n - A_n} = \varphi = \frac{\sqrt{5} - 1}{2} \simeq 0.61804.$$

For $n \geq 1$, the reduction rate r_n is constant and equals φ. This algorithm is known to be asymptotically worst-case optimal in the class of all uniextremal functions among algorithms using function values, see [1, 3]. However, better asymptotic convergence rates can be obtained for locally symmetric functions, even if only comparison of function values is used [5, 7]. We shall consider here the asymptotic and finite sample behaviour of the window algorithm discussed in [6].

The algorithm is defined by

$$\begin{aligned} [A_1, B_1] &= [A - \epsilon(B - A), B + \epsilon(B - A)], \\ E_1 &= (A_1 + B_1)/2 - w(B_1 - A_1)/2, \end{aligned} \tag{2}$$

and

$$E'_n = \begin{cases} E_n + w(B_n - A_n) & \text{if } E_n < \frac{1}{2}(A_n + B_n)\,, \\ E_n - w(B_n - A_n) & \text{otherwise,} \end{cases} \tag{3}$$

where $\epsilon \geq 0$ and $w > 0$ are tuning parameters. The ratio $|E'_n - E_n|/(B_n - A_n)$, defining the window width, is thus fixed and equals w. Note that $r_0 > 1$ for $\epsilon > 0$, the reason for expanding the interval $[A, B]$ will be explained later, see Section 3.2. Also note that the GS algorithm corresponds to $\epsilon = 0$, $w = 2\varphi - 1$.

2. Ergodic behaviour

Consider the special case when $f(x)$ is symmetric around x^* and let x_n and u_n, respectively denote the renormalised locations of x^* and U_n in the current uncertainty interval $[A_n, B_n]$, that is:

$$x_n = \frac{x^* - A_n}{B_n - A_n}, \quad u_n = \frac{U_n - A_n}{B_n - A_n}, \qquad n \geq 1.$$

Due to the symmetry of $f(\cdot)$, the conditions for application of the right or left deletion rule in (1) then become

$$\begin{cases} (R) & \text{if } x_n < u_n + w/2 \\ (L) & \text{if } x_n \geq u_n + w/2. \end{cases}$$

Renormalisation back to $[0,\ 1]$ yields the following two-dimensional dynamic process for the window algorithm:

$$\begin{aligned} x_{n+1} &= \begin{cases} \dfrac{x_n}{u_n + w} & \text{if } x_n < u_n + w/2 \\ \dfrac{x_n - u_n}{1 - u_n} & \text{if } x_n \geq u_n + w/2 \end{cases} \\ u_{n+1} &= \begin{cases} \dfrac{u_n}{u_n + w} - w & \text{if } x_n < u_n + w/2 \\ \dfrac{w}{1 - u_n} & \text{if } x_n \geq u_n + w/2 \end{cases} \end{aligned} \tag{4}$$

Figure 2 presents a plot of the sequence of iterates $(x_n, u_n), n = 1, \ldots, 50{,}000$, when $w = 1/8$.

The sequence of reduction rates associated with the dynamical system (4) is

$$r_n = \begin{cases} u_n + w & \text{if } x_n < u_n + w/2\,, \\ 1 - u_n & \text{if } x_n \geq u_n + w/2\,, \end{cases}$$

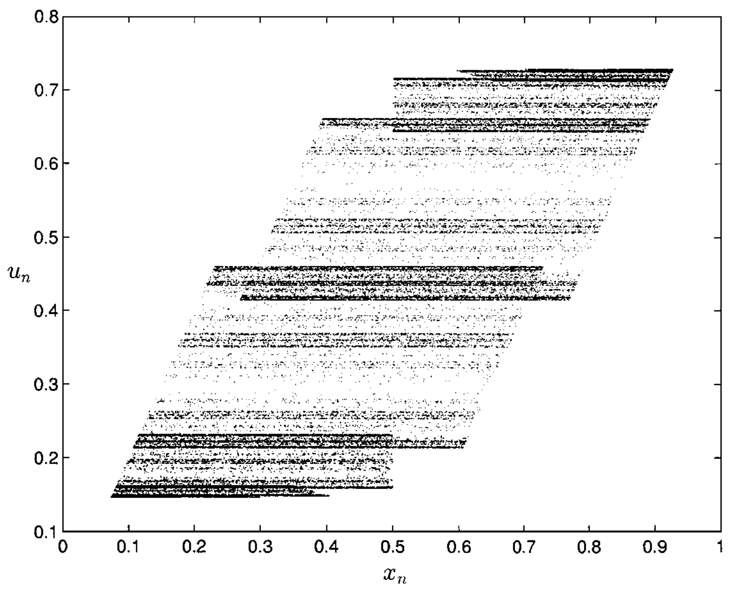

Figure 2. Iterates of the dynamic system (4).

