Journal of Global Optimization **24:** 51–60, 2002.

Global Optimization using Dynamic Search Trajectories**

ALBERT A. GROENWOLD and J.A. SNYMAN
Department of Mechanical Engineering, University of Pretoria, Pretoria, 0002, South Africa
(e-mail: Albert.Groenwold@eng.up.ac.za; e-mail: Jan.Snyman@eng.up.ac.za)

(Received 23 January 2001; accepted in revised form 7 December 2001)

Abstract. Two global optimization algorithms are presented. Both algorithms attempt to minimize an unconstrained objective function through the modeling of dynamic search trajectories. The first, namely the Snyman–Fatti algorithm, originated in the 1980's and still appears an effective global optimization algorithm. The second algorithm is currently under development, and is denoted the modified bouncing ball algorithm. For both algorithms, the search trajectories are modified to increase the likelihood of convergence to a low local minimum. Numerical results illustrate the effectiveness of both algorithms.

Key words: Global optimization, Dynamic search trajectories

1. Introduction

The problem of globally optimizing a real valued function is inherently intractable (unless hard restrictions are imposed on the objective function) in that no practically useful characterization of the global optimum is available. Indeed, the problem of determining an accurate estimate of the global optimum is mathematically ill-posed in the sense that very similar objective functions may have global optima very distant from each other [1]. Nevertheless, the need in practice to find a relative low local minimum has resulted in considerable research over the last decade to develop algorithms that attempt to find such a low minimum, e.g. see [2].

The general global optimization problem may be formulated as follows. Given a real valued objective function $f(\boldsymbol{x})$ defined on the set $\boldsymbol{x} \in D$ in $\mathbb{R}^n$, find the point $\boldsymbol{x}^*$ and the corresponding function value f^* such that

$$f^* = f(\boldsymbol{x}^*) = \text{ minimum } \{f(\boldsymbol{x}) | \boldsymbol{x} \in D\} \tag{1.1}$$

if such a point $\boldsymbol{x}^*$ exists. If the objective function and/or the feasible domain D are non-convex, then there may be many local minima which are not global.

If D corresponds to all $\mathbb{R}^n$ the optimization problem is *unconstrained*. Alternatively, simple bounds may be imposed, with D now corresponding to the hyper

** An abbreviated version of this paper was read at the Conference on Discrete and Global Optimization, Chania, Crete, May 1998.

box (or domain or region of interest) defined by

$$D = \{\boldsymbol{x} | \boldsymbol{\ell} \leq \boldsymbol{x} \leq \boldsymbol{u}\} \tag{1.2}$$

where $\boldsymbol{\ell}$ and $\boldsymbol{u}$ are n-vectors defining the respective lower and upper bounds on $\boldsymbol{x}$.

From a *mathematical* point of view, Problem (1.1) is essentially *unsolvable*, due to a lack of mathematical conditions characterizing the global optimum, as opposed to the local optimum of a smooth continuous function, which is characterized by the behavior of the problem function (Hessians and gradients) at the minimum [3] (viz. the Karush–Kuhn–Tucker conditions). Therefore, the global optimum f^* can only be obtained by an exhaustive search, except if the objective function satisfies certain subsidiary conditions [4], which mostly are of limited practical use [5]. Typically, the conditions are that f should satisfy a Lipschitz condition with known constant L and that the search area is bounded, e.g. for all $\boldsymbol{x}, \bar{\boldsymbol{x}} \in \boldsymbol{X}$

$$|f(\boldsymbol{x}) - f(\bar{\boldsymbol{x}})| \ \leq \ L||\boldsymbol{x} - \bar{\boldsymbol{x}}|| \tag{1.3}$$

So-called space-covering deterministic techniques have been developed [6] under these special conditions. These techniques are expensive, and due to the need to know L, of limited practical use.

Global optimization algorithms are divided into two major classes [6]: deterministic and stochastic (from the Greek word *stokhastikos*, i.e. 'governed by the laws of probability'). Deterministic methods can be used to determine the global optimum through exhaustive search. These methods are typically extremely expensive. With the introduction of a stochastic element into deterministic algorithms, the deterministic *guarantee* that the global optimum can be found is relaxed into a *confidence measure*. Stochastic methods can be used to assess the probability of having obtained the global minimum. Stochastic ideas are mostly used for the development of stopping criteria, or to approximate the regions of attraction as used by some methods [3].

