CWI Quarterly Volume 8 (3) 1995, pp. 201 – 239

# Dynamic Systems in Search and Optimisation[1]

Luc Pronzato
*Laboratoire I3S, CNRS-URA 1376, Sophia Antipolis, 06560 Valbonne, France,*
Henry P. Wynn
*Dept of Statistics, University of Warwick, Coventry CV4 7AL, UK*
Anatoly A. Zhigljavsky
*Dept of Mathematics, St.Petersburg University, Bibliotechnaya sq. 2, 198904, Russia*

Even in simple cases, 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 $f(.)$ is optimum, so that the target remains in a standard region $\mathcal{R}_0$. Therefore $x^*$ is no longer fixed but moves in $\mathcal{R}_0$ at each iteration, and follows the evolution of a dynamic system. Various algorithms are considered, e.g. line search, ellipsoid, steepest descent. The associated dynamic systems show different features: in some cases the asymptotic behaviour depends on $x^*$ and/or on the function $f(.)$, and periodic and strange attractors are typical. A main purpose is to use the theory of dynamic systems to analyse the asymptotic behaviour of the algorithms, and, when possible, to obtain optimal rates of convergence.

## 1. Renormalisation

There exist well known applications of dynamical systems to the study of optimisation and search algorithms. A classical example is the study of the Newton map for the cube roots of unity, where the boundaries between the basins of attraction to the roots form a fractal. The present work develops another approach, in which the main ingredient is *renormalisation*. The idea can be explained as follows.

[1] This work was supported by a French-British Alliance grant (nx94002) and by the UK Science and Engineering Research Council grant for the second author, including a Visiting Fellowship for the third author, and also by a grant of the French Ministère de l'Education Nationale (invitation triennale, procédure PAST) for the third author.

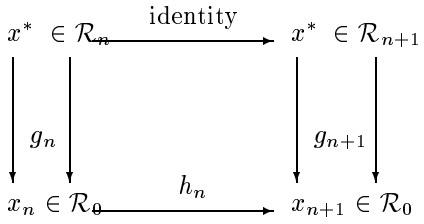


FIGURE 1. Renormalisation

Certain kinds of search and optimisation problems to find a target point $x^* \in \mathcal{R}$ generate a nested sequence of regions $\mathcal{R}_i$, of the same shape, containing $x^*$

$$\mathcal{R} \subseteq \mathcal{R}_1 \supseteq \mathcal{R}_2 \supseteq \ldots .$$

Convergence to $x^*$ takes place if the diameter of the sets converges in some metric to zero. The nesting of the regions occurs because additional observations or data are available at each stage creating additional assumptions on the set $\mathcal{R}_n$ of allowable, or *consistent*, target values. Note that $\mathcal{R}_1$ should have a simple shape, and might be taken larger than $\mathcal{R}$.

Suppose that for each $n$ there is a function $g_n(.)$ mapping $\mathcal{R}_n$ back to some base region $\mathcal{R}_0$:

$$g_n : \mathcal{R}_n \longmapsto \mathcal{R}_0 .$$

Let $x_n = g_n(x^*)$ be the location of the target in the renormalised region. The relation

$$x_{n+1} = h_n(x_n)$$

describes the path of the process for a given $x^*$, with the starting value $x_1 = g_1(x^*)$. The mapping $h_n(.)$ sets up a dynamic system $\{x_n\}$ in $\mathcal{R}_0$ itself: instead of the target being fixed and $\mathcal{R}_n$ changing, $\mathcal{R}_0$ is fixed and the target moves. Figure 1 describes the mapping.

Perhaps the simplest example is the ordinary (first-order) bisection method to find zero of a monotonic function $f(x)$ in some interval $\mathcal{R}$, with $\mathcal{R}_1 = \mathcal{R} = \mathcal{R}_0 = [0,1]$. Define the renormalised function $f_n(.)$ on $\mathcal{R}_0$ by

$$f_n(x) = f(g_n^{-1}(x)) . \tag{1}$$

At each iteration $n$ of the algorithm $f(.)$ is observed at the midpoint of the current uncertainty interval. When the interval has been renormalised to $[0,1]$ and $f_n(0) < 0 < f_n(1)$, in the case $f_n(\frac{1}{2}) > 0$ delete the interval $(\frac{1}{2}, 1]$ and renormalise $[0, \frac{1}{2}]$ to $[0,1]$. Analogously, if $f_n(\frac{1}{2}) \leq 0$ delete the interval $[0, \frac{1}{2})$ and renormalise $[\frac{1}{2}, 1]$ to $[0,1]$.

In many cases, the form of the function $f(.)$ is such that the dynamic system $\{x_n\}$ is *time homogeneous*, that is the mapping $x_n \to x_{n+1}$ does not depend on $n$. In particular, for the bisection method, if $f(.)$ is linear and $x_n$ denotes the location of $x^*$ in the current renormalised interval, then the updating rule for $\{x_n\}$ is $h: \; x_n \to x_{n+1}$, where

$$h(x_n) = \begin{cases} 2x_n & \text{if} \quad 0 \le x_n < \frac{1}{2} \\ \\ 2x_n - 1 & \text{if} \quad \frac{1}{2} \le x_n \le 1 \; . \end{cases} \tag{2}$$

This mapping is sometimes called the *Bernoulli shift*. It is well known to have an invariant measure which is uniform in $[0,1]$: for starting values $x_1 = x^*$ which are *normal numbers*[2] to the base 2, the sequence $x_n$ has a uniform asymptotic density.

Another class of examples arises from minimisation of a uniextremal function on some interval $\mathcal{R}$ using a "second-order" algorithm. The algorithm, in its renormalised form, compares function values at two points, first, $e_n$, from the previous iteration and second, $e'_n$, selected by the algorithm at the current iteration. Let $\mathcal{R}_0 = [0,1]$. Any choice for $e_1 \in [0,1]$ and any function $\phi(.) : [0,1] \longrightarrow [0,1]$, with $e'_n = \phi(e_n)$ then defines a second-order algorithm. Define

$$a_n = \min\{e_n, e'_n\}, \quad b_n = \max\{e_n, e'_n\} \, ,$$

then the deletion rule is:

$$\begin{cases} (\mathrm{R}): & \text{if } f_n(a_n) < f_n(b_n) \text{ delete } (b_n, 1] \\ (\mathrm{L}): & \text{if } f_n(a_n) \ge f_n(b_n) \text{ delete } [0, a_n) \, , \end{cases}$$

with $f_n(.)$ defined by (1). Here (R) and (L) stand for right and left deletion. The remaining interval is then renormalised to $[0,1]$.

The *Golden Section algorithm* corresponds to $\mathcal{R}_1 = \mathcal{R}$ and

$$b_n = 1 - a_n = \lambda = \frac{\sqrt{5}-1}{2} \simeq 0.61804 \, ,$$

i.e. to $e'_n = 1 - e_n$, with $e_1 = \lambda$. In the special case when $f(x)$ is symmetric around $x^*$ the algorithm yields the time homogeneous dynamic process $x_{n+1} = h(x_n)$, where $x_1 = x^*$ and

$$h(x_n) = \begin{cases} x_n(1+\lambda) & \text{if} \quad x_n < \frac{1}{2} \\ \\ x_n(1+\lambda) - \lambda & \text{if} \quad x_n \ge \frac{1}{2} \; . \end{cases} \tag{3}$$

[2] A number in $[0,1]$ is normal to the base 2 if the infinite sequence of 0's and 1's corresponding to its binary representation is such that each finite string of 0's and 1's of given length occurs with the same frequency in the sequence.

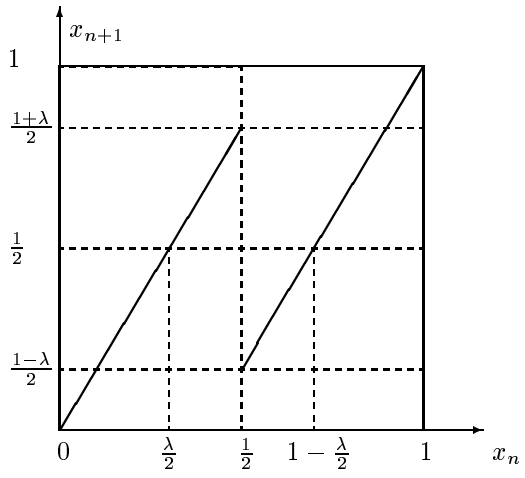


FIGURE 2. The Golden Section iteration

Figure 2 shows this transformation.

In many situations, such as the line-search algorithms of Sections 3 and 4, the dynamic system cannot be defined only in terms of $x_n$, the renormalised location of $x^*$. We shall still denote by $h_n(.)$ the updating rule for the *state* of the dynamic system, which may depend on other variables. When necessary, we shall then distinguish the dimension $d'$ of the search object $x^*$ from the dimension $d$ of the state of the dynamic system. In particular, in optimisation problems, if the objective function $f(.) = f_\theta(.)$ belongs to a finite dimensional space with parameters $\theta$, the parameters $\theta_n$ defining the renormalized function $f_n(.)$ (1) can be taken as state variables. The dynamic system then becomes time homogeneous, and $x_n$ is completely determined by $\theta_n$, see for example the ellipsoid algorithm of Section 5. Note that the condition of finite dimension for $f(.)$ is not necessary, see the dynamic process (3) induced by the Golden Section algorithm for a symmetric function. Also note that there are usually many possible choices for the state variables, and this choice is crucial for the complexity of the study of the associated dynamic system.

In several important cases the regions $\mathcal{R}_n$ are larger than the region in which $x^*$ is known to lie, the true consistent region, which may be difficult to renormalise easily. This trades a lower efficiency in the localisation of $x^*$ with the benefit of a simpler renormalisation rule, or updating rule for the original process. This is true for an important example in optimisation namely the *ellipsoid algorithm* of linear and convex programming, which we address is Section 5. In its simplest form, the so-called central-cut case, every $\mathcal{R}_n$ is a

$d'$-dimensional ellipsoid. The ellipsoid $\mathcal{R}_{n+1}$ is constructed from the ellipsoid $\mathcal{R}_n$ as the minimal volume ellipsoid which contains half of $\mathcal{R}_n$. The half is obtained by cutting $\mathcal{R}_n$ through its centre with a $(d'-1)$-dimensional hyperplane. Both the orientation of the hyperplane and the choice of which half of the ellipsoid depend on details from the iteration step. In the linear programming case they depend on the orientation of the constraints or the objective function, and in the convex programming case on information about the gradient. It is straightforward to renormalise an ellipsoid to a unit sphere, which is our standard region $\mathcal{R}_0$. Another example is provided by the *steepest descent* algorithm in Section 6. Here we again normalise $\mathcal{R}_n$ to a unit sphere, but so that the optimising point is fixed at the centre and its current approximation is on the boundary. Renormalisation by a cube in linear programming is considered in Section 7.

## 2. Ergodic theory, Lyapunov exponents and rates of convergence

The advantage of setting up the problem as a dynamic process is that it opens up the possibility of using the very considerable machinery of *ergodic theory* and *chaos*. Conversely search and optimisation may provide an operationally useful area of application of dynamic systems with perhaps a different flavour to physical, biological and economic applications. But we need to prove our point. The first issue is to link the rate of convergence of the algorithm with the characteristics of the dynamic process. A long term objective of the theory is to use this link to improve rates of convergence. Consider the mapping $h_n(.)$ of the last section. By tracing the diagram in Figure 1 in the clockwise direction we see that

$$h_n(.) = g_{n+1}(g_n^{-1}(.)) .$$

It is clear then that since $\mathcal{R}_n \supseteq \mathcal{R}_{n+1}$ the function $h_n(.)$ *expands* intervals in $\mathcal{R}$ to reflect the *contraction* in size from $\mathcal{R}_n$ to $\mathcal{R}_{n+1}$. This expansion is related to the convergence rate of the algorithm.

We shall only consider time-homogeneous dynamic systems, for the study of which we recall some concepts of ergodic theory. We refer to Mañé (1987), Ruelle (1989), Bedford, Keane and Series (1991) or Ott (1993) for a more detailed exposition.

*Some concepts in ergodic theory.*

Consider a dynamic system

$$x_{n+1} = h(x_n), \quad n = 1, 2, \ldots \tag{4}$$

where $h : \mathcal{X} \to \mathcal{X}$ is a mapping of a compact measurable set $\mathcal{X}$ onto itself. A measure $\nu$ on $\mathcal{X}$ is *invariant* under the dynamic system (4) if

$$\nu(\mathcal{A}) = \nu(h^{-1}(\mathcal{A}))$$

for any measurable set $\mathcal{A} \subseteq \mathcal{X}$. If an invariant probability measure $\nu$ cannot be written as $\frac{1}{2}\nu_1 + \frac{1}{2}\nu_2$, where $\nu_1$ and $\nu_2$ are again invariant probability measures and $\nu_1 \neq \nu_2$, then $\nu$ is *ergodic*. This is equivalent to the condition: $h^{-1}(\mathcal{A}) = \mathcal{A}$ for some measurable set $\mathcal{A} \subseteq \mathcal{X} \Rightarrow \nu(\mathcal{A}) = 0$ or 1, see Eckmann and Ruelle (1985). If an ergodic invariant measure exists, then the dynamic system is called ergodic with respect to this measure.