with $r_0 = 1 + 2\epsilon$. Obviously, after n evaluations of $f(\cdot)$, the length L_n of the current uncertainty interval depends on the value of x^*, that is $L_n = L_n(x^*)$.

The advantage of setting up the problem as a dynamic process is that it allows us to use results from ergodic theory, and we can link the rate of convergence of the algorithm with the characteristics of the dynamic process. The ergodic reduction rate is defined by

$$r = \lim_{N\to\infty} (L_N)^{\frac{1}{N}} = \lim_{N\to\infty} \left(L_0 \prod_{n=0}^{N-1} r_n \right)^{\frac{1}{N}},$$

if this limit exists for almost all x^* with respect to the Lebesgue measure. Since $L_0 = B - A < \infty$ and $\epsilon < \infty$, this becomes

$$r = \lim_{N\to\infty} \left(\prod_{n=1}^{N} r_n \right)^{\frac{1}{N}}. \tag{5}$$

The Jacobian matrix of the transformation $(x_n,\ u_n) \to (x_{n+1},\ u_{n+1})$ is upper triangular and can be written as

$$J(x_n,\ u_n) = \begin{pmatrix} \frac{1}{r_n} & \alpha_n \\ 0 & \frac{w}{r_n^2} \end{pmatrix}, \tag{6}$$

where α_n is given by

$$\alpha_n = \begin{cases} -\dfrac{x_n}{r_n^2} & \text{if } x_n < u_n + w/2, \\ -\dfrac{1 - x_n}{r_n^2} & \text{if } x_n \geq u_n + w/2. \end{cases}$$

The Lyapunov exponents of the dynamic system (4) are defined by, see [2],

$$\Lambda_i(x_1, u_1) = \lim_{n\to\infty} \log|\lambda_{i,n}(x_1, u_1)|, \quad i = 1, 2, \tag{7}$$

where $\lambda_{i,n}(x_1, u_1)$ is the ith eigenvalue of $J^{(n)}(x_1, u_1)$, with

$$J^{(n)}(x_1, u_1) = J(x_n, u_n)J(x_{n-1}, u_{n-1}) \cdots J(x_1, u_1).$$

Ergodic theory (Oseledec's Theorem) guarantees that the set of points (x_1, u_1) for which the limit in (7) exists and is the same has full measure. The triangular form of the Jacobian (6) gives

$$J^{(n)}(x_1, u_1) = \begin{pmatrix} \left(\prod_{i=1}^n r_i\right)^{-1} & \beta_n \\ 0 & w\left(\prod_{i=1}^n r_i\right)^{-2} \end{pmatrix},$$

where β_n is some number. The Lyapunov exponents of the dynamic system (4) are thus related to r and w by $\lambda_1 = -\log r$, $\lambda_2 = 2\lambda_1 + \log w$. This implies that $r = r(x^*)$ is the same for almost all $x^* \in [A, B]$, hence the name of ergodic reduction rate. In particular, this implies

$$\lambda_1 = \lim_{n\to\infty} E\left\{\frac{\log L_n(x^*)}{n}\right\},$$

where the expectation is taken with respect to a probability measure absolutely continuous with respect to the Lebesgue measure.

A numerical evaluation of the second Lyapunov exponent shows that it is negative for $w < 2\varphi - 1$, which indicates that the dynamic system $\{(x_n, u_n)\}$ may attract to a fractal. For instance when $w = 1/8$, the Lyapunov exponents are approximately $\lambda_1 \simeq 0.639$, $\lambda_2 \simeq -0.801$. The Lyapunov dimension of the attractor of the corresponding dynamic system, presented in figure 2, is then $1 - \frac{\lambda_1}{\lambda_2} \simeq 1.798 < 2$ and the ergodic rate is $r \simeq 0.528$. When $w = 0.15$, the value which will be used in the algorithm of Section 3, we get $\lambda_1 \simeq 0.630$, $\lambda_2 \simeq -0.636$ and $r \simeq 0.532$. This increase in the value of r permits, however, to obtain better performances for finite N ($N \leq 30$). More generally, for any w in $[1/8, 2\varphi - 1)$ the ergodic rate of the window algorithm is better than the rate $r = \varphi = (\sqrt{5} - 1)/2$ of the Golden Section algorithm.