The stochastic algorithms presented herein, namely the Snyman–Fatti algorithm and the modified bouncing ball algorithm, both depend on dynamic search trajectories to minimize the objective function. The respective trajectories, namely the motion of a particle of unit mass in a n-dimensional conservative force field, and the trajectory of a projectile in a conservative gravitational field, are modified to increase the likelihood of convergence to a low local minimum.

2. The Snyman–Fatti trajectory method

The essentials of the original SF algorithm [5] using dynamic search trajectories for unconstrained global minimization will now be discussed. The algorithm is based on the local algorithms presented in [7, 8]. For more details concerning the motivation of the method, its detailed construction, convergence theorems, computational aspects and some of the more obscure heuristics employed, the reader is referred to the original paper.

2.1. DYNAMIC TRAJECTORIES

In the SF algorithm successive sample points $\boldsymbol{x}^j$, $j = 1, 2, ...$, are selected at random from the box D defined by (2). For *each* sample point $\boldsymbol{x}^j$, a sequence of trajectories T^i, $i = 1, 2, ...$, is computed by numerically solving the successive initial value problems:

$$\begin{aligned} \ddot{\boldsymbol{x}}(t) &= -\nabla f(\boldsymbol{x}(t)) \\ \boldsymbol{x}(0) = \boldsymbol{x}_0^i\,; &\qquad \dot{\boldsymbol{x}}(0) = \dot{\boldsymbol{x}}_0^i \end{aligned} \tag{2.4}$$

This trajectory represents the motion of a particle of unit mass in a n-dimensional conservative force field, where the function to be minimized represents the potential energy.

Trajectory T^i is terminated when $\boldsymbol{x}(t)$ reaches a point where $f(\boldsymbol{x}(t))$ is arbitrarily close to the value $f(\boldsymbol{x}_0^i)$ while moving 'uphill', or more precisely, if $\boldsymbol{x}(t)$ satisfies the conditions.

$$\begin{aligned} &f(\boldsymbol{x}(t)) > f(\boldsymbol{x}_0^i) - \epsilon_u \\ &\text{and} \quad \dot{\boldsymbol{x}}(t)^T \nabla f(\boldsymbol{x}(t)) > 0 \end{aligned} \tag{2.5}$$

where ϵ_u is an arbitrary small prescribed positive value.

An argument is presented in [5] to show that when the level set $\{\boldsymbol{x} | f(\boldsymbol{x}) \leq f(\boldsymbol{x}_0^i)\}$ is bounded and $\nabla f(\boldsymbol{x}_0^i) \neq \boldsymbol{0}$, then conditions (2.5) above will be satisfied at some finite point in time.

Each computed step along trajectory T^i is monitored so that at termination the point $\boldsymbol{x}_m^i$ at which the minimum value was achieved is recorded together with the associated velocity $\dot{\boldsymbol{x}}_m^i$ and function value f_m^i. The values of $\boldsymbol{x}_m^i$ and $\dot{\boldsymbol{x}}_m^i$ are used to determine the initial values for the next trajectory T^{i+1}. From a comparison of the minimum values the best point $\boldsymbol{x}_b^i$, for the current j over all trajectories to date is also recorded. In more detail the minimization procedure for *a given sample point* $\boldsymbol{x}^j$, in computing the sequence $\boldsymbol{x}_b^i$, $i = 1, 2, ...$, is as follows.

2.2. MINIMIZATION PROCEDURE *MP1*

1. For given sample point $\boldsymbol{x}^j$, set $\boldsymbol{x}_0^1 := \boldsymbol{x}^j$ and compute T^1 subject to $\dot{\boldsymbol{x}}_0^1 := 0$; record $\boldsymbol{x}_m^1, \dot{\boldsymbol{x}}_m^1$ and f_m^1 ; set $\boldsymbol{x}_b^1 := \boldsymbol{x}_m^1$ and $i := 2$,
2. compute trajectory T^i with $\boldsymbol{x}_0^i := \frac{1}{2}\left(\boldsymbol{x}_0^{i-1} + \boldsymbol{x}_b^{i-1}\right)$ and $\dot{\boldsymbol{x}}_0^i := \frac{1}{2}\dot{\boldsymbol{x}}_m^{i-1}$, record $\boldsymbol{x}_m^i, \dot{\boldsymbol{x}}_m^i$ and f_m^i,
3. if $f_m^i < f(\boldsymbol{x}_b^{i-1})$ then $\boldsymbol{x}_b^i := \boldsymbol{x}_m^i$; else $\boldsymbol{x}_b^i := \boldsymbol{x}_b^{i-1}$,
4. set $i := i + 1$ and go to 2.