Suppose an invariant probability measure $\nu$ has a density $p(x)$. For instance, existence of an invariant measure with such a density is guaranteed by the uniform hyperbolicity of the map, see e.g. the midpoint algorithm in Section 3. Then one can write the following equation which expresses invariance with respect to the movement of the process forward in time:

$$p(x) = \int p(y)\delta[x - h(y)]dy\,,$$

where $\delta(.)$ is the Dirac delta-function. This equation is called the *Frobenius-Perron equation*. In many cases the following differential form of the Frobenius-Perron equation is more useful:

$$p(x) = \sum_k \frac{p(y_k)}{|\det Dh(y_k)|}\,, \tag{5}$$

where $\{y_k\} = h^{-1}(x)$ and where $Dh$ is the Jacobian matrix of $h$ with entries

$$(Dh(x))_{ij} = \partial h_i(x)/\partial x_j\,,$$

where $h_i(x)$ here denotes the $i$th component of $h(x)$. Birkhoff's ergodic theorem states that if (4) is an ergodic dynamic system, with $\nu$ a finite ergodic invariant measure, and $f \in L^1(\mathcal{X}, \nu)$, then

$$\lim_{N\to\infty} \frac{1}{N} \sum_{n=1}^{N} f(x_n) = \int_{\mathcal{X}} f(x)\nu(dx)$$

for $\nu$-almost all initial points $x_1 \in \mathcal{X}$.

A dynamic system will be called chaotic if it is ergodic and two orbits $\{x_n\}$ and $\{x'_n\}$ starting at close points $x_1 \simeq x'_1$ exponentially diverge. The rate of the divergence is measured by the so called Lyapunov exponents.

*Lyapunov exponents.*

In the one-dimensional case the Lyapunov exponent of an ergodic system $x_{n+1} = h(x_n)$ is defined as

$$\Lambda = \lim_{N\to\infty} \frac{1}{N} \sum_{n=1}^{N} \log|h'(x_n)|\,,$$

where the limit exists for $\nu$-almost all initial points $x_1$ and $\Lambda = \int \log|h'(x)|d\nu(x)$.

In short, the Lyapunov exponents for a $d$-dimensional map $h: \mathcal{X} \to \mathcal{X}$ can be defined as follows. Let $x_1 \in \mathcal{X}$ be an initial point in $\mathcal{X}$, $\{x_i\}$ the

corresponding orbit and $u_1 \in R^d$, $||u_1|| = 1$ be an arbitrary vector of unit length. If we consider an infinitesimal displacement from $x_1$ in the direction of a tangent vector $u_1$, then the evolution of the tangent vector, given by $u_{n+1} = Dh(x_n)u_n$, determines the evolution of the infinitesimal displacement of the orbit from the unperturbed point $x_n$.This gives $u_{n+1} = Dh^n(x_1)u_1$, where

$$Dh^n(x_1) = Dh(x_{n-1})\, Dh(x_{n-2}) \ldots Dh(x_1).$$

The Lyapunov exponent for initial condition $x_1$ and initial orientation $u_1$, $||u_1|| = 1$, is defined as

$$\Lambda(x_1, u_1) = \lim_{n\to\infty} \frac{1}{n} \log(||u_{n+1}||/||u_1||) = \lim_{n\to\infty} \frac{1}{n} \log(||Dh^n(x_1)\, u_1||). \qquad (6)$$

There are $d$ or less different Lyapunov exponents for a given $x_1$, and which one of them applies depends on the initial orientation $u_1$.

If the Jacobian matrices $Dh^n(x_1)$ have real eigenvalues for all $n$ and the limits in (6) exist, then the $d$ Lyapunov exponents corresponding to the point $x_1$ can be computed as

$$\Lambda_i(x_1) = \lim_{n\to\infty} \frac{1}{n} \log(\mu_{i,n}(x_1)), \quad , i = 1, \ldots, d\,, \qquad (7)$$

where $\mu_{1,n}(x_1) \geq \ldots \geq \mu_{d,n}(x_1)$ are the absolute values of the eigenvalues of $Dh^n(x_1)$ taken in decreasing order.

Ergodic theory (Oseledec's theorem) guarantees the existence of the limits used in the definition of the Lyapunov exponents under very general conditions. In particular, if $\{x_n\}$ is an ergodic sequence with $\nu$ as an invariant measure, then the Lyapunov exponents $\Lambda_i(x_1)$ are the same for $\nu$-almost every $x_1$. In this case the Lyapunov exponents can be denoted $\Lambda_1, \ldots, \Lambda_d$, with $\Lambda_1 \geq \ldots \geq \Lambda_d$.

An ergodic system is chaotic if the largest Lyapunov exponent is positive. Some dynamic systems asymptotically approach fractal attractors. There are several definitions of fractal dimensions. We only quote one of them. Let $K$ be the largest integer such that $\sum_{j=1}^{K} \Lambda_j \geq 0$, then the Lyapunov dimension is defined as

$$D_L = K + \frac{1}{|\Lambda_{K+1}|} \sum_{j=1}^{K} \Lambda_j\,.$$

*Ergodic convergence rate.*

Consider a one-dimensional algorithm for an unknown target $x^*$ in $\mathcal{R} = [A, B]$, and define $\mathcal{R}_0 = [0, 1], \mathcal{R}_1 = [A_1, B_1] \supseteq \mathcal{R}$. The length $L_n$ of the current uncertainty interval $\mathcal{R}_n = [A_n, B_n]$ containing $x^*$ is $L_n = \text{length}(\mathcal{R}_n) = B_n - A_n$. This obviously depends on the location of $x^*$ in $\mathcal{R}$, and thus on $x_1$, i.e. $L_n = L_n(x_1)$. We define the reduction or *convergence* rate of the $n$-th iteration as

$$r_n = r_n(x_1) = \frac{\text{length}(\mathcal{R}_{n+1})}{\text{length}(\mathcal{R}_n)} = \frac{A_{n+1} - B_{n+1}}{A_n - B_n}\,, \quad n \geq 1\,, \tag{8}$$

with $r_0 = (A_1 - B_1)/(A - B)$. We also introduce

$$r = r(x_1) = \lim_{N\to\infty} (L_N(x_1))^{\frac{1}{N}} = \lim_{N\to\infty} \left( L \prod_{n=0}^{N-1} r_n(x_1) \right)^{\frac{1}{N}}\,,$$

with $L = B - A$. If this limit exists and is the same for almost all $x_1$ with respect to the Lebesgue measure, then $r$ will be called the *ergodic convergence rate*. Since $L < \infty$ this becomes

$$r = \lim_{N\to\infty} (\prod_{n=1}^{N} r_n)^{\frac{1}{N}}\,,$$

and the logarithmic form is

$$\rho = -\log r = -\lim_{N\to\infty} \frac{1}{N} \sum_{n=1}^{N} \log r_n\,. \tag{9}$$

Note that $x_1$ is obtained from $x^*$ through the construction of $\mathcal{R}_1$, which can be made such that $x_1$ is in a set of full measure. The definition (8) of the reduction rate can be generalized to multidimensional algorithms, using the volumes of regions $\mathcal{R}_n$, i.e. we define in this case

$$r_n = \frac{\text{vol } \mathcal{R}_{n+1}}{\text{vol } \mathcal{R}_n}\,, \quad n \geq 1\,,$$

and $r_0 = \text{vol } \mathcal{R}_1/\text{vol } \mathcal{R}$.

An important question concerns the relation between the ergodic log-rate $\rho$ and the Lyapunov exponents $\Lambda_i$. Consider the situation where the renormalisation is *affine* with respect to $x^*$, that is

$$x_n = g_n(x^*) = \Omega_n x^* + \omega_n\,, \tag{10}$$

where the full-rank matrix $\Omega_n$ and the vector $\omega_n$ may depend on some other state variables $z_n$ of the dynamic process, not depending explicitely on $x^*$. Then vol $\mathcal{R}_n = \text{vol } \mathcal{R}_0/|\det\, \Omega_n|$, and therefore

$$r_n = \frac{|\det\, \Omega_n|}{|\det\, \Omega_{n+1}|}\,, \quad n \geq 1\,.$$

The dynamic process is $(x_{n+1}, z_{n+1}) = h(x_n, z_n)$, with

$$x_{n+1} = \Omega_{n+1}\Omega_n^{-1} x_n - \Omega_{n+1}\Omega_n^{-1}\omega_n + \omega_{n+1}\,,$$

and $z_{n+1}$ not depending explicitly on $x_n$. The Jacobian matrix of the transformation can then be written in the block-triangular form

$$Dh(x_n, z_n) = \begin{pmatrix} \Omega_{n+1}\Omega_n^{-1} & \Sigma'_n \\ 0 & \Sigma_n \end{pmatrix} ,$$

where $\Sigma_n$ and $\Sigma'_n$ are some matrices. We thus get

$$Dh^n(x_1, z_1) = \begin{pmatrix} \Omega_{n+1}\Omega_1^{-1} & \Sigma''_n \\ 0 & \prod_{i=1}^n \Sigma_i \end{pmatrix} ,$$

where $\Sigma''_n$ is again some matrix. Provided these Jacobian matrices have real eigenvalues for all $n$ and the limits in (7) exist and are the same for almost all starting points $(x_1, z_1)$, the log-rate $\rho$ is equal to the sum of $d'$ Lyapunov exponents of $h(.)$, with $d' = \dim\, x$. Indeed, the Lyapunov exponents associated with the $d'$ components of $x$ are

$$\Lambda'_j = \lim_{n\to\infty} \frac{1}{n} \log \mu'_{j,n}(x_1, z_1) ,$$

where the $\mu'_{j,n}(x_1, z_1)$ are the absolute values of the eigenvalues of $\Omega_{n+1}\Omega_1^{-1}$, taken in decreasing order. We thus obtain

$$\begin{aligned} \sum_{j=1}^{d'} \Lambda'_j &= \lim_{n\to\infty} \frac{1}{n} \log |\det\, \Omega_{n+1}\Omega_1^{-1}| \\ &= \lim_{n\to\infty} \frac{1}{n} \log \left( \prod_{i=1}^n r_i^{-1} \right) \\ &= -\lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^n \log r_i = \rho . \end{aligned}$$

This explains why, in the case $d' = 1$, and in particular for the line search algorithms of Sections 3 and 4, $\rho$ is one of the Lyapunov exponents of the dynamic system (usually the largest).

*Examples.*

The invariant measure for the bisection method is uniform on $[0, 1]$. The rate of the reduction of the uncertainty interval is always $\frac{1}{2}$. Both the log-rate and the Lyapunov exponent are equal to $\log 2$.

For the Golden Section algorithm the invariant density $p(x)$ can be easily computed and is shown in Figure 3. The rate $r_n$ is constant and equals $\lambda \simeq 0.6180$. The log-rate and Lyapunov exponents both equal $\log(1+\lambda)$. Some other second-order line search algorithms are presented in the next two sections.

Consider now the Gauss-map transformation $h(x) = \{\frac{1}{x}\}, x \in [0, 1]$, where $\{x\}$ denotes the fractional part of $x$, which corresponds to continued fraction expansion,

$$x_0 = \cfrac{1}{a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cdots}}} ,$$

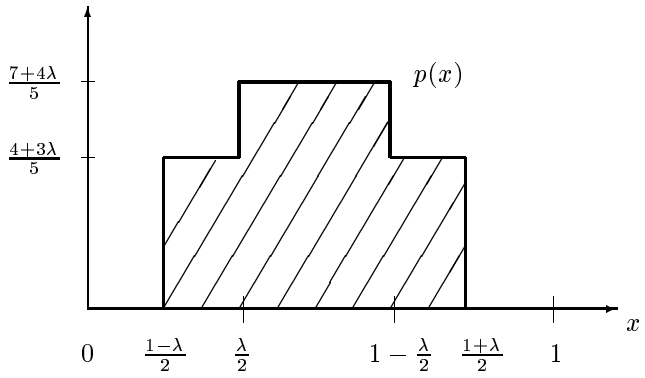


FIGURE 3. The graph of the invariant density for the Golden Section. algorithm

i.e. $x_n = \frac{1}{a_n + x_{n+1}}$, with $a_n = k$ if $x_n \in (\frac{1}{k+1}, \frac{1}{k}]$. The invariant density of the process is given by $p(x) = \frac{1}{(1+x)\log 2}$, $x \in [0,1]$. The Lyapunov exponent is

$$\Lambda = \int_0^1 (-2\log x) p(x) dx = \frac{\pi^2}{6\log 2} \simeq 2.37314$$

The mapping $h(.)$ being here nonlinear, $\Lambda$ does not coincide with the ergodic log-rate $\rho$, where at iteration $n$

$$r_n = \frac{1}{k} - \frac{1}{k+1} = \frac{1}{k(k+1)} \text{ when } \frac{1}{k+1} < x_n \leq \frac{1}{k}.$$

This gives $\rho = \sum_{k=1}^{\infty} p_k \log(k(k+1)) \simeq 2.3976$, where

$$p_k = \int_{1/(k+1)}^{1/k} p(x) dx = \frac{2\log(k+1) - \log k - \log(k+2)}{\log 2}.$$

## 3. FURTHER LINE SEARCH ALGORITHMS

We consider two of the algorithms discussed in Wynn and Zhigljavsky (1993) for the minimisation of a uni-extremal function $f(.)$, and use the notation $e_n, e'_n$ of Section 1, with $\mathcal{R}_0 = [0,1]$. Both algorithms yield a special dynamic system and, under the symmetry condition for $f(.)$, that is $f(x^* - \delta) = f(x^* + \delta)$ for any $\delta$, are faster than the Golden Section algorithm in the ergodic sense. This symmetry condition makes the corresponding dynamic system time-homogeneous.