One can show that we cannot achieve an ergodic rate less than $1/2$ for all x^* lying in a set of positive Lebesgue measure [4], and a theoretical algorithm achieving this optimal ergodic rate for a class of locally smooth functions has been found in [7]. However, this algorithm is not efficient for reasonably small values of N. Other algorithms with values of r less that φ can be found in [5]. Their theoretical asymptotic properties can be investigated more deeply than for the window algorithm presented here. However, none of them achieves the performances of the window algorithm for N reasonably small.

3. Finite sample behaviour

3.1. *Performance criteria*

We investigate now the behaviour of the window algorithm for finite N and locally smooth functions, under the assumption that x^* is uniformly distributed over the initial interval $[A, B]$. Various performance criteria are considered, which are functions of ϵ, w and N:

$$EL_N = E\{L_N(x^*)\}, \quad ML_N = \max_{x^* \in [A,B]} L_N(x^*),$$

$$L_N^{1-\alpha} = \sup\{t \mid Pr\{L_N(x^*) \geq t\} < \alpha\},$$

$$P_N^{GS} = Pr\{L_N(x^*) < L_0 \varphi^{N-1}\}, \quad P_N^F = Pr\left\{L_N(x^*) < \frac{L_0}{F_{N+1}}\right\},$$

where F_n is the nth term of the Fibonacci sequence, i.e., $F_1 = F_2 = 1$ and $F_n = F_{n-1} + F_{n-2}$, $n > 2$. The criteria P_N^{GS} and P_N^F, respectively, correspond to the probability that the window algorithm converges faster than the Golden Section and the Fibonacci [3] algorithms. Other criteria could also be considered, although they are less appealing from a practical point of view, for instance: $\phi_N^1 = E\{-\log L_N(x^*)\}/N$, $\phi_N^2 = E\{(L_N(x^*))^{1/N}\}$.

When the function $f(\cdot)$ is symmetric with respect to x^*, the behaviour of the algorithm is completely determined by that of the dynamic system (4). From the discussion of previous section one then has

$$\lim_{N\to\infty} \phi_N^1 = -\log r, \quad \lim_{N\to\infty} \phi_N^2 = r\,.$$

One can also easily prove that P_N^{GS} and P_N^F tend to 1 as N tends to infinity.

3.2. *Algorithm*

The ergodic characteristic r does not depend on ϵ. However, a good tuning of w for the ergodic criterion r, e.g., $w = 1/8$, is far from being optimal for small N for all criteria considered here, whatever the choice of ϵ. A proper tuning of ϵ and w is thus required.

For a fixed value of w, choosing ϵ large enough ($\epsilon \geq \frac{1-w}{2(1+w)}$), that is expanding the initial interval $[A, B]$, guarantees that x_1 belongs to the support of the invariant measure for x_n, see figure 2. This is of crucial importance, since it permits to obtain finite sample characteristics close to their asymptotic values. Also, numerical investigations demonstrate that initial points x_1 outside the support of the invariant measure for x_n give bad convergence rates in the first iterations.

The values of ϵ and w could be chosen optimally for each N and each criterion. A rather exhaustive analysis for reasonable values of N, $10 \leq N \leq 30$, which corresponds to standard values for this type of line-search algorithms, has lead to the choice $\epsilon = 0.3772$ and $w = 0.15$, which is close to optimality for the criteria of Section 3.1. When $f(\cdot)$ is symmetric with respect to x^*, and x^* has a uniform distribution on $[A, B]$, these criteria can be computed exactly as shown in Section 3.3. The rest of the paper analyses the finite sample behaviour of the corresponding algorithm, summarized as follows.

Preliminary step. Let $f(\cdot)$ be the function to be minimised on $[A, B]$. Choose $N \geq 2$ and/or $\delta > 0$, which defines the stopping rule, see Step 3.

Step 1. Compute $[A_1, B_1]$ and E_1 as indicated in (2), with $w = 0.15$, $\epsilon = 0.3772$. Compute $f(E_1)$, set $n = 1$.