In the original paper [5] an argument is presented to indicate that under normal conditions on the continuity of f and its derivatives, $\boldsymbol{x}_b^i$ will converge to a local minimum. Procedure *MP1*, for a given j, is accordingly terminated at step 3 above if $||\nabla f(\boldsymbol{x}_b^i)|| \leq \epsilon$, for some small prescribed positive value ϵ, and $\boldsymbol{x}_b^i$ is taken

as the local minimizer $\boldsymbol{x}_f^j$, i.e. set $\boldsymbol{x}_f^j := \boldsymbol{x}_b^i$ with corresponding function value $f_f^j := f(\boldsymbol{x}_f^j)$.

Reflecting on the overall approach outlined above, involving the computation of energy conserving trajectories and the minimization procedure, it should be evident that, in the presence of many local minima, the probability of convergence to a relative low local minimum is increased. This one expects because, with a small value of ϵ_u (see conditions (4)), it is likely that the particle will move through a trough associated with a relative high local minimum, and move over a ridge to record a lower function value at a point beyond. Since we assume that the level set associated with the starting point function is bounded, termination of the search trajectory will occur as the particle eventually moves to a region of higher function values.

3. The modified bouncing ball trajectory method

The essentials of the modified bouncing ball algorithm using dynamic search trajectories for unconstrained global minimization are now presented. The algorithm is in an experimental stage, and details concerning the motivation of the method, its detailed construction, and computational aspects will be presented in future.

3.1. DYNAMIC TRAJECTORIES

In the MBB algorithm successive sample points $\boldsymbol{x}^j$, $j = 1, 2, ...,$ are selected at random from the box D defined by (2). For *each* sample point $\boldsymbol{x}^j$, a sequence of *trajectory steps* $\boldsymbol{\Delta x}^i$ and associated *projection points* $\boldsymbol{x}^{i+1}$, $i = 1, 2, ...,$ are computed from the successive analytical relationships (with $\boldsymbol{x}^1 := \boldsymbol{x}^j$ and prescribed $V_{0_1} > 0$):

$$\boldsymbol{\Delta x}^i = V_{0_i} t_i \cos\theta_i \nabla f(\boldsymbol{x}^i)/||\nabla f(\boldsymbol{x}^i)|| \tag{3.6}$$

where:

$$\theta_i = \tan^{-1}(||\nabla f(\boldsymbol{x}^i)||) + \frac{\pi}{2}, \tag{3.7}$$

$$t_i = \frac{1}{g}\left[V_{0_i}\sin\theta_i + \left\{(V_{0_i}\sin\theta_i)^2 + 2gh(\boldsymbol{x}^i)\right\}^{1/2}\right], \tag{3.8}$$

$$h(\boldsymbol{x}^i) = f(\boldsymbol{x}^i) + k \tag{3.9}$$

with k a constant chosen such that $h(\boldsymbol{x}) > 0 \; \forall \; \boldsymbol{x} \in D$, g a positive constant, and

$$\boldsymbol{x}^{i+1} = \boldsymbol{x}^i + \boldsymbol{\Delta x}^i \tag{3.10}$$

For the next step, select $V_{0_{i+1}} < V_{0_i}$. Each step $\boldsymbol{\Delta x}^i$ represents the ground or horizontal displacement obtained by projecting a particle in a vertical gravitational field (constant g) at an elevation $h(\boldsymbol{x}^i)$ and speed V_{0_i} at an inclination θ_i. The angle

θ_i represents the angle that the outward normal $\boldsymbol{n}$ to the hypersurface represented by $y = h(\boldsymbol{x})$ makes, at $\boldsymbol{x}^i$ in $n+1$ dimensional space, with the horizontal. The time of flight t_i is the time taken to reach the ground corresponding to $y = 0$.

More formally, the minimization trajectory for *a given sample point* $\boldsymbol{x}^j$ and some initial prescribed speed V_0 is obtained by computing the sequence $\boldsymbol{x}^i$, $i = 1, 2, ...,$ as follows.