*Midpoint algorithm* The midpoint algorithm always places the new point at $e'_n = \frac{1}{2}$ and for symmetric case yields the dynamic system:

$$x_{n+1} = \begin{cases} 2x_n & \text{if } x_n < c_n, e_n < \frac{1}{2} \\ \frac{x_n}{2e_n} & \text{if } x_n < c_n, e_n \geq \frac{1}{2} \\ 2x_n - 1 & \text{if } x_n \geq c_n, e_n \geq \frac{1}{2} \\ \frac{x_n - e_n}{1 - e_n} & \text{if } x_n \geq c_n, e_n < \frac{1}{2} \end{cases}$$

$$e_{n+1} = \begin{cases} 2e_n & \text{if } x_n < c_n, e_n < \frac{1}{2} \\ \frac{1}{2e_n} & \text{if } x_n < c_n, e_n \geq \frac{1}{2} \\ 2e_n - 1 & \text{if } x_n \geq c_n, e_n \geq \frac{1}{2} \\ \frac{\frac{1}{2} - e_n}{1 - e_n} & \text{if } x_n \geq c_n, e_n < \frac{1}{2} \end{cases}$$

where $c_n = \frac{e_n + e'_n}{2} = \frac{1}{4} + \frac{e_n}{2}$, $x_1 = x^*$ and $e_1$ is any irrational number in $[0, 1]$. Obviously the dynamic system is two-dimensional, with the renormalised target $x^*$ as one component and the renormalised location of the observation point as the other. It has the special feature that the second component $e_{n+1}$ only depends functionally on $x_n$ through the test for left or right deletion, which is in agreement with the equation (10). This implies that for any $x_n$ and $e_n$ the Jacobian matrix of the transformation $h : (x_n, e_n) \to (x_{n+1}, e_{n+1})$ is upper triangular. The same is true for some other second-order line-search algorithms. One can check that the mapping $h^2(.,.) = h(h(.,.))$ is uniformly expanding, which implies the existence of an invariant measure absolutely continuous with respect to the Lebesgue measure.

The rate sequence associated with the dynamical system $\{(x_n, e_n)\}_n$ is defined as

$$r_n = \begin{cases} \frac{1}{2} & \text{if } x_n < c_n, e_n < \frac{1}{2} \\ e_n & \text{if } x_n < c_n, e_n \geq \frac{1}{2} \\ \frac{1}{2} & \text{if } x_n \geq c_n, e_n \geq \frac{1}{2} \\ 1 - e_n & \text{if } x_n \geq c_n, e_n < \frac{1}{2} \end{cases}$$

The Jacobian matrix $Dh(x_n, e_n)$ at the point $(x_n, e_n)$ can easily be computed, see Wynn and Zhigljavsky (1993). We get for any $n$:

$$Dh(x_n, e_n) = \begin{pmatrix} \frac{1}{r_n} & u_n \\ 0 & \pm\frac{1}{2r_n^2} \end{pmatrix},$$

where $u_n$ is some number we are not interested in. This gives the following expressions for the Lyapunov exponents:

$$\Lambda_1 = -\lim_{N\to\infty} \frac{1}{N} \sum_{n=1}^{N} \log r_n = \rho \simeq 0.5365\,,$$

$$\Lambda_2 = -\lim_{N\to\infty} \frac{1}{N} \sum_{n=1}^{N} \log(2r_n^2) = 2\Lambda_1 - \log 2 \simeq 0.3799\,.$$

The numerical values were obtained by simulation as well as by numerical solution of the Frobenius-Perron equation (5). The largest Lyapunov exponent $\Lambda_1$ coincides with the log-rate $\rho = -\log r$. The ergodic rate $r \simeq 0.5848$ of the midpoint algorithm is a little better than the rate $r = \frac{\sqrt{5}-1}{2} \simeq 0.6180$ of the Golden Section algorithm. Other algorithms exist with even faster ergodic rates. *Window algorithm*

The window algorithm has a fixed width $2\epsilon$, $0 < \epsilon < \frac{1}{6}$, between $e_n$ and $e'_n$:

$$e'_n = \begin{cases} e_n + 2\epsilon & \text{if } e_n < \frac{1}{2} \\ e_n - 2\epsilon & \text{if } e_n \geq \frac{1}{2}\,, \end{cases}$$

with $e_1$ any number in $(2\epsilon, 1-2\epsilon)$, and yields the dynamic system:

$$x_{n+1} = \begin{cases} \frac{x_n}{c_n+\epsilon} & \text{if } x_n < c_n \\ \frac{x_n - c_n + \epsilon}{1-c_n+\epsilon} & \text{if } x_n \geq c_n \end{cases} \quad c_{n+1} = \begin{cases} 1 - \epsilon - \frac{2\epsilon}{c_n+\epsilon} & \text{if } x_n < c_n \\ \epsilon + \frac{2\epsilon}{1-c_n+\epsilon} & \text{if } x_n \geq c_n \end{cases}$$

where $c_n = \frac{1}{2}(e_n + e'_n)$ and $x_1 = x^*$.

The rate sequence associated with the dynamical system $\{(x_n, c_n)\}_n$ is defined as

$$r_n = \begin{cases} c_n + \epsilon & \text{if } x_n < c_n \\ 1 - c_n + \epsilon & \text{if } x_n \geq c_n\,. \end{cases}$$

As in the case of the midpoint algorithm, the Jacobian matrix of the transformation $h : (x_n, c_n) \to (x_{n+1}, c_{n+1})$ is upper triangular and can be written as, see Wynn and Zhigljavsky (1993),

$$Dh(x_n, c_n) = \begin{pmatrix} \frac{1}{r_n} & v_n \\ 0 & \frac{2\epsilon}{r_n^2} \end{pmatrix}\,,$$

where $v_n$ is some number. This gives the following expressions for the Lyapunov exponents:

$$\Lambda_1 = \rho = -\log r,\ \Lambda_2 = 2\Lambda_1 + \log(2\epsilon).$$

The largest Lyapunov exponent $\Lambda_1$ again gives the log-rate $\rho$ but the second Lyapunov exponent is now negative, which indicates that the dynamic system $\{(x_n, c_n)\}_n$ may attract to a fractal.

The Lyapunov exponents for $\epsilon = \frac{1}{16}$ determined by numerical simulations, are

$$\Lambda_1 = \rho \simeq 0.639,\ \ \Lambda_2 = 2\Lambda_1 + \log(2\epsilon) \simeq -0.801\,.$$

The ergodic rate $r$ for $\epsilon = \frac{1}{16}$ is $r \simeq 0.528$ and the Lyapunov dimension of the attractor of the dynamic system $\{(x_n, c_n)\}_n$ is $1 - \frac{\Lambda_1}{\Lambda_2} \simeq 1.798 < 2$. Numerical results show that the invariant measure for this system looks like the so called SRB measure, see Ruelle , that is a measure which is absolutely continuous

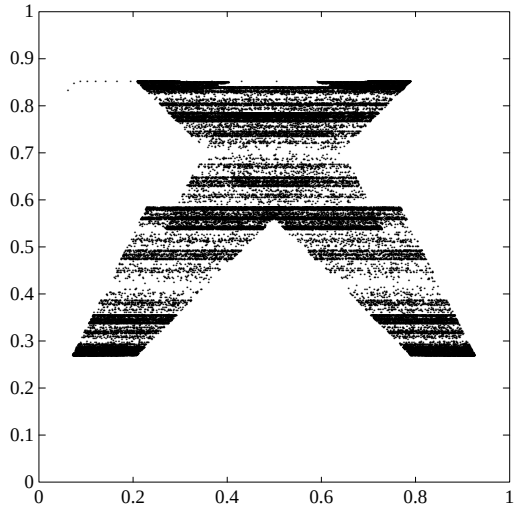


FIGURE 4. A fractal in the window algorithm: sequence of iterates $\{x_n, r_n\}$.

with respect to the Lebesgue measure along the unstable direction. Here the unstable direction coincides with the direction of the first component. Figure 4 presents $\{r_n\}$ as a function of $\{x_n\}$, and reveals the fractal structure of the attractor.

*Lower bound for the ergodic rate*

Consider the case of minimisation over $[A, B]$, when we know that $f(x)$ is symmetric about $x^*$. Let $a_n, b_n$ be as in Section 1, $c_n = \frac{a_n+b_n}{2}$, and suppose $f(a_n) > f(b_n)$. Then symmetry implies that $x_n > c_n$. We can thus achieve a greater interval reduction than in the usual case by eliminating $[0, c_n)$ rather than only $[0, a_n)$. We refer to this as the *optimistic* rule. The recurrent relation for $\{x_n\}$ and the rates $\{r_n\}$ are given by the formulas

$$x_{n+1} = \begin{cases} \frac{x_n}{c_n} & \text{if } x_n < c_n \\ \frac{x_n - c_n}{1-c_n} & \text{if } x_n \geq c_n \end{cases} \qquad r_n = \begin{cases} c_n & \text{if } x_n < c_n \\ 1 - c_n & \text{if } x_n \geq c_n \end{cases}$$

One can clearly achieve the ergodic rate $r = \frac{1}{2}$ if $r_n = \frac{1}{2}$ for all $n$. A way to do this is always to place the new observation point at one of the endpoints of the new interval obtained by the optimistic rule. An important question is: can we achieve an ergodic rate less than $\frac{1}{2}$ for all $x^*$ lying in a set of positive Lebesgue measure? The answer is *no* and we sketch the proof.

THEOREM: For any sequence $\{c_n\}, 0 < c_n < 1,\ n = 0, 1, \ldots$

$$\mu\left\{x^* \in [0,1] \mid \liminf_{N\to\infty} (L_N(x^*))^{\frac{1}{N}} < \frac{1}{2}\right\} = 0\,,$$

where $\mu$ is the Lebesgue measure and $L_N(x^*) = L_N = Lr_0r_1r_2\ldots r_{N-1}$, with $L = B - A$.

*Sketch of the proof.* Each second-order minimisation algorithm using the optimistic rule is determined by a sequence $\{c_n\}$. After $N$ iterations, the interval $K_N$ for $x^*$ for which the reduction coefficients coincide with $\{r_n\}_{n=1}^N$ has the length $L_N(x^*)$.

Let us fix $\delta,\ 0 < \delta < \frac{1}{2}$ and consider the sets

$$X_{N,\delta} = \left\{x^* \in [0,1] \mid (L_N(x^*))^{\frac{1}{N}} < \frac{1}{2} - \delta\right\}.$$

The total length of the union of the subintervals $K_N \cap X_{N,\delta}$ cannot exceed the number of disjoint intervals $K_N$ times the value of $L_N$ and is therefore bounded above by $2^N(\frac{1}{2} - \delta)^N$. Thus, applying Borel-Cantelli arguments

$$\mu\left\{x^* \in [0,1] \mid \liminf_{N\to\infty}(L_N(x^*))^{\frac{1}{N}} < \frac{1}{2} - \delta\right\}$$

$$= \mu\left\{x^* \in [0,1] \mid (L_N(x^*))^{\frac{1}{N}} < \frac{1}{2} - \delta \ \text{ infinitely often}\right\} = 0\,,$$

since for any $0 < \delta < \frac{1}{2}$

$$\sum_{N=1}^{\infty} \mu\left\{x^* \in [0,1] \mid (L_N(x^*))^{\frac{1}{N}} < \frac{1}{2} - \delta\right\} \le \sum_{N=1}^{\infty} 2^N(\frac{1}{2} - \delta)^N < \infty\,.$$

This completes the proof.