Step 2. Compute E'_n according to (3) and the corresponding value $f(E'_n)$ (if $E'_n < A$ take $f(E'_n) = f(A) + A - E'_n$, if $E'_n > B$ take $f(E'_n) = f(B) + E'_n - B$).

If $E_n \leq E'_n$ set $U = E_n,\ V = E'_n,\ f_U = f(E_n),\ f_V = f(E'_n)$,
otherwise set $U = E'_n,\ V = E_n,\ f_U = f(E'_n),\ f_V = f(E_n)$.

Step 3.

If $f_U < f_V$ set $[A_{n+1}, B_{n+1}] = [A_n, V]$,
otherwise set $[A_{n+1}, B_{n+1}] = [U, B_n]$.

If ($n + 1 \geq N$ and/or $B_{n+1} - A_{n+1} \leq \delta$) then stop, otherwise set $n + 1 \to n$ and go to Step 2.

3.3. *$f(\cdot)$ symmetric with respect to x^**

When $f(\cdot)$ is symmetric with respect to x^*, the decision about the right or left deletion in (1) only depends on the location of x^* in the interval. For any fixed N, the initial interval $[A, B]$ can thus be partitioned into an union of disjoint subintervals $\cup_i I_N^i$, such that the behaviour of the algorithm is the same up to iteration N for all x^* in I_N^i. Note that the cardinality of this partition grows exponentially with N (it is already 11,760 for $N = 16$). Let $I_1^1 = [A, B]$, $U_1^1 = U_1 = E_1$. At iteration 2, I_1^1 is split into two subintervals $I_2^1 = [A, (A + B)/2)$, $I_2^2 = [(A + B)/2, B]$, corresponding respectively to right and left deletions. In the first case U_1^1 is updated into $U_2^1 = U_1^1 - w(E'_1 - A_1)$, in the second into $U_2^2 = E'_1$. Using similar arguments, we can track the path of the endpoints of I_n^i, together with U_n^i, in order to construct the partition $\cup_i I_n^i$ for any n: at iteration n, each interval I_n^i is updated into either I_{n+1}^i or the union of two disjoint intervals, depending on the location of U_n^i with respect to I_n^i.

Table 1. Performance characteristics of the Golden Section, Fibonacci and window algorithms ($L_0 = 1$).