3.2. MINIMIZATION PROCEDURE *MP2*

1. For given sample point $\boldsymbol{x}^j$, set $\boldsymbol{x}^1 := \boldsymbol{x}^j$ and compute trajectory step $\boldsymbol{\Delta x}^1$ according to (3.6) – (3.9) and subject to $V_{0_1} := V_0$; record $\boldsymbol{x}^2 := \boldsymbol{x}^1 + \boldsymbol{\Delta x}^1$, set $i := 2$ and $V_{0_2} := \alpha V_{0_1}$ ($\alpha < 1$)
2. Compute $\boldsymbol{\Delta x}^i$ according to (3.6) – (3.9) to give $\boldsymbol{x}^{i+1} := \boldsymbol{x}^i + \boldsymbol{\Delta x}^i$, record $\boldsymbol{x}^{i+1}$ and set $V_{0_{i+1}} := \alpha V_{0_i}$
3. set $i := i + 1$ and go to 2

In the vicinity of a local minimum $\hat{\boldsymbol{x}}$ the sequence of projection points $\boldsymbol{x}^i$, $i = 1, 2, ...,$ constituting the search trajectory for starting point $\boldsymbol{x}^j$ will converge since $\boldsymbol{\Delta x}^i \to 0$ (see (3.6)). In the presence of many local minima, the probability of convergence to a relative low local minimum is increased, since the kinetic energy can only decrease for $\alpha < 1$.

Procedure *MP2*, for a given j, is successfully terminated if $||\nabla f(\boldsymbol{x}^i)|| \leq \epsilon$ for some small prescribed positive value ϵ, or when $\alpha V_0^i < \beta V_0^1$, and $\boldsymbol{x}^i$ is taken as the local minimizer $\boldsymbol{x}_f^j$ with corresponding function value $f_f^j := h(\boldsymbol{x}_f^j) - k$.

Clearly, the condition $\alpha V_0^i < \beta V_0^1$ will always occur for $0 < \beta < \alpha$ and $0 < \alpha < 1$.

MP2 can be viewed as a variant of the steepest descent algorithm. However, as opposed to steepest descent, MP2 has (as has MP1) the ability for 'hill-climbing', as is inherent in the physical model on which MP2 is based (viz., the trajectories of a bouncing ball in a conservative gravitational field.) Hence, the behavior of MP2 is quite different from that of steepest descent and furthermore, because of it's physical basis, it tends to seek local minima with relative low function values and is therefore suitable for implementation in global searches, while steepest descent is not.

For the MBB algorithm, convergence to a local minimum is not proven. Instead, the underlying physics of a bouncing ball is exploited. Unsuccessful trajectories are terminated, and do not contribute to the probabilistic stopping criterion (although these points are included in the number of unsuccessful trajectories $\tilde{n}$.) In the validation of the algorithm, the philosophy adopted here is that the practical demonstration of convergence of a proposed algorithm on a variety of demanding test problems may be as important and convincing as a rigorous mathematical convergence argument. Indeed, although for the steepest descent method convergence can be proven, in practice it often fails to converge because effectively an infinite number of steps is required for convergence.

4. Global stopping criterion

The above methods require a termination rule for deciding when to end the sampling and to take the current overall minimum function value $\tilde{f}$, i.e.

$$\tilde{f} = \text{ minimum } \left\{ f_f^j, \text{ over all } j \text{ to date } \right\} \tag{4.11}$$

as the global minimum value f^*.

Define the *region of convergence* of the dynamic methods for a local minimum $\hat{\boldsymbol{x}}$ as the set of all points $\boldsymbol{x}$ which, used as starting points for the above procedures, converge to $\hat{\boldsymbol{x}}$. One may reasonably expect that in the case where the *regions of attraction* (for the usual gradient-descent methods, see [9]) of the local minima are more or less equal, that the region of convergence of the global minimum will be relatively increased.

Let R_k denote the region of convergence for the above minimization procedures *MP1* and *MP2* of local minimum $\hat{\boldsymbol{x}}^k$ and let α_k be the associated probability that a sample point be selected in R_k. The region of convergence and the associated probability for the global minimum $\boldsymbol{x}^*$ are denoted by R^* and α^*, respectively. The following basic assumption, which is probably true for many functions of practical interest, is now made.

A. Basic assumption:

$$\alpha^* \geq \alpha_k \text{ for all local minima } \hat{\boldsymbol{x}}^k. \tag{4.12}$$

The following theorem may be proved (see Appendix A).