□

The statement of the theorem implies that there are no second-order minimisation algorithms with an ergodic rate smaller than $\frac{1}{2}$ for the class of symmetric functions, and therefore for any wider class of objective functions.

The authors are able to present an algorithm which achieves the optimal ergodic rate of $\frac{1}{2}$ for a class of locally smooth functions satisfying

$$f(x) = f(x^*) + (x - x^*)^2\,\frac{f''(x^*)}{2} + O(|x - x^*|^{2+\beta}) \quad \text{for some } \beta > 0\,. \quad (11)$$

The algorithm and the full proof of its convergence are very technical and can be found in Wynn and Zhigljavsky (1995). The algorithm spends most of its time behaving in the optimistic fashion described above. To avoid getting trapped, a simple test is made which leads either to the continuation of the optimistic process or to a correction involving *backtracking* to the point where a wrong judgement was made. Thus, the algorithm requires an optimistic uncertainty interval $[u_n, v_n]$ constructed according to the optimistic rule which lies inside

the ordinary (renormalised) interval $[0,1]$. When $v_n - u_n$ becomes smaller than a predefined value $\delta_n$, a check is made outside $[u_n, v_n]$ to test for a mistake. Backtracking then switches to a new correct optimistic interval. The authors can show that the parameter $\delta_n$ may be controlled in such a way that when $x^*$ is a normal number to the base 2, the number of corrections is finite and the effect of all iterations "wasted" on checking is asymptotically negligible.

## 4. More on line search: golden numbers and symbolic dynamics

*Bernoulli shift and symbolic dynamics.*

There is a well known connection between what we call in Section 1 the Bernoulli shift and the binary expansion of a number in $[0,1]$. The basic idea is that the forward transformation of the dynamic system induces a shift $\{x_n\} \longrightarrow \{x'_n\}$ on infinite sequences of 0's and 1's, with

$$x'_n = x_{n+1}\,. \tag{12}$$

This shift, also called the *Bernoulli shift*, is equivalent to the transformation (2) on $[0,1]$, and is a key point for many important ideas in ergodic theory. The corresponding dynamic system induces a measure on the sequences, which is invariant with respect to the shift (12). Moreover, the same measure can be found as an equilibrium measure, in the probabilistic sense, generated by a bi-infinite sequence of independent Bernoulli trials $\{X_n\}$, where $\text{prob}\{X_n = 1\} = \text{prob}\{X_n = 0\} = \frac{1}{2}$. Forward Bernoulli trials are easily obtained from the dynamic system (2), initialised by a random variable $x_0 = U_0$ uniformly distributed on $[0,1]$. If one chooses $X_n = 1$ when $x_n < \frac{1}{2}$ and $X_n = 0$ when $x_n \geq \frac{1}{2}$, then $\{X_n\}$ is a Bernoulli trial. The dynamic system (2) *codes* the value $x^*$ in $[0,1]$ in terms of its binary expansion, through the sequence $\{X_n\}$.

This translation of a dynamic system into a shift on an infinite sequence of symbols, and its probabilistic counterpart, is referred to as *symbolic dynamics*, and is an important tool for the study of systems. It is closely connected to *coding theory*. The "flows" in the continuous topology of the system are "coded" into sequences of symbols. This coding is most easily carried out when there is a so-called Markov partition imbedded in the original dynamic system. This is the generalisation of the partition $[0,1] = [0,\frac{1}{2}) \bigcup [\frac{1}{2},1]$ of the Bernoulli shift. In addition, the natural stochastic process is a Markov chain, possibly of high order.

*The Golden Section algorithm revisited.*

We first give the coding for the Golden Section algorithm of Section 1. Observe that the invariant measure is supported on $I = [\frac{1-\lambda}{2}, \frac{1+\lambda}{2}]$, with $\lambda = \frac{\sqrt{5}-1}{2}$, see Figure 3. We shall restrict attention only to this interval. Form the partition

$$I = I_1 \cup I_2 \cup I_3 \cup I_4\,,$$

where

| $e$ | $e'$ | $L$ | $R$ |
|---|---|---|---|
| $1-\lambda$ | $\lambda$ | $1-\lambda$ | $\lambda$ |
| $\lambda$ | $1-\lambda$ | $1-\lambda$ | $\lambda$ |

TABLE 1. The Golden Section table

$$I_1 = \left[\frac{1-\lambda}{2}, \frac{\lambda}{2}\right) \ , \ I_2 = \left[\frac{\lambda}{2}, \frac{1}{2}\right) \ , \ I_3 = \left[\frac{1}{2}, 1-\frac{\lambda}{2}\right) \ , \ I_4 = \left[1-\frac{\lambda}{2}, \frac{1+\lambda}{2}\right] \ .$$

Now if we consider an initial random variable $U_0$, with probability density given by the invariant density of Figure 3, and iterate according to the transformation (3), we obtain a 4-state Markov chain $\{X_n\}$ with

$$\{X_n = j\} \ \Leftrightarrow \ \{U_n \in I_j\} \, , \quad j = 1, \ldots, 4 \, .$$

This chain has the transition matrix

$$P = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \lambda & 1-\lambda \\ 1-\lambda & \lambda & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \, .$$

The behaviour of the Golden Section algorithm, in the case where the objective function $f(.)$ is symmetric, is completely described by the sequence of left (L) and right (R) deletions for the trajectory of the algorithm, whereas the process of the last paragraph has a four symbol "alphabet". It is straightforward to see that the 4-symbol process is equivalent to a Markov chain in which the states are pairs of successive deletions, with the equivalence

$$A \leftrightarrow LL \, , \ \ B \leftrightarrow LR \, , \ \ C \leftrightarrow RL \, , \ \ D \leftrightarrow RR \, .$$

The dynamic process (3) thus codes $x^*$ in terms of a sequence of symbols in the alphabet $\{A, B, C, D\}$. It is clear that for this Markov chain certain subsequences are prohibited ($LLL$ and $RRR$), as can also be seen by tracking the original algorithm in the symmetric case.

For that purpose, the Golden Section algorithm can be written as in Table 1. The first two columns refer to the point $e$ from previous iteration, $e = e_n$ at iteration $n$, and the control point $e'$. The last two columns refer to the point $e_{n+1}$ carried forward to the next iteration according to left and right deletions.

*Other examples with four symbols.*

The general table with four symbols is as follows:

Assume with no loss of generality that $a < b$. This implies $a' > a$, $b' < b$, $R(a) = b$, $L(b) = a$, and leaves only four possibilities. The Golden Section

| $e$ | $e'$ | $L$ | $R$ |
|---|---|---|---|
| $a$ | $a'$ | $L(a)$ | $R(a)$ |
| $b$ | $b'$ | $L(b)$ | $R(b)$ |

TABLE 2. General table with four symbols

(a)

| $e$ | $e'$ | $L$ | $R$ |
|---|---|---|---|
| $\frac{1}{2}$ | $\frac{3}{4}$ | $\frac{1}{2}$ | $\frac{2}{3}$ |
| $\frac{2}{3}$ | $\frac{1}{3}$ | $\frac{1}{2}$ | $\frac{1}{2}$ |

(b)

| $e$ | $e'$ | $L$ | $R$ |
|---|---|---|---|
| $\frac{1}{3}$ | $\frac{2}{3}$ | $\frac{1}{2}$ | $\frac{1}{2}$ |
| $\frac{1}{2}$ | $\frac{1}{4}$ | $\frac{1}{3}$ | $\frac{1}{2}$ |

(c)

| $e$ | $e'$ | $L$ | $R$ |
|---|---|---|---|
| $a$ | $a'$ | $b$ | $b$ |
| $b$ | $b'$ | $a$ | $a$ |

TABLE 3. Three "Golden" tables

algorithm corresponds to Table 1, the other three are given in Table 3(a), (b) and (c).
In Table 3(c) the numerical values are $b = \psi$,

$$\psi = \frac{1}{3}((\frac{3\sqrt{69}+11}{2})^{\frac{1}{3}} - 5(\frac{2}{3\sqrt{69}+11})^{\frac{1}{3}} + 1) \simeq 0.5698\,,$$

which is solution of the equation $(1-\psi)^2 = \psi^3$,

$$a = 1 - b \simeq 0.4302,\ a' = 1 - \psi + \psi^2 \simeq 0.7549,\ b' = 1 - a' \simeq 0.2451.$$

Following similar arguments, the authors have enumerated all tables with 3 rows, that is with 3 different carried forward points $e$, each with its own control $e'$. From symmetry considerations, the number of tables was reduced to 47, see Pronzato, Wynn and Zhigljavsky (1995).

All these dynamic processes have an invariant measure which is best expressed as an invariant measure in the "state" associated with each row together with a discrete measure across the states. These processes often, but not always, possess a finite Markov partition in the following sense. For each state there is a partition of $[0,1]$ which maps under left or right deletion to a member of the partition for another state. For each state, the set of partition members thus forms a closed set of subintervals. Then, as for the Golden Section algorithm, there is a Markov chain associated with the partitions, and a corresponding finite set of symbols. These symbols are thus related to the evolution of the process, and characterize both the current and future states.

Consider Table 3(a). First, the conditional invariant densities for states 1 and 2, multiplied by the probability of being in the state, are respectively, see Pronzato, Wynn and Zhigljavsky (1995):

$$p_1(x) = \frac{8}{7} I_{[\frac{1}{4},\frac{3}{4}]}(x)\ ,\quad p_2(x) = \frac{6}{7} I_{[\frac{1}{3},\frac{5}{6}]}(x)\,,$$

with $\int_0^1 (p_1(x) + p_2(x))dx = 1$. The associated Markov chain has 4 symbols $A, B, C$ and $D$, two for each state, given by

$$A : \text{state 1}, \left[\frac{1}{4}, \frac{5}{8}\right)\ ,\quad B : \text{state 1}, \left[\frac{5}{8}, \frac{3}{4}\right]\ ,$$

$$C : \text{state 2}, \left[\frac{1}{3}, \frac{1}{2}\right)\ ,\quad D : \text{state 2}, \left[\frac{1}{2}, \frac{5}{6}\right]\ .$$

The transition matrix is

$$P = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{2}{3} & \frac{1}{3} \\ \frac{1}{4} & \frac{3}{4} & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 \end{bmatrix}\ .$$

To the symbols $A, B, C, D$ one can respectively associate the symbols $L_1, R_1$, $L_2, R_2$, where $L_i, R_i$ refer to left/right deletion in state $i = 1, 2$.

The ergodic log-rate for the process can be computed from the chain:

$$\rho = -\sum_j p_j \log r_j\ ,$$

where the $r_j$'s are the rates associated with the symbols, and the $\{p_j\}$'s correspond to the equilibrium distribution for the symbols, given by the eigenvector of $P^T$ with associated eigenvalue 1. For the Golden Section algorithm,

$$(p_1, p_2, p_3, p_4) = \left(\frac{1}{5} - \frac{\lambda}{10}, \frac{3}{10} + \frac{\lambda}{10}, \frac{3}{10} + \frac{\lambda}{10}, \frac{1}{5} - \frac{\lambda}{10}\right)\ ,$$

and the log-rate is $\log(1 + \lambda)$. For Example (a) in Table 3, $(p_1, p_2, p_3, p_4) = (\frac{1}{7}, \frac{3}{7}, \frac{2}{7}, \frac{1}{7})$, with rates $(r_1, r_2, r_3, r_4) = (\frac{1}{2}, \frac{3}{4}, \frac{2}{3}, \frac{2}{3})$. The ergodic log-rate is $\frac{4}{7}\log 2$.

We complete this section with a more difficult example which concerns a family of algorithms with ergodic rate $r$ arbitrarily close to $\frac{1}{2}$.

*$\epsilon$-optimal algorithms with finite number of states.*

Consider the second-order line search algorithms defined by Table 4, where $k = 1, 2, \ldots, m-1$, with $m \geq 2$ a fixed integer, a parameter of the algorithm, and

$$u_i = 1 - v_i,\quad v_i = \frac{1}{1 + 2^{i-1}}\quad \text{for } i = 1, \ldots, m.$$

(a)

| $e$ | $e^{'}$ | $L$ | $R$ |
|---|---|---|---|
| $v_k$ | $\frac{1}{2}v_k$ | $v_{k+1}$ | $u_1$ |
| $v_m$ | $\frac{1}{2}(1+v_m)$ | $v_1$ | $v_{m-1}$ |

(b)

| $e$ | $e^{'}$ | $L'$ | $R'$ |
|---|---|---|---|
| $u_k$ | $\frac{1}{2}(1+u_k)$ | $v_1$ | $u_{k+1}$ |
| $u_m$ | $\frac{1}{2}u_m$ | $u_{m-1}$ | $u_1$ |

TABLE 4. $\epsilon$-optimal algorithms

| $e$ | $e^{'}$ | $L$ | $R$ |
|---|---|---|---|
| 0 | 1 | 1 | 0 |
| 1 | 1/2 | 1 | $u_1$ |
| 0 | 1/2 | $v_1$ | 0 |

TABLE 5. Initialisation of the algorithms of Table 4

In particular, $v_1 = u_1 = \frac{1}{2}$, but we distinguish the states $v_1$ and $u_1$ since their controls $e'$ differ.