N	φ^{N-1}	$\frac{1}{F_{N+1}}$	EL_N	ML_N	P_N^{GS}	P_N^F	$L_N^{0.99}$
3	0.3820	0.3333	0.6084	0.7456	0.0000	0.0000	0.7456
4	0.2361	0.2000	0.3593	0.5943	0.2135	0.0000	0.5943
5	0.1459	0.1250	0.1960	0.4825	0.1958	0.1958	0.4825
6	9.017×10^{-2}	7.692×10^{-2}	9.760×10^{-2}	0.1615	0.4445	0.4445	0.1615
7	5.573×10^{-2}	4.762×10^{-2}	5.433×10^{-2}	9.660×10^{-2}	0.6887	0.5864	9.660×10^{-2}
8	3.444×10^{-2}	2.941×10^{-2}	2.939×10^{-2}	7.237×10^{-2}	0.7993	0.5570	7.237×10^{-2}
9	2.129×10^{-2}	1.818×10^{-2}	1.546×10^{-2}	5.788×10^{-2}	0.7893	0.7318	5.788×10^{-2}
10	1.316×10^{-2}	1.124×10^{-2}	8.326×10^{-3}	1.954×10^{-2}	0.7581	0.7211	1.937×10^{-2}
11	8.131×10^{-3}	6.944×10^{-3}	4.516×10^{-3}	1.495×10^{-2}	0.8831	0.8831	1.159×10^{-2}
12	5.025×10^{-3}	4.292×10^{-3}	2.420×10^{-3}	9.271×10^{-3}	0.9619	0.9376	8.657×10^{-3}
13	3.106×10^{-3}	2.653×10^{-3}	1.295×10^{-3}	6.944×10^{-3}	0.9900	0.9254	3.044×10^{-3}
14	1.919×10^{-3}	1.639×10^{-3}	6.983×10^{-4}	2.344×10^{-3}	0.9806	0.9535	2.209×10^{-3}
15	1.186×10^{-3}	1.013×10^{-3}	3.763×10^{-4}	1.807×10^{-3}	0.9736	0.9572	1.302×10^{-3}
16	7.331×10^{-4}	6.261×10^{-4}	2.020×10^{-4}	1.114×10^{-3}	0.9890	0.9890	8.506×10^{-4}
17	4.531×10^{-4}	3.870×10^{-4}	1.085×10^{-4}	8.432×10^{-4}	0.9982	0.9974	3.476×10^{-4}
18	2.800×10^{-4}	2.392×10^{-4}	5.840×10^{-5}	2.877×10^{-4}	0.9990	0.9938	2.084×10^{-4}
19	1.731×10^{-4}	1.478×10^{-4}	3.141×10^{-5}	2.168×10^{-4}	0.9993	0.9934	1.396×10^{-4}
20	1.070×10^{-4}	9.136×10^{-5}	1.688×10^{-5}	1.337×10^{-4}	0.9977	0.9966	5.590×10^{-5}
21	6.611×10^{-5}	5.646×10^{-5}	9.074×10^{-6}	1.012×10^{-4}	0.9996	0.9996	3.656×10^{-5}
22	4.086×10^{-5}	3.490×10^{-5}	4.881×10^{-6}	3.477×10^{-5}	1.0000	1.0000	2.140×10^{-5}
23	2.525×10^{-5}	2.157×10^{-5}	2.625×10^{-6}	2.633×10^{-5}	0.9999	0.9998	9.926×10^{-6}
24	1.561×10^{-5}	1.333×10^{-5}	1.411×10^{-6}	1.623×10^{-5}	1.0000	0.9994	5.928×10^{-6}
25	9.645×10^{-6}	8.238×10^{-6}	7.587×10^{-7}	1.228×10^{-5}	0.9999	0.9999	3.447×10^{-6}
26	5.961×10^{-6}	5.091×10^{-6}	4.080×10^{-7}	4.173×10^{-6}	1.0000	1.0000	1.646×10^{-6}
27	3.684×10^{-6}	3.147×10^{-6}	2.194×10^{-7}	3.158×10^{-6}	1.0000	1.0000	9.468×10^{-7}
28	2.277×10^{-6}	1.945×10^{-6}	1.180×10^{-7}	1.947×10^{-6}	1.0000	1.0000	5.598×10^{-7}
29	1.407×10^{-6}	1.202×10^{-6}	6.342×10^{-8}	1.474×10^{-6}	1.0000	1.0000	2.908×10^{-7}
30	8.697×10^{-7}	7.428×10^{-7}	3.410×10^{-8}	5.065×10^{-7}	1.0000	1.0000	1.558×10^{-7}

When x^* has a uniform distribution on $[A, B]$, the values of all criteria above can easily be computed with any given arbitrary precision once the partition is constructed. The results are summarized in Table 1. Figure 3 gives the evolution of some performance characteristics as functions of N.

Table 2 presents the value of N required for the corresponding characteristic to reach the precision indicated. For example, the Fibonacci algorithm requires 30 function evaluations to reduce the length of the initial interval by a factor 10^6, while the window algorithm requires respectively 25 and 27 evaluations to achieve the same precision, on the average and with probability 0.99.

Table 2. Value of N required to achieve a given precision ($L_0 = 1$).

Criterion	10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}
φ^{N-1}	6	11	16	21	25	30
$\frac{1}{F_{N+1}}$	6	11	16	20	25	30
EL_N	6	10	14	18	21	25
ML_N	7	12	17	22	26	30
$L_N^{0.99}$	7	12	16	20	23	27

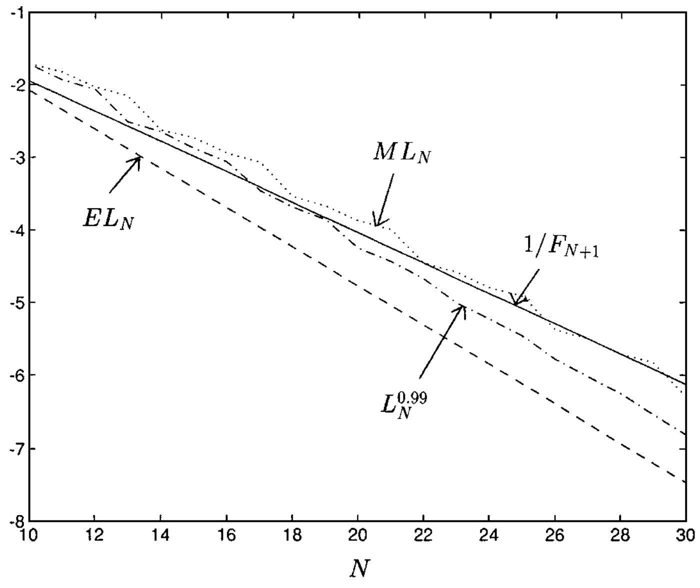

Figure 3. Decimal logarithm of various performance characteristics ($L_0 = 1$).