B. Theorem: (Ref. [5]): Let r be the number of sample points falling within the region of convergence of the current overall minimum $\tilde{f}$ after $\tilde{n}$ points have been sampled. Then under assumption A and a statistically non-informative prior distribution the probability that $\tilde{f}$ corresponds to f^* may be obtained from:

$$Pr\left[\tilde{f} = f^*\right] \geq q(\tilde{n}, r) = 1 - \frac{(\tilde{n}+1)!(2\tilde{n}-r)!}{(2\tilde{n}+1)!(\tilde{n}-r)!} \tag{4.13}$$

On the basis of this theorem the *stopping rule* becomes: STOP when $Pr\left[\tilde{f} = f^*\right] \geq q^*$, where q^* is some prescribed desired confidence level, typically chosen as 0.99.

5. Numerical results

The test functions used are tabulated in Table I, and tabulated numerical results are presented in Tables II and III. In the tables, the reported number of function values N_f are the average of 10 independent (random) starts of each algorithm.

Unless otherwise stated, the following settings were used in the SF algorithm (see [5]): $\gamma = 2.0$, $\alpha = 0.95$, $\epsilon = 10^{-2}$, $\omega = 10^{-2}$, $\delta = 0.0$, $q^* = 0.99$, and $\Delta t = 1.0$. For the MBB algorithm, $\alpha = 0.99$, $\epsilon = 10^{-4}$, and $q^* = 0.99$ were used. For each problem, the initial velocity V_0 was chosen such that $\boldsymbol{\Delta x}^1$ was equal to

Table I. The test functions

No.	Name	ID	n	Ref.
1	Griewank G1	G1	2	[2, 4]
2	Griewank G2	G2	10	[2, 4]
3	Goldstein–Price	GP	2	[2, 10]
4	Six-hump Camelback	C6	2	[2, 11]
5	Shubert, Levi No. 4	SH	2	[12]
6	Branin	BR	2	[2, 13]
7	Rastrigin	RA	2	[2]
8	Hartman 3	H3	3	[2, 10]
9	Hartman 6	H6	6	[2, 10]
10	Shekel 5	S5	4	[2, 10]
11	Shekel 7	S7	4	[2, 10]
12	Shekel 10	S10	4	[2, 10]

Table II. Numerical results

No.	ID	SF – This study			Ref. [5]		MBB		
		N_f	$(r/\tilde{n})_b$	$(r/\tilde{n})_w$	N_f	$r/\tilde{n}$	N_f	$(r/\tilde{n})_b$	$(r/\tilde{n})_w$
1	G1	4199	6/40	6/75	1606	6/20	2629	5/8	6/23
2	G2	25969	6/84	6/312	26076	6/60	19817	6/24	6/69
3	GP	2092	4/4	5/12	668	4/4	592	4/4	5/10
4	C6	426	4/4	5/9	263	4/4	213	4/4	5/10
5	SH	8491	6/29	6/104	—	—	1057	5/7	6/26
6	BR	3922	4/4	5/12	—	—	286	4/4	5/6
7	RA	4799	6/67	6/117	—	—	1873	4/4	6/42
8	H3	933	4/4	5/8	563	5/6	973	5/9	6/29
9	H6	1025	4/4	5/10	871	5/8	499	4/4	5/9
10	S5	1009	4/4	6/24	1236	6/17	2114	5/8	6/39
11	S7	1057	5/8	6/37	1210	6/17	2129	6/16	6/47
12	S10	845	4/4	6/31	1365	6/20	1623	5/7	6/39

half the ‘radius’ of the domain D. A local search strategy was implemented with varying α in the vicinity of local minima.

In Table II, $(r/\tilde{n})_b$ and $(r/\tilde{n})_w$ respectively indicate the best and worst $r/\tilde{n}$ ratios (see Equation (4.13)), observed during 10 independent optimization runs of both algorithms. The SF results compare well with the previously published results by Snyman and Fatti, who reported values for a single run only. For the

Table III. Cost (N_f) using *a priori* stopping condition

Method	Test function					
	BR	C6	GP	RA	SH	H3
TRUST	55	31	103	59	72	58
MBB	25	29	74	168	171	24

Shubert, Branin and Rastrigin functions, the MBB algorithm is superior to the SF algorithm. For the Shekel functions (S5, S7 and S10), the SF algorithm is superior. As a result of the stopping criterion (4.13), the SF and MBB algorithms found the global optimum between 4 and 6 times for each problem.