The values $v_k$ and $u_k$ have the same interpretation as in Section 3, that is they can be considered as the endpoints of the normalized optimistic interval for $x_k$.

The process can be initialised by $(e, e') = (0, 1)$. Consider the pairs of states and controls $(e, e') = (1, 1/2)$ and $(e, e') = (0, 1/2)$. We use the following table: We can easily check that this initialisation has no influence from the asymptotic point of view. For almost all $x^*$, after a finite number of iterations the process arrives at $e = u_1$ or $e = v_1$ and the rules of Table 4 then apply.

In the case of functions symmetric around $x^*$, one can compute the invariant measure for the dynamic system $\{x_n\}$ characterizing the behaviour of $x^*$ within the renormalised intervals. This computation is based on the fact that the normalized variable

$$z_n = \begin{cases} 2\frac{x_n}{e_n} - 1 & \text{if} \quad e_n = v_k,\ 1 \le k \le m \\ 2\frac{x_n - e_n}{1 - e_n} & \text{if} \quad e_n = u_k,\ 1 \le k \le m \end{cases}$$

always belongs to $[0, 1]$ and obeys the following updating rule:

$$z_{n+1} = \begin{cases} 2z_n & \text{if} \quad z_n \le \frac{1}{2} \text{ and } e_n \ne v_m \text{ or } u_m \\ 2z_n - 1 & \text{if} \quad z_n > \frac{1}{2} \text{ and } e_n \ne v_m \text{ or } u_m \\ z_n & \text{if } e_n = v_m \text{ or } u_m \end{cases}$$

In many respects the asymptotic behaviour of this dynamic system coincides with that of the Bernoulli shift (2).

To the left and right deletions in the state $v_i$ we respectively associate the symbols $L_i$ and $R_i$, and do the same for state $u_i$ with symbols $L_i'$ and $R_i'$. We

| symbols | transition to | transition probabilities | stationary probabilities | rates |
|---|---|---|---|---|
| $L_k\ (k \le m-2)$ | $L_{k+1},\ R_{k+1}$ | $\frac{1}{2},\ \frac{1}{2}$ | $\frac{2^{m-k-3}}{1+2^{m-1}}$ | $\frac{1+2^k}{2+2^k}$ |
| $R_k\ (k \le m-2)$ | $L'_1,\ R'_1$ | $\frac{1}{2},\ \frac{1}{2}$ | $\frac{2^{m-k-3}}{1+2^{m-1}}$ | $\frac{1}{1+2^{k-1}}$ |
| $L_{m-1}$ | $R_m$ | 1 | $\frac{1}{2+2^m}$ | $\frac{1+2^{m-1}}{2+2^{m-1}}$ |
| $R_{m-1}$ | $L'_1,\ R'_1$ | $\frac{1}{2},\ \frac{1}{2}$ | $\frac{1}{2+2^m}$ | $\frac{1}{1+2^{m-2}}$ |
| $L_m$ | $L_1,\ R_1$ | $\frac{1}{2},\ \frac{1}{2}$ | 0 | $\frac{2^{m-1}}{1+2^{m-1}}$ |
| $R_m$ | $L_{m-1},\ R_{m-1}$ | $\frac{1}{2},\ \frac{1}{2}$ | $\frac{1}{2+2^m}$ | $\frac{1+2^{m-2}}{1+2^{m-1}}$ |
| $R'_k\ (k \le m-2)$ | $R'_{k+1},\ L'_{k+1}$ | $\frac{1}{2},\ \frac{1}{2}$ | $\frac{2^{m-k-3}}{1+2^{m-1}}$ | $\frac{1+2^k}{2+2^k}$ |
| $L'_k\ (k \le m-2)$ | $L_1,\ R_1$ | $\frac{1}{2},\ \frac{1}{2}$ | $\frac{2^{m-k-3}}{1+2^{m-1}}$ | $\frac{1}{1+2^{k-1}}$ |
| $R'_{m-1}$ | $L'_m$ | 1 | $\frac{1}{2+2^m}$ | $\frac{1+2^{m-1}}{2+2^{m-1}}$ |
| $L'_{m-1}$ | $L_1,\ R_1$ | $\frac{1}{2},\ \frac{1}{2}$ | $\frac{1}{2+2^m}$ | $\frac{1}{1+2^{m-2}}$ |
| $L'_m$ | $R'_{m-1},\ L'_{m-1}$ | $\frac{1}{2},\ \frac{1}{2}$ | $\frac{1}{2+2^m}$ | $\frac{1+2^{m-2}}{1+2^{m-1}}$ |
| $R'_m$ | $L'_1,\ R'_1$ | $\frac{1}{2},\ \frac{1}{2}$ | 0 | $\frac{2^{m-1}}{1+2^{m-1}}$ |

TABLE 6. Transition matrix and stationary probabilities for the algorithms of Table 4

can then write down the transition matrix for these symbols, which is given in the first three columns of Table 6. The transition probabilities in Table 6 determine the symbolic dynamics of the chain.

To compute the stationary probabilities we define new aggregated symbols as

$$s_i = \{L_i, R_i, L'_i, R'_i\}\ \ i = 1, \ldots, m-1;\ \ s_m = \{R_m, L'_m\}\,.$$

All the symbols in $s_i$ are equiprobable. We therefore end up with a $m$ symbol Markov chain, with the following transition probabilities:

$$\Pr\{s_k \to s_1\} = \Pr\{s_k \to s_{k+1}\} = \frac{1}{2}\ \text{ for } k = 1, \ldots, m-1;$$

$$\Pr\{s_m \to s_{m-1}\} = 1\,,$$

and all other probabilities equal to zero. The invariant probabilities

$$p_i = \lim_{N\to\infty} \frac{1}{N}\{\text{number of times } t\,,\ t \le N\,,$$

$$\text{such that the process is in state } s_i\}$$

can be easily computed:

$$p_i = \frac{2^{m-i-1}}{1+2^{m-1}} \text{ for } i = 1, \ldots, m-2,\ \ p_{m-1} = \frac{2}{1+2^{m-1}}\ \ p_m = \frac{1}{1+2^{m-1}}\,.$$

This yields the ergodic probabilities in Table 6, and the following simple expression for the ergodic log-rate:

$$\rho = -\log r = \frac{2^{m-1}}{1+2^{m-1}} \log 2.$$

This formula shows how large $m$ should be in order to achieve a given approximation of the optimal log-rate $\log 2$. An important feature of the above consideration is that all the ergodic arguments remain true when the objective function $f(.)$ is locally symmetric around $x^*$, in particular if $f(.)$ satisfies condition (11). The proof is along the lines of Wynn and Zhigljavsky (1995).

## 5. Attractors in the ellipsoid algorithm

We consider the ellipsoid algorithm for a linear programming problem. We broadly follow the exposition in Bland, Goldfarb and Todd (1981). We shall use $\zeta$ to denote the variable in $d$ dimensions. At step $k$, assume that the solution is known to lie in the unit sphere $\mathcal{R}_0 = \{\zeta | \ ||\zeta|| \leq 1\}$. We specialise to the case of exactly $d$ linear constraints, whose expressions at iteration $k$ are

$$A_k \zeta \leq b_k \, ,$$

where we use the standard convention that $\leq$ means entrywise inequality. The renormalised objective function is

$$f_k(\zeta) = c_k^T \zeta \, .$$

We first describe the general version, which will be specialized later into the *central-cut* and *deep-cut* algorithms. Let the $j$-th coordinate of the vector $b_k$ be $b_{jk}$. At iteration $k$ a cut is made in $\mathcal{R}_0$ with a hyperplane which is either parallel to a constraint (constraint step) or to the objective function (objective step), according to the following rules.

*Constraint.* If $\min_j b_{jk} < 0$ and the minimum is achieved for $j = r$ then define $s_k = r$-th row of $A_k$ and $b_{rk} = r$-th component of $b_k$, $s_k$ and $b_{rk}$ correspond to the most violated constraint at $\zeta = 0$.

*Objective.* If $\min_j b_{jk} \geq 0$, no constraint is violated at $\zeta = 0$. Then take $s_k = c_k$, the current vector defining the objective function.

The hyperplane with which $\mathcal{R}_0$ is cut is:

$$\mathcal{H}_k = \{\zeta \mid s_k^T \zeta = \beta_k\}.$$

When the cut is by the objective, the function $f(.)$ is evaluated at $\zeta = 0$, and $\beta_k$ is taken equal to 0, although fixed negative values of $\beta_k$ will be considered later. This does not seem to have been done in the literature. The consistent half of the sphere is identified, that is the half where the minimising point is known to lie.

When the cut is by the constraint, we can still consider a cut through the centre of the sphere, parallel to the constraint, that is take $\beta_k = 0$. This corresponds to the *central-cut* algorithm. Another possibility is to use deeper cuts, defined by $\beta_k = b_{rk}$, which corresponds to the so-called *deep-cut* algorithm.

We shall denote by $\alpha_k = -\beta_k/\|s_k\|$ the algebraic distance from the origin to $\mathcal{H}_k$. The ellipsoid is constructed, with minimum volume, which contains the consistent part of the sphere. This ellipsoid has its centre at

$$\zeta_k = -\tau \frac{s_k}{||s_k||}$$

and is given by

$$\mathcal{E}_{k+1} = \left\{\zeta | (\zeta - \zeta_k)^T (J_k J_k^T)^{-1} (\zeta - \zeta_k) \leq 1\right\} ,$$

where

$$J_k = \delta^{\frac{1}{2}} \left( I - \pi \frac{s_k s_k^T}{||s_k||^2} \right) ,$$

with $I$ the $d$-dimensional identity matrix. The full iteration, including renormalisation back to the unit sphere, is given by

$$A_{k+1} = A_k J_k , \quad c_{k+1} = J_k c_k , \quad b_{k+1} = b_k - A_k \zeta_k .$$

The various constants are related by

$$\tau = \frac{1 + d\alpha}{1 + d} , \quad \sigma = \frac{2\tau}{1 + \alpha} , \quad \delta = \frac{(1 - \alpha^2) d^2}{d^2 - 1} , \quad \pi = 1 - \sqrt{1 - \sigma} ,$$

where, at iteration $k$, $\alpha = \alpha_k$. In the central-cut case $\alpha_k = 0$ and the constants become $\tau = \frac{1}{d+1}$, $\sigma = \frac{2}{d+1}$, $\delta = \frac{d^2}{d^2-1}$.

We make one standardisation in addition to the renormalisation just described. This is to rotate the figure so that the objective is perpendicular to the $d$-th axis with coordinate vector $e = (0, ..., 0, 1)^T$. Thus let $Q_k$ be a rotation matrix such that $Q_k^T Q_k = I$ and that

$$Q_k c_{k+1} = Q_k J_k c_k = e .$$

Then we can write

$$\tilde{\zeta} = Q_k \zeta$$

and we have the new equations

$$\tilde{A}_{k+1} = A_k J_k Q_k^T , \quad \tilde{b}_{k+1} = b_k - A_k \zeta_k , \quad \tilde{c}_{k+1} = e .$$

According to (8), we define the convergence (reduction) rate at iteration $k$ by

$$r_k = \text{vol } \mathcal{E}_{k+1} = \mid \det J_k \mid = \delta^{d/2} (1 - \pi) ,$$

which, for fixed $d$, is only function of $\alpha_k$. For the central-cut algorithm, $\alpha_k = 0$ for all $k$, and the rate is $d^d \sqrt{1 - 2/(1 + d)}/(d^2 - 1)^{d/2}$ for all $k$. For $d = 2$, one gets $r \simeq 0.7698$. In order to obtain a small rate, $\alpha_k$ should be chosen as large as possible. However, simulations show that if the depth $\alpha_k$ of the constraint cut

is fixed, the renormalised ellipsoid can be inconsistent with the data, that is it may not contain $x_k = A_k^{-1} b_k$. On the other hand, the depth for the objective cuts can be fixed, while classical deep cuts $\alpha_k = -\beta_k / \|s_k\|$ are used for the constraints.

We shall study the 2-dimensional case $d = 2$ in some detail. With the renormalisation and the additional standardisation just described the dynamic system can be written as a four dimensional process. Without loss of generality we may take

$$A = \begin{bmatrix} \cos\theta & \sin\theta \\ \cos\phi & \sin\phi \end{bmatrix}, \qquad \beta = \begin{bmatrix} \beta_1 \\ \beta_2 \end{bmatrix},$$

and consider $z = (\tan\theta, \tan\phi, \beta_1, \beta_2)$ as the four variables.