3.4. $f(\cdot)$ locally smooth at x^*

Consider the class of functions satisfying

$$C|x - x^*|^2 - D|x - x^*|^3 \leq |f(x) - f(x^*)| \leq C|x - x^*|^2 + D|x - x^*|^3,$$

with $C > 0$, which includes the class of all functions three times continuously differentiable at x^*. We assume, however, that no information about derivatives is available. Without any restriction, one can assume that $L_0 = B - A = 1$. Since the algorithm only uses function comparisons, $f(\cdot)$ can be scaled down so that one can also assume that $C = 1$. Figure 4 presents the graph of the functions $f_{0.5}(x - x^*)$ and $f_2(x - x^*)$, with

$$f_D(z) = \begin{cases} \dfrac{4}{27D^2} & \text{if } z \leq -\dfrac{2}{3D} \\ z^2 + Dz^3 & \text{if } -\dfrac{2}{3D} < z, \end{cases}$$

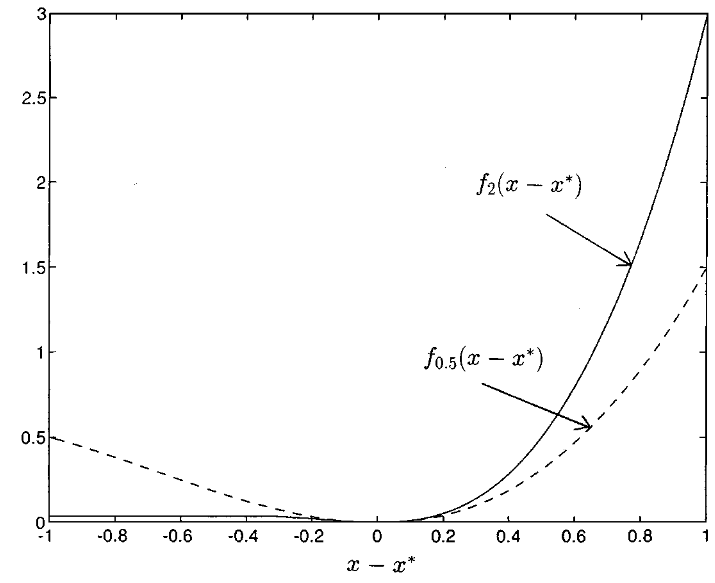

Figure 4. Graphs of $f_{0.5}(\cdot)$ and $f_2(\cdot)$.

which can be considered as the worst uniextremal function in the class above with respect to the symmetry condition, for a given value D.

Numerical simulations with x^* uniformly distributed in $[A, B]$ have been performed, with the factor of non symmetry D varying from 0 to 10. Figure 5 presents the evolution of the empirical values of the performance characteristics EL_{30} and ML_{30} as functions of D. Note that the scales are different on figures 5(a) and (b). Figure 6 presents the evolution of the empirical probability P_N^F as a function of N for several values of D. The performances of the window algorithm remain fairly stable while non symmetry increases. Note that $f_D(\cdot)$ is already quite non symmetric when $D = 2$ as can be seen in figure 4. Performances with functions locally symmetric at x^* (and not necessarily locally smooth) would be similar.

4. Conclusions

The construction of line-search algorithms with better ergodic performances than the famous Golden Section algorithm [6, 7] was an important theoretical achievement. More important for the practical point of view is the construction of algorithms with good performances for finite N. We have shown that for functions symmetric with respect to the optimum, a very simple algorithm permits to achieve any given precision with significantly less comparisons of function values than the Golden Section algorithm. In the case of locally smooth functions, numerical simulations have shown that the performances remain quite robust with respect to non-symmetry of the function. This makes the algorithm suggested a promising alternative to the Golden Section and Fibonacci algorithms.