The results for the trying Griewank functions (Table II) are encouraging. G1 has some 500 local minima in the region of interest, and G2 several thousand. The values used for the parameters are as specified, with $\Delta t = 5.0$ for G1 and G2 in the SF-algorithm. It appears that both the SF and MBB algorithms are highly effective for problems with a large number of local minima in D, and problems with a large number of design variables.

In Table III the MBB algorithm is compared with the recently published deterministic TRUST algorithm [14]. Since the TRUST algorithm was terminated when the global approximation was within a specified tolerance of the (known) global optimum, a similar criterion was used for the MBB algorithm. The table reveals that the two algorithms compare well. Note, however, that the highest dimension of the test problems used in [14] is 3. It is unclear if the deterministic TRUST algorithm will perform well for problems of large dimension, or problems with a large number of local minima in D.

6. Conclusions

Two stochastic global optimization methods based on dynamic search trajectories are presented. The algorithms are the Snyman–Fatti trajectory method and the modified bouncing ball trajectory method. Numerical results indicate that both algorithms are effective in finding the global optimum efficiently. In particular, the results for the trying Griewank functions are encouraging. Both algorithms appear effective for problems with a large number of local minima in the domain, and problems with a large number of design variables. A salient feature of the algorithms is the availability of an apparently effective global stopping criterion.

References

1. Schoen, F. (1991), Stochastic Techniques for Global Optimization: A Survey of Recent Advances, *J. Global Optim.* **1**, 207–228.
2. Törn, A. and Zilinskas, A. (1989), *Global Optimization: Lecture Notes in Computer Science*, **350**, Springer-Verlag, Berlin.
3. Arora, J.S., El-wakeil, O.A., Chahande, A.I. and Hsieh, C.C. (1995), Global Optimization Methods for Engineering Applications: A Review, *Preprint*.
4. Griewank, A.O. (1981), Generalized Descent for Global Optimization, *J. Optim. Theory Appl.* **34**, 11–39.
5. Snyman, J.A. and Fatti, L.P. (1987), A Multi-start Global Minimization Algorithm with Dynamic Search Trajectories, *J. Optim. Theory Appl.*, **54**, 121–141.
6. Dixon, L.C.W., Gomulka, J. and Szegö, G.P. (1975), Towards a Global Optimization Technique, (in Dixon, L.C.W. and Szegö, G.P., (eds.), *Towards Global Optimization*, North-Holland, Amsterdam, pp. 29–54.
7. Snyman, J.A. (1982), A New and Dynamic Method for Unconstrained Minimization, *Appl. Math. Modeling* **6**, 449–462.
8. Snyman, J.A. (1983), An Improved Version of the Original Leap-Frog Dynamic Method for Unconstrained Minimization: LFOP1(b), *Appl. Math. Modeling* **7**, 216–218.
9. Dixon, L.C.W., Gomulka, J. and Hersom, S.E. (1976), Reflections on the Global Optimization Problem, in Dixon, L.C.W., (ed.), *Optimization in Action*, Academic Press, London, pp. 398–435.
10. Dixon, L.C.W. and Szegö, G.P. (1978), The Global Optimization Problem: An Introduction, in Dixon, L.C.W. and Szegö, G.P., (eds.), *Towards Global Optimization 2*, North-Holland, Amsterdam, pp. 1–15.
11. Branin, F.H. (1972), Widely Used Convergent Methods for Finding Multiple Solutions of Simultaneous Equations, *IBM J. Research Develop.* 504–522.
12. Lucidi, S. and Piccioni, M. (1989), Random Tunneling by means of Acceptance–rejection Sampling for Global Optimization, *J. Opt. Theory Appl.* **62**, 255–277.
13. Branin, F.H. and Hoo, S.K. (1972), A Method for Finding Multiple Extrema of a Function of n Variables, in Lootsma, F.A. (ed.), *Numerical Methods of Nonlinear Optimization*, Academic Press, London, pp. 231–237.
14. Barhen, J., Protopopescu, V. and Reister, D. (16 May 1997), TRUST: A Deterministic Algorithm for Global Optimization, *Science* **276**, 1094–1097.
15. Zielinsky, R. (1981), A Statistical Estimate of the Structure of Multiextremal Problems, *Math. Program.* **21**, 348–356.