We consider the case in which $\alpha_k$ is fixed to positive constant $\alpha^0$ in the objective case and deep cut is performed in the constraint case.

In the objective case $\beta_1 \geq 0,\ \beta_2 \geq 0$ and a typical iteration step

$$z = (\tan\theta, \tan\phi, \beta_1, \beta_2) \rightarrow z' = (\tan\theta', \tan\phi', \beta_1', \beta_2')$$

is given by

$$\begin{cases} \tan\theta' &= (1-\pi)\tan\theta \\ \tan\phi' &= (1-\pi)\tan\phi \\ \beta_1' &= \frac{1}{\sqrt{\delta}} \frac{\beta_1 (1+\tan^2\theta)^{\frac{1}{2}} - \tau \tan\theta}{(1+(1-\pi)^2 \tan^2\theta)^{\frac{1}{2}}} \\ \beta_2' &= \frac{1}{\sqrt{\delta}} \frac{\beta_2 (1+\tan^2\phi)^{\frac{1}{2}} + \tau \tan\phi}{(1+(1-\pi)^2 \tan^2\phi)^{\frac{1}{2}}} \end{cases}$$

Consider the constraint case $\beta_1 < 0,\ \ \beta_2 \geq \beta_1$. Then $\alpha_k = \alpha^A = -\beta_1$ and the iteration is

$$\begin{cases} \tan\theta' &= (1-\pi)\tan\theta \\ \tan\phi' &= \frac{\tan\phi}{1-\pi}\left(1 - \pi(2-\pi)\frac{\tan^2\theta}{1+\tan^2\theta}\right) - \frac{\pi(2-\pi)}{1-\pi}\frac{\tan\theta}{1+\tan^2\theta} \\ \beta_1' &= \frac{\beta_1 + \tau}{\sqrt{\delta}(1-\pi)} \\ \beta_2' &= \frac{1}{\sqrt{\delta}} \frac{\beta_2 (1+\tan^2(\phi-\theta))^{\frac{1}{2}} + \tau \mathrm{sign}[\tan(\phi-\theta)]}{((1-\pi)^2 + \tan^2(\phi-\theta))^{\frac{1}{2}}} \end{cases}$$

For the case $\beta_1 > \beta_2,\ \ \beta_2 < 0$ we have $\alpha_k = \alpha^B = -\beta_2$ and

$$\begin{cases} \tan\theta' &= \frac{\tan\theta}{1-\pi}\left(1 - \pi(2-\pi)\frac{\tan^2\phi}{1+\tan^2\phi}\right) - \frac{\pi(2-\pi)}{1-\pi}\frac{\tan\phi}{1+\tan^2\phi} \\ \tan\phi' &= (1-\pi)\tan\phi \\ \beta_1' &= \frac{1}{\sqrt{\delta}} \frac{\beta_1 (1+\tan^2(\phi-\theta))^{\frac{1}{2}} + \tau \mathrm{sign}[\tan(\phi-\theta)]}{((1-\pi)^2 + \tan^2(\phi-\theta))^{\frac{1}{2}}} \\ \beta_2' &= \frac{\beta_2 + \tau}{\sqrt{\delta}(1-\pi)} \end{cases}$$

This algorithm exhibits a complex pattern of special periodic attractors which depends on the fixed value $\alpha^0$ chosen for the objective case, see Figure

5. For each attractor, the solution $x_k = A_k^{-1} b_k$ lives on a number of distinct concentric circles. At various critical values of $\alpha^0$ the number of points in the attractor and the radius of the concentric circles jump, see Figure 5. Between these critical points the rate improves with increasing $\alpha^0$, see Figure 6. From the point $\alpha^0 \simeq 0.225$ to $\alpha^0 = \frac{1}{2}$ there is a 3-point attractor with the rate decreasing smoothly to its minimum $1/3$ in the range $0 < \alpha^0 \leq \frac{1}{2}$. Note the improvement in the convergence rate when going from standard deep cuts, that is $\alpha^0 = 0$, to deep cuts with $\alpha^0 = 1/2$, especially compared to the marginal improvement obtained when going from central cuts ($r \simeq 0.77$) to standard deep cuts ($r \simeq 0.72$).

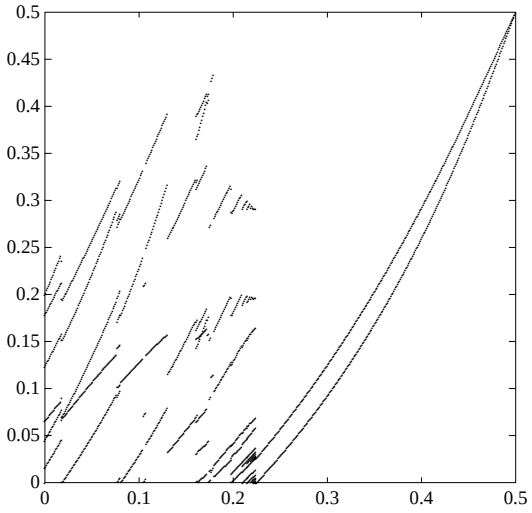


FIGURE 5. Attractors in the ellipsoid algorithm: depths $\alpha^A$ and $\alpha^B$ of constraint cuts as function of the fixed depth $\alpha^0$ of the objective cut.

After some work the authors have been able to find the attractor at $\alpha^0 = \frac{1}{2}$. It is given in Table 7 and cycles in the order $z_a \to z_b \to z_c \to z_a \ldots$

Numerical simulations show that even deep cuts can be made deeper by using $\alpha_k = -\lambda \beta_k / \|s_k\|$, with $\lambda > 1$, for constraint cuts. Figure 7 presents the renormalised sphere $\mathcal{R}_0$ and the attractor for the sequence $\{x_k = A_k^{-1} b_k\}$ in the case $\lambda = 1.121, \alpha^0 = 0.5195$. The attractor is unusual, and illustrates the difficulty of studying the properties of the limiting behaviour of the process.

Consider now the central-cut algorithm: for all iterations, objective and constraint, $\alpha = 0, \tau = \frac{1}{3}, \sigma = \frac{2}{3}, \delta = \frac{4}{3}, \pi = 1 - \frac{1}{\sqrt{3}}$. It can be shown that if we start with $\tan\phi < 0, \tan\theta > 0, \tan\phi \times \tan\theta > -1$ then for all iterations

$$\tan(\phi - \theta) < 0, \quad \tan\phi < 0, \quad \tan\theta > 0, \quad \tan\phi \times \tan\theta > -1 \,.$$

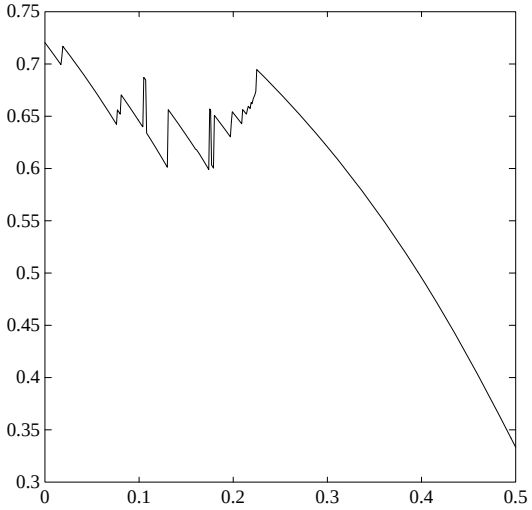


FIGURE 6. Ergodic convergence rate $r$ as a function of the fixed depth $\alpha^0$of the objective cut.

| | $\beta_1$ | $\beta_2$ | $\tan\theta$ | $\tan\phi$ |
|---|---|---|---|---|
| $z_a$ | $-\frac{1}{2}$ | $\frac{1}{2}$ | $\frac{3\sqrt{3}}{\sqrt{13}}$ | $-\frac{1}{\sqrt{3}\sqrt{13}}$ |
| $z_b$ | $\frac{1}{2}$ | $\frac{\sqrt{3}-1}{\sqrt{10}}$ | $\frac{3}{\sqrt{13}}$ | $-\frac{3\sqrt{3}}{\sqrt{13}}$ |
| $z_c$ | $\frac{\sqrt{3}-1}{\sqrt{10}}$ | $-\frac{1}{2}$ | $\frac{1}{\sqrt{3}\sqrt{13}}$ | $-\frac{3}{\sqrt{13}}$ |

TABLE 7. A three point attractor in the ellipsoid algorithm; d=2

It is possible to find the inverse of the iteration and write down the Frobenius-Perron equations, but Figure 8 shows that the invariant measure for $(\tan\theta, \tan\phi)$ lives on a fractal, indicating a totally different limiting behaviour from the deep-cut case. The numerical evaluation of the box counting dimension of the fractal of the four dimensional process, see for example Barndorff-Nielsen, Jensen and Kendall (1993), gives the estimate $\simeq 2.1$.

## 6. A FRACTAL IN THE STEEPEST DESCENT ALGORITHM

We turn our attention now to the classical steepest descent optimisation algorithm. Although this algorithm is not used much in practice, many other algorithms such as the conjugate gradient and variable metric methods being preferred, it provides an excellent framework for studying the dynamic systems

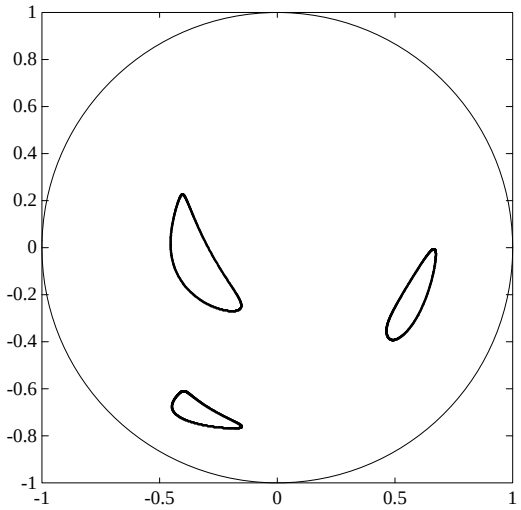


FIGURE 7. An attractor in the ellipsoid algorithm when using very deep cuts.

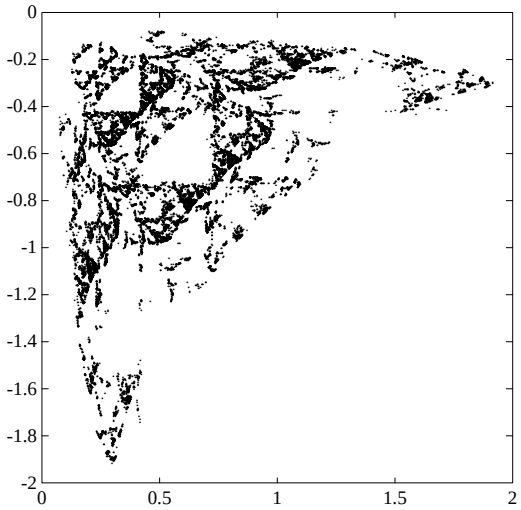


FIGURE 8. A fractal in the ellipsoid algorithm: sequence of iterates $\{\tan\theta_k, \tan\phi_k\}$.

approach. It is easy to write down the algorithm and discuss renormalisation. Even in the three-dimensional case, studied here in some detail, it is clear that much about the behaviour is unknown, and this fact can be deduced from the complexity of the basin of attraction for the normalised version of the algorithm. The following conclusions can be drawn from the study of the three dimensional case for a quadratic function.

(i) The asymptotic convergence rate of the algorithm depends on both the function and the starting point. For the quadratic case the rate depends on the largest and smallest eigenvalues of the quadratic form defining the function, a well-known result. The dependence on the starting point is extremely complex owing to the presence of a fractal-like structure.

(ii) It is conjectured, but not proven, that the renormalised algorithm attracts to a two-dimensional plane. The presence of the fractal, see Figure 9, encourages the authors to doubt whether a real proof of this is available in the literature.

(iii) Under the assumption of attraction to 2-dimensions, there is a two-point attractor whose values give precisely the rate, and the attractor depends on the initial starting point.