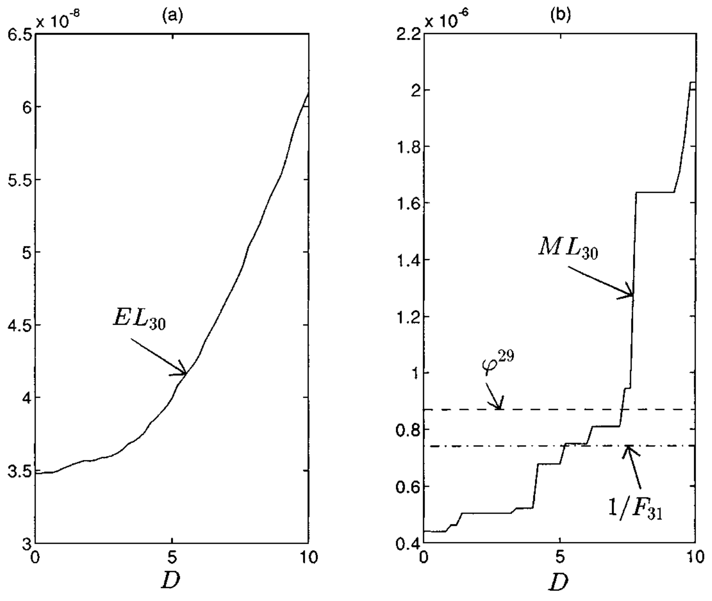

Figure 5. Evolution of EL_{30}, ML_{30}, φ^{29} and $1/F_{31}$ as functions of D ($L_0 = 1$).

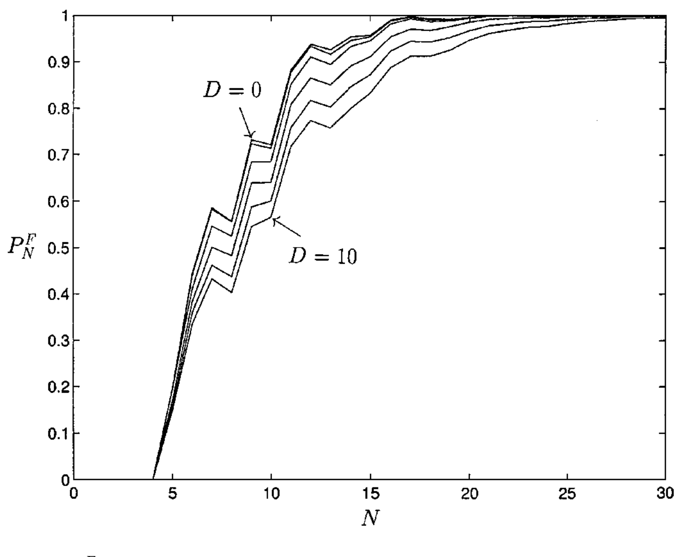

Figure 6. Evolution of P_N^F as a function of N for different values of D (from top to bottom: $D = 0, 2, 4, 6, 8, 10$).

References

1. D.-Z. Du and F.K. Hwang, Combinatorial Group Testing and its Applications, World Scientific: Singapore, 1993.
2. J.-P. Eckmann and D. Ruelle, “Ergodic theory of chaos and strange attractors,” Rev. Mod. Phys., vol. 57, pp. 617–656, 1985.
3. J. Kiefer, “Optimum sequential search and approximation methods under minimum regularity assumptions,” J. Soc. Indust. Appl. Math., vol. 5, no. 3, pp. 105–136, 1957.
4. L. Pronzato, H.P. Wynn, and A.A. Zhigljavsky, “Dynamic systems in search and optimisation,” CWI Quarterly, vol. 8, no. 3, pp. 201–236, 1995.
5. L. Pronzato, H.P. Wynn, and A.A. Zhigljavsky, “Stochastic analysis of convergence via dynamic representation for a class of line-search algorithms,” Combinatorics, Probability and Computing, vol. 6, pp. 205–229, 1997.
6. H.P. Wynn and A.A. Zhigljavsky, “Chaotic behaviour of search algorithms,” Acta Applicandae Mathematicae, vol. 32, pp. 123–156, 1993.
7. H.P. Wynn and A.A. Zhigljavsky, “Achieving the ergodically optimal convergence rate for a one-dimensional minimization problem,” J. of Complexity, vol. 11, pp. 196–226, 1994.