Appendix

A. Proof of stopping criterion

We present here an outline of the proof of (4.13), and follow closely the presentation in [5]. (We have since learned that the proof can be shown to be a generalization of the procedure proposed by Zielinski [15].) Given $\tilde{n}^*$ and α^*, the probability that at least one point, $\tilde{n} \geq 1$, has converged to f^* is:

$$\Pr[\tilde{n}^* \geq 1 | \tilde{n}, r] = 1 - (1 - \alpha^*)^{\tilde{n}} \ . \tag{1.14}$$

In the Bayesian approach, we characterize our uncertainty about the value of α^* by specifying a prior probability distribution for it. This distribution is modified

using the sample information (namely, $\tilde{n}$ and r) to form a posterior probability distribution. Let $p_*(\alpha^*|\tilde{n}, r)$ be the posterior probability distribution of α^*. Then,

$$\begin{aligned}\Pr[\tilde{n}^* \geq 1|\tilde{n}, r] &= \int_0^1 \left[1 - (1-\alpha^*)^{\tilde{n}}\right] p_*(\alpha^*|\tilde{n}, r)\mathrm{d}\alpha^* \\ &= 1 - \int_0^1 (1-\alpha^*)^{\tilde{n}} p_*(\alpha^*|\tilde{n}, r)\mathrm{d}\alpha^*. \end{aligned} \tag{1.15}$$

Now, although the r sample points converge to the current overall minimum, we do not know whether this minimum corresponds to the global minimum of f^*. Utilizing (4.12), and noting that $(1-\alpha)^{\tilde{n}}$ is a decreasing function of α, the replacement of α^* in the above integral by α yields

$$\Pr[\tilde{n}^* \geq 1|\tilde{n}, r] \geq \int_0^1 \left[1 - (1-\alpha)^{\tilde{n}}\right] p(\alpha|\tilde{n}, r)\mathrm{d}\alpha \ . \tag{1.16}$$

Now, using Bayes theorem we obtain:

$$p(\alpha|\tilde{n}, r) = \frac{p(r|\alpha, \tilde{n})\, p(\alpha)}{\int_0^1 p(r|\alpha, \tilde{n})\, p(\alpha)\, \mathrm{d}\alpha} \ . \tag{1.17}$$

Since the $\tilde{n}$ points are sampled at random and each point has a probability α of converging to the current overall minimum, r has a binomial distribution with parameters α and $\tilde{n}$. Therefore:

$$p(r|\alpha, \tilde{n}) = \binom{\tilde{n}}{r} \alpha^r (1-\alpha)^{\tilde{n}-r} \ . \tag{1.18}$$

Substituting (1.18) and (1.17) into (1.16) gives:

$$\Pr[\tilde{n}^* \geq 1|\tilde{n}, r] \geq 1 - \frac{\int_0^1 \alpha^r (1-\alpha)^{2\tilde{n}-r} p(\alpha)\, \mathrm{d}\alpha}{\int_0^1 \alpha^r (1-\alpha)^{\tilde{n}-r} p(\alpha)\, \mathrm{d}\alpha} \ . \tag{1.19}$$

A suitable flexible prior distribution $p(\alpha)$ for α is the beta distribution with parameters a and b. Hence:

$$p(\alpha) = \left[1/\boldsymbol{\beta}(a, b)\right] \alpha^{a-1}(1-\alpha)^{b-1}, \qquad 0 \leq \alpha \leq 1 \tag{1.20}$$

Using this prior distribution gives:

$$\begin{aligned}\Pr[\tilde{n}^* \geq 1|\tilde{n}, r] &\geq 1 - \frac{\Gamma(\tilde{n}+a+b)\ \Gamma(2\tilde{n}-r+b)}{\Gamma(2\tilde{n}+a+b)\ \Gamma(\tilde{n}-r+b)} \\ &= 1 - \frac{(\tilde{n}+a+b-1)!\ (2\tilde{n}-r+b-1)!}{(2\tilde{n}+a+b-1)!\ (\tilde{n}-r+b-1)!}, \end{aligned}$$

Assuming a prior expectation of 1, (viz. $a = b = 1$), we obtain:

$$\Pr[\tilde{n}^* \geq 1|\tilde{n}, r] = 1 - \frac{(\tilde{n}+1)!\ (2\tilde{n}-r)!}{(2\tilde{n}+1)!\ (\tilde{n}-r)!},$$

which is the required result.