Let $A$ be a $d$-dimensional symmetric non-negative definite matrix and set the quadratic objective function to be

$$f(x) = x^T A x \, .$$

Without loss of generality we immediately replace $A$ by a diagonal matrix $H$ whose diagonal elements are the ordered eigenvalues of $A$. The algorithm can then be written:

$$x_{k+1} = (I - \alpha_k H) x_k \, , \tag{13}$$

where $I$ is the $d$-dimensional identity matrix. The optimum step length $\alpha_k$ for the steepest descent algorithm is chosen to minimise

$$x_k^T (I - \alpha_k H) H (I - \alpha_k H) x_k \, ,$$

and takes the value

$$\alpha_k = \frac{x_k^T H^2 x_k}{x_k^T H^3 x_k} \, .$$

We adopt here the renormalisation which keeps the minimising point at the origin $(0, ..., 0)$, and simply rescales $x_k$ by its own norm to keep the renormalised process always on the unit sphere. The convergence rate at iteration $k$ is simply the ratio of norms:

$$r_k = \frac{||x_{k+1}||}{||x_k||} \, .$$

This can also be interpreted as the inverse of the ratio of the renormalisation constants from each stage.

For the case $d = 3$ it is convenient to write the algorithm in unrenormalised form and to assume that the eigenvalues are distinct. Without any loss of generality, we then take

$$H = \begin{pmatrix} 1 & 0 & 0 \\ 0 & a & 0 \\ 0 & 0 & b \end{pmatrix},$$

with $1 < a < b$. The algorithm (13) can then be written

$$\begin{cases} x_{1,k+1} = (1 - f(x_k, a, b))x_{1k} \\ x_{2,k+1} = (1 - af(x_k, a, b))x_{2k} \\ x_{3,k+1} = (1 - bf(x_k, a, b))x_{3k}\,, \end{cases}$$

where $f(x, a, b) = \frac{x_1^2 + a^2 x_2^2 + b^2 x_3^2}{x_1^2 + a^3 x_2^2 + b^3 x_3^2}$ and where where $x_{jk}$ is the $j$-th component of $x_k$. Notice that the terms in the brackets on the right hand sides are scale invariant and therefore do not depend on the renormalisation. We adopt a change in variables which incorporates the renormalisation. Thus define

$$t = \frac{x_3}{x_1}\,, \quad z_i = x_i / ||(x_1, x_2, x_3)||\,, \quad i = 1, 2, 3\,.$$

It is useful to visualise the process as lying on a 3-dimensional unit sphere $\{z \mid \|z\| \leq 1\}$, with $z_2$ as the vertical component and $t$ as the tangent of the angle $\theta$ between the two horizontal components $(z_1, z_3)$. The vertical component is the eigen-direction for the middle eigenvalue $(= a)$ and the horizontal directions for the largest $(= b)$ and the smallest $(=1)$ eigenvalues.

Simulations show that the process attracts to the horizontal $(z_1, z_3)$ plane. In doing so it attracts to two "conjugate" points on the unit circle $z_1^2 + z_3^2 = 1$ which depend on the starting point. Figure 9 shows the projection onto the plane $(z_1, z_3)$ of the region of attraction to a small neighbourhood, radius $< 0.02$, of the point $(z_1, z_2, z_3) = (0.9606, 0, 0.2781)$, with $a = 2$ and $b = 4$. It illustrates the difficulty in studying the convergence rate as a function of the starting point: given a starting point, the convergence rate is unpredictable.

If the limit set of the algorithm is indeed the unit circle then certain facts can be established.

*Conjugate points and rates.* The conjugate points have $\tan\theta$ values $t$ and $t'$ that satisfy

$$tt' = -\frac{1}{b^2}\,.$$

If indeed the process attracts to these points, then it switches between them on alternative iterations. The asymptotic rate of the algorithm can be computed by taking both iterations into account. The rate, $r_k$, for one iteration from $t_k = x_{3k}/x_{1k}$ to $t_{k+1}$ satisfies

$$r_k^2 = \frac{x_{1,k+1}^2 + x_{3,k+1}^2}{x_{1k}^2 + x_{3k}^2}\,.$$

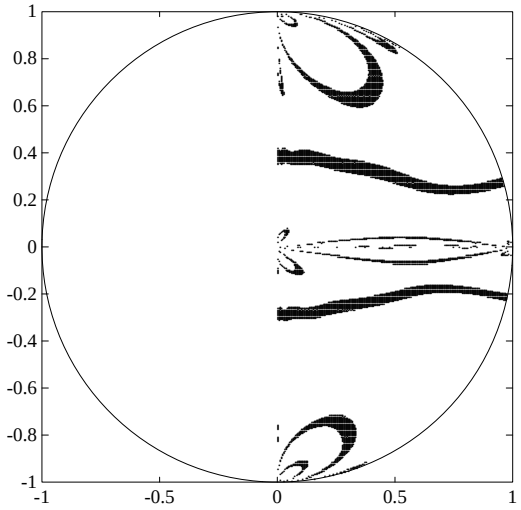


FIGURE 9. Fractal for steepest descent: projection onto the plane $(z_1, z_3)$of a region of attraction.

Then the full 2-step rate from $t_k$ to $t_{k+2}$ is

$$r^2 = r_k r_{k+1} = \frac{(1-b)^2 b t_k^2}{(1+b^3 t_k^2)(1+bt_k^2)}\,. \tag{14}$$

This achieves a maximum at $t_k = \frac{1}{b}$, giving a global worst rate (recall that smaller $r$ is better) of $\frac{b-1}{1+b}$, and clearly, then, $r_{k+1} = r_k$. This result is well known in the literature, see for example Luenberger (1973).

*A special point.* It is instructive to take one inverse iteration of the algorithm starting on the circle at some $z_{k+1}$. The solutions then satisfy either $z_{2k} = 0$, which corresponds to the conjugate point for $z_{k+1}$ on the circle, or

$$t_k^2 = \frac{a-1}{b^2(b-a)}\,. \tag{15}$$

The set of points satisfying (15) forms a pair of geodesics on the unit sphere, with a straight line projection into two-dimensions, the curve and the line intersecting the unit circle at special points satisfying (15) and $z_{2k} = 0$. One can consider this as the only "exit" route for the inverse process starting on the unit circle at points given by $z_{2,k+1} = 0$ and

$$t_{k+1}^2 = \frac{b-a}{b^2(a-1)}\,. \tag{16}$$

*A special measure.* It appears then that the algorithm attracts to points determined by the start and the largest and smallest eigenvalues of $H$. In this sense the limiting behaviour is complex but stable. Figure 10 shows the distribution induced on $\theta$, the limiting angle for the limit points on the circle, for a large random sample of starting points chosen uniformly on the unit sphere. It is seen that there are peaks at the above special points (16), indicating a preference for "entry" to a neighbourhood of the circle at these points. The other symmetrical peaks correspond to the conjugates (15) of the points (16).

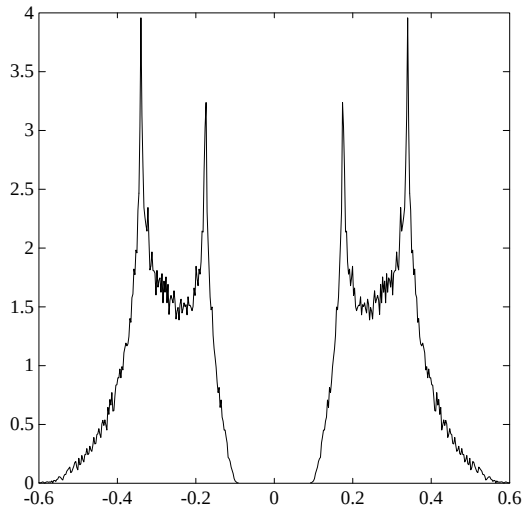


FIGURE 10. Density of $\theta$ produced by a uniform distribution of starting pointson the sphere ($d = 3$, eigenvalues $1, 2, 4$).

*Behaviour close to the circle.* Consider the situation in which $z_k$ is close to the circle ($x_2 = 0$), and thus close to one of the conjugate limit points. It follows that after two iterations, jumping to be near the conjugate point and jumping back again, $z_{k+2}$ should be close to $z_k$. We can study this behaviour by computing the Jacobian of this double transformation at $z_{2k} = 0$. In fact, it is more convenient to compute the value of the Jacobian matrix of the transformation $(t_k^2, z_{2k}^2) \to (t_{k+2}^2, z_{2,k+2}^2)$ at $z_{2k} = 0$. This is

$$J = \begin{bmatrix} 1 & J_{12} \\ 0 & J_{22} \end{bmatrix},$$

where

$$\begin{aligned} J_{12} &= \frac{2a^2(1+t^2)(1+b^2t^2)^2(a-1)(b-a)}{b^4t^2(b-1)^3}[(a-1)+b^2t^2(a-b)] \\ J_{22} &= \frac{(a-1+b^2t^2(a-b))^2(a-b+b^2t^2(a-1))^2}{b^4t^4(b-1)^4}, \end{aligned}$$

with $t = t_k$. Stability near the circle is controlled by whether, as a function of $t^2$, $J_{22}(t^2) < 1$: stable, or $J_{22}(t^2) > 1$: unstable. We always have instability as $|t| \to 0$ or $\infty$, since then $J_{22} \to \infty$. Now, $J_{22} = 0$ has roots in $t^2$ at $t_A^2 = \frac{b-a}{b^2(a-1)}$, $t_B^2 = \frac{a-1}{b^2(b-a)}$ with $t_A^2 \leq t_B^2$ if and only if $b < 2a-1$. Between $t_A$ and $t_B$, $J_{22}$ has a single local maximum at $t^2 = \frac{1}{b^2}$ with value

$$J_{22}(\frac{1}{b^2}) = \frac{(2a-b-1)^4}{(b-1)^4} = \frac{((b-a)-(a-1))^4}{((b-a)+(a-1))^4} < 1\,.$$

The stable region for $t^2$ for which $J_{22} < 1$ is the interval between $t^{*2}$ and its conjugate $\frac{1}{b^4 t^{*2}}$, where

$$t^{*2} = \frac{(a-1)^2 + (b-a)^2 + (b-a)(a-1) - \sqrt{(b-a)^2 + (a-1)^2}(b-1)}{b^2(a-1)(b-a)}\,.$$

This interval contains $\{t_A^2, t_B^2\}$ and gives the support of the density presented on Figure 10. Since $r^2 = r_k r_{k+1}$, given by (14), is minimum when $t_k \to 0$ or $t_k \to \infty$, the best asymptotic rate $r^*$, when the starting point is not on the circle $z_2 = 0$, is obtained for $t_k^2 = t^{*2}$ or $t_k^2 = \frac{1}{b^4 t^{*2}}$. Note that $r^*$ thus depends on $a$, although the limiting behaviour is in the plane $z_2 = 0$. It tends to zero when $a$ tends to 1 or $b$, and its largest (worst) value $(b-1)/\sqrt{b^2+6b+1}$ is achieved for $a = (b+1)/2$ and $t^* = (\sqrt{2}-1)/b$.

A similar study can be made for the circle $x_3 = 0$. One has then $J_{22} > 1$ for all $t^2$ and the circle $x_3 = 0$ is therefore locally unstable.

*Steepest descent with relaxation.* The introduction of a relaxation coefficient in the steepest descent algorithm, that is

$$x_{k+1} = (I - \gamma \frac{x_k^T H^2 x_k}{x_k^T H^3 x_k} H) x_k\,, \quad 0 < \gamma < 1\,,$$

totally changes the asymptotic behaviour of the algorithm. For fixed $H$, depending on the value of $\gamma$, the renormalised process attracts to periodic orbits independent of the starting point or exhibits a chaotic behaviour. Figure 11 presents the (classical) period doubling phenomenon for $d = 2$ and $H = \begin{pmatrix} 1 & 0 \\ 0 & b \end{pmatrix}$: the attractor of the sequence $\{|\theta_k| = |\text{arc tan}(x_{2k}/x_{1k})|\}$ is plotted as a function of $\gamma$ when $b = 4$. One can easily show that the first period-doubling value of $\gamma$ is $4b/(b+1)^2$. Using a relaxation coefficient $\gamma$ close to 1 is thus of special importance, since it allows us to avoid meeting the worst ergodic convergence rate, which is always possible when $\gamma = 1$.

## 7. A "CUBIC" ALGORITHM FOR LINEAR PROGRAMMING

While the natural region for renormalisation in the ellipsoidal algorithm was a sphere, other regions can be used, e.g. cubes, leading to a quite different behaviour. We consider the linear programming problem, and for the sake of simplicity we restrict our attention to dimension $d = 2$.

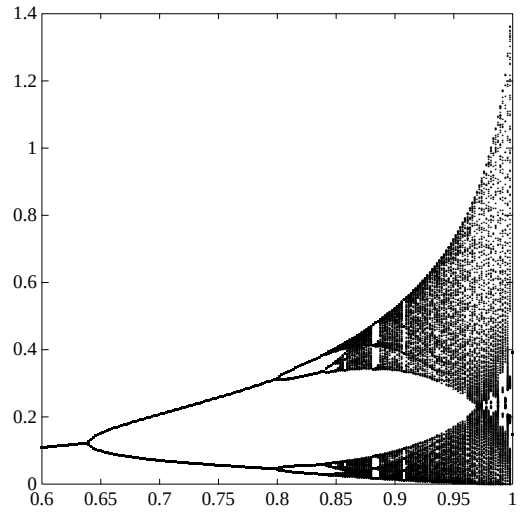


FIGURE 11. The period-doubling phenomenon for steepest descent with relaxation.

*Deep-cut algorithm*

At step $k$, assume that the solution is known to lie in the unit square $\mathcal{R}_0 = \{\zeta \mid \|\zeta\|_\infty \leq 1\}$ of the $(x, y)$ plane. The orientation of the square is chosen such that the objective is $f(\zeta) = c^T\zeta$, with $c^T = (0\ \ 1)$. The constraints are $y \geq a_k x + b_k$ and $y \geq c_k x + d_k$.

We consider central cuts for the objective case: when $b_k < 0$ and $d_k < 0$, cancel the upper half of the square, see Figure 12, and renormalise back to the unit square. Central cuts for the constraint case, parallel to the constraint and passing through the origin, do not yield a convergent algorithm. We thus consider deep cuts, that is cancel the part under the most violated constraint at the origin, where the optimum cannot lie, see Figure 12, and renormalise back to the unit square.

One can easily show that the ratio $c_k/a_k$ remains constant, and we shall denote its value by $\tau$. When $\tau = 1 - 2^k$ or $1/\tau = 1 - 2^k$, the convergence of the unrenormalised process is finite, and we shall thus exclude this situation. The updating equations are then as follows, with $r_k$ the reduction rate in terms of volume:

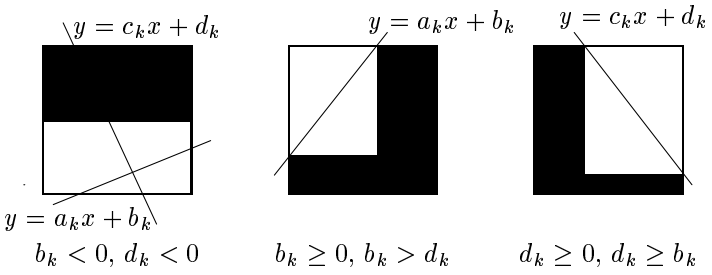


FIGURE 12. The "cubic" algorithm with deep cuts for $d = 2$

if $b_k < 0,\ d_k < 0$
$a_{k+1} = 2a_k,\ b_{k+1} = 2b_k + 1,\ d_{k+1} = 2d_k + 1,\ r_k = \frac{1}{2}$,

if $b_k \geq 0,\ b_k > d_k$
$a_{k+1} = \frac{1+\alpha}{1-\beta} a_k,\ b_{k+1} = \frac{2b_k + a_k(\alpha-1) - (\beta+1)}{1-\beta}$,
$d_{k+1} = \frac{2d_k + \tau(\alpha-1) - (\beta+1)}{1-\beta},\ r_k = \frac{(1+\alpha)(1-\beta)}{4}$,
where $\alpha = \min(\frac{1-b_k}{a_k}, 1),\ \beta = \max(b_k - a_k, -1)$

if $d_k \geq 0,\ d_k \geq b_k$
$a_{k+1} = \frac{1-\alpha}{1-\beta} a_k,\ b_{k+1} = \frac{2b_k + a_k(\alpha+1) - (\beta+1)}{1-\beta}$,
$d_{k+1} = \frac{2d_k + \tau a_k(\alpha+1) - (\beta+1)}{1-\beta},\ r_k = \frac{(1-\alpha)(1-\beta)}{4}$,
where $\alpha = \max(\frac{1-d_k}{\tau a_k}, -1),\ \beta = \max(\tau a_k + d_k, -1)$.

One can show that after a finite number of iterations the behaviour of the dynamic system $x_k = (a_k, b_k, d_k) \longrightarrow x_{k+1} = (a_{k+1}, b_{k+1}, d_{k+1})$ is periodic. We assume first that $\tau < -1$ and define $k_\tau$ by

$$1 - 2^{k_\tau + 1} < \tau < 1 - 2^{k_\tau}\,.$$

We also define the following states

$$s_j = \begin{pmatrix} 1 \\ 0 \\ \tau + 2^{j+1} - 1 \end{pmatrix},\ j = 0, \ldots, k_\tau\,,$$

$$s'_j = \begin{pmatrix} 2 \\ 1 \\ 2\tau + 2^{j+1} - 1 \end{pmatrix},\ j = 1, \ldots, k_\tau\,,$$

$$z = \begin{pmatrix} \frac{\tau + 2^{k_\tau} - 1}{\tau} \\ \frac{1 - 2^{k_\tau}}{\tau} \\ \tau + 2^{k_\tau} \end{pmatrix},\ z' = \begin{pmatrix} -\frac{1}{\tau} \\ \frac{\tau+1}{\tau} \\ 0 \end{pmatrix}.$$

The behaviour of the process is then completely described by

$$s_j \xrightarrow{1/2} s'_{j+1} \xrightarrow{1/2} s_{j+1},\ j = 0, \ldots, k_\tau - 1\,,$$

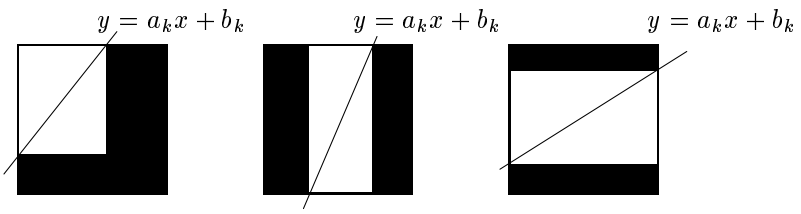


FIGURE 13. The "cubic" algorithm with deep cuts above and under the constraints for $d = 2$

with reduction rate $\frac{1}{2}$ for each transition,

$$s_{k_\tau} \xrightarrow{r_1} z \xrightarrow{r_2} s_0\,,$$

with $r_1 = 1 - \frac{1-2^{k_\tau}}{\tau}$, $r_2 = 1 + \frac{2^{k_\tau}-1}{\tau}$, if $1 - 2^{k_\tau+1} < \tau < -2^{k_\tau}$, and

$$s_{k_\tau} \xrightarrow{r'_1} z' \xrightarrow{r'_2} s_0\,,$$

with $r'_1 = (1-\tau-2^{k_\tau})(1-\frac{1-2^{k_\tau}}{\tau})$, $r_2 = -\frac{1}{\tau}$, if $-2^{k_\tau} \leq \tau < 1-2^{k_\tau}$. The period is $2k_\tau + 2$, and the ergodic log-rate is

$$\rho = \frac{k_\tau}{k_\tau+1}\log 2 - \frac{1}{k_\tau+1}\log(1 - \frac{1-2^{k_\tau}}{\tau})\,.$$

The log-rate $\rho$ is infinite for $\tau = 1-2^k$, which corresponds to finite convergence for the original unnormalised process. The same study is valid for $-1 < \tau < 0$, and the same results hold with $\tau$ replaced by $\frac{1}{\tau}$. We have $\inf_\tau \rho = \frac{\log 3}{2}$.

*Double cuts*

The reduction rate of the algorithm can be improved by considering deep cuts above and under the constraints, as indicated in Figure 13 for the constraint $y \geq a_k x + b_k$.

When $\tau = 1 - 2^k$ or $1/\tau = 1 - 2^k$ the convergence is finite, and we exclude this situation. For other values of $\tau$, the behaviour is again periodic, but with a more complex structure than for ordinary deep cuts. Consider the case $\tau < -1$. One can show that the process passes through states of the form $(1, 0, d)$, with $d < 0$. We then have the following transitions, with reduction rates indicated above the arrows:

$$\begin{pmatrix} 1 \\ 0 \\ d \end{pmatrix} \overset{1/2}{\to} \begin{pmatrix} 2 \\ 1 \\ f_1(d)+\tau \end{pmatrix} \overset{1/2}{\to} \begin{pmatrix} 1 \\ 0 \\ f_1(d) \end{pmatrix} \overset{1/2}{\to} \dots$$

$$\ldots \overset{1/2}{\to} \begin{pmatrix} 2 \\ 1 \\ f_{j^*}(d)+\tau \end{pmatrix} \overset{1/2}{\to} \begin{pmatrix} 1 \\ 0 \\ f_{j^*}(d) \end{pmatrix},$$

with

$$f_j(d) = 2^j d + (2^j - 1)(1-\tau)$$

and $j^* = j^*(d)$ such that $f_{j^*-1}(d) \le 0 < f_{j^*}(d)$. Then, if $f_{j^*}(d) > -(\tau+1)$

$$\begin{pmatrix} 1 \\ 0 \\ f_{j^*}(d) \end{pmatrix} \overset{-(1-\tau-f_{j^*}(d))/(4\tau)}{\to} \begin{pmatrix} -\frac{1}{\tau} \\ \frac{\tau+1}{\tau} \\ 0 \end{pmatrix} \overset{-1/\tau}{\to} \begin{pmatrix} 1 \\ 0 \\ \tau+1 \end{pmatrix},$$

otherwise

$$\begin{pmatrix} 1 \\ 0 \\ f_{j^*}(d) \end{pmatrix} \overset{-1/\tau}{\to} \begin{pmatrix} -\frac{1}{\tau} \\ -\frac{f_{j^*}(d)}{\tau} \\ 0 \end{pmatrix} \overset{-1/\tau}{\to} \begin{pmatrix} 1 \\ 0 \\ -f_{j^*}(d) \end{pmatrix}.$$

Consider now the imbedded process

$$\omega_k = d \longrightarrow \omega_{k+1} = \begin{cases} \tau+1 & \text{if } f_{j^*}(d) > -(\tau+1) \\ -f_{j^*}(d) & \text{otherwise.} \end{cases}$$

We can write $\omega_{k+1} = h(\omega_k)$, with

$$h(\omega) = \max(\tau+1, -f(\omega)),\ \tau+1 \le \omega \le 0,$$

and

$$f(\omega) = f_j(\omega) \text{ when } \frac{(1-\tau)(1-2^j)}{2^j} < \omega \le \frac{(1-\tau)(1-2^{j-1})}{2^{j-1}}.$$

This imbedded process is periodic, its period $n$ is a function of $\tau$. Let $s_1, \ldots, s_n$ be the states visited by the imbedded process. The original process is periodic too, with period $T(\tau) = \sum_{i=1}^n 2j^*(s_i) + 2$, for instance $T(-9) = 6$, $T(-10) = 16$, $T(-11) = 4$ and $T(-11.0001) = 102$. Its ergodic log-rate is $\frac{1}{T(\tau)} \sum_{i=1}^n (2j^*(s_i)\log 2 - 2\log r_i)$, with $r_i = \frac{1-\tau-f_{j^*}(s_i)}{4\tau^2}$ if $f_{j^*}(s_i) > -(\tau+1)$ and $r_i = \frac{1}{\tau^2}$ otherwise. The worst log-rate is $\log 2$ obtained for $\tau = -2$. The same study is valid for $-1 < \tau < 0$, simply replacing $\tau$ by $\frac{1}{\tau}$.

## References


1. O.E. Barndorff-Nielsen, J.L. Jensen, W.S. Kendall (Eds.) *Networks and Chaos – Statistical and Probabilistic Aspects.* Chapman & Hall, London, 1993.
2. T. Bedford, M. Keane, and C. Series, editors (1991). *Ergodic Theory, Symbolic Dynamics and Hyperbolic Spaces.* Oxford University Press, New York.

3. R.G. Bland, D. Goldfarb, and M.J. Todd (1981). The ellipsoid method: a survey. *Operations Research*, **29**(6):1039–1091.
4. J.-P. Eckmann and D. Ruelle (1985). Ergodic theory of chaos and strange attractors. *Rev. Mod. Phys.*, **57**:617–656.
5. D.G. Luenberger (1973). *Introduction to Linear and Nonlinear Programming.* Addison-Wesley, Reading, Massachusetts.
6. R. Mañé (1987). *Ergodic Theory and Differentiable Dynamics.* Springer, Berlin.
7. E. Ott (1993). *Chaos in Dynamical Systems.* Cambridge University Press, New York.
8. L. Pronzato, H.P. Wynn, and A.A. Zhigljavsky (1995). Stochastic analysis of convergence via dynamic representation for a class of line-search algorithms. Submitted to *Combinatorics, Probability and Computing.*
9. D. Ruelle (1989). *Chaotic Evolution and Strange Attractors.* Cambridge University Press, New York.
10. H.P. Wynn and A.A. Zhigljavsky (1993). Chaotic behaviour of search algorithms. *Acta Applicandae Mathematicae*, **32**:123–156.
11. H.P. Wynn and A.A. Zhigljavsky (1994). Achieving the ergodically optimal convergence rate for a one-dimensional minimization problem. *J. of Complexity*, **11**:196–226.