Multistart Method with Estimation Scheme for Global Satisfycing Problems

L. HE and E. POLAK
Department of Electrical Engineering and Computer Sciences and the Electronics Research Laboratory, University of California, Berkeley, CA 94720 U.S.A.

(Received: 28 June 1990; accepted: 11 May 1992)

Abstract. We present a multistart method for solving global satisfycing problems. The method uses data generated by linearly converging local algorithms to estimate the cost value at the local minimum to which the local search is converging. When the estimate indicates that the local search is converging to a value higher than the satisfycing value, the local search is interrupted and a new local search is initiated from a randomly generated point. When the satisfycing problem is difficult and the estimation scheme is fairly accurate, the new method is superior over a straightforward adaptation of classical multistart methods.

Key words. Global optimization, satisfycing problem, linear convergence.

1. Introduction

The task of finding parameters which satisfy performance specifications is of great importance both in engineering design and in economics planning. In engineering, parameters which satisfy specifications are referred to as feasible designs (see [14]), while in economics the special term *satisfycing* decisions has been coined to describe them (see, e.g. [22, 10]).

Frequently, performance specifications can be expressed in terms of a system of inequalities, such as

$$\max_{y_j \in \mathbf{Y}_j} \phi^j(x, y_j) \leq b_j \,, \quad j = 1, 2, \ldots, l \,, \tag{1.1}$$

where $\phi^j: \mathbb{R}^n \times \mathbb{R}^{m_j} \to \mathbb{R}$, $\mathbf{Y}_j \subset \mathbb{R}^{l_j}$, and the b_j express the desired *satisfycing* level for the j-th specification. The system of inequalities (1.1) is obviously equivalent to the more compact form

$$\Psi^j(x) \leq b_j \,, \quad j = 1, 2, \ldots, l \,, \tag{1.2}$$

where

$$\Psi^j(x) \overset{\Delta}{=} \max_{y_j \in \mathbf{Y}_j} \phi^j(x, y) \,, \quad j = 1, 2, \ldots, l \,. \tag{1.3}$$

Assuming that all $b_j > 0$, we can define a normalized *satisfycing* function $\Psi: \mathbb{R}^n \to \mathbb{R}$ by

$$\Psi(x) \overset{\Delta}{=} \max_{j \in \underline{l}} \frac{\Psi^j(x)}{b_j} - 1 \,, \tag{1.4}$$

Journal of Global Optimization **3**: 139–156, 1993.

where $\underline{l} \overset{\Delta}{=} \{1, 2, \ldots, l\}$. Then we can replace the original system of inequalities (1.1) by the single nonsmooth inequality

$$\Psi(x) \leq \nu , \tag{1.5}$$

where $\nu \geq 0$ is the overall satisfycing level, introduced to account for the possibility that the specifications are not consistent.

Quite commonly, the functions $\phi^j(\cdot, \cdot)$ and their gradients $\nabla\phi^j(\cdot, \cdot)$ are Lipschitz continuous and the sets Y_j are compact intervals of the real line. In such cases, it for every local minimizer $\hat{x}$, $\Psi(\hat{x}) < 0$, then an *implementable* algorithm such as the one presented in [17], which combines the Pshenichnyi–Pironneau–Polak minimax algorithm [18, 12, 14] with appropriate schemes for the discretization of the sets Y_j, is capable of finding a satisfying solution x^* in a finite number of iterations. Unfortunately, it is not uncommon for a satisfycing function to have strictly positive local minima, which can trap a minimax algorithm. Thus we see that although the satisfycing problem is easier than that of finding the global minimum of $\Psi(\cdot)$, the two problems do have common features.

The Pshenichnyi–Pironneau–Polak minimax algorithm converges linearly (see [15]). However, the standard rate of convergence analysis addresses only the *conceptual* algorithm: it does not take into account the effect of the inevitable discretizations of the sets Y_j in (1.1), required for the evaluation of the functions $\Psi^j(\cdot)$. Recently, we have developed a rate preserving adaptive discretization scheme which results in an implementable version of this algorithm which converges with the same rate as the conceptual version [16].

Now, as we will see in Section 3, whenever we use a linearly converging algorithm to minimize a function $\Psi(\cdot)$, we can use the values of $\Psi(\cdot)$ at the points it constructs to estimate the local minimum value to which it is converging. Clearly, if the estimated local minimum value is larger than ν, it makes sense to terminate the run and use a random reinitialization. The combination of this simple idea with well tried and highly efficient multistart global optimization techniques, including clustering, as in [3, 19, 20] and [21], is the basis of the novel multistart method for satisfycing problems presented in this paper.

In Section 2 we present our algorithm and analyze its performance under the assumption that the functions $\Psi^j(\cdot)$ can be evaluated without discretization (e.g., when the sets Y_j are singletons), because it is currently too difficult to account for the discretization effects. We show that when our local minimum value prediction method is fairly reliable and the satisfycing problem is inherently difficult, our multistart satisfycing method is demonstrably better than the underlying multistart method which does not use such estimates. Our analysis does not take into account the beneficial effects of clustering, since these affect both our method and the underlying multistart method more or less equally. In Section 3 we present our method for estimating the value of $\Psi(\cdot)$ at the local minimum to which the algorithm appears to be converging. Numerical results are presented in Section 4. These show that by and large, our local minimum value prediction method is highly reliable and that the new multistart method is considerably superior to the original multistart method.

2. Multistart Methods for Global Satisfycing Problem

To be realistic, the decision variable x in (1.1) must be assumed to be bounded. Hence consider global satisfycing problems of the form

$$\textbf{GSP}: \text{ find } x^* \in B \subset \mathbb{R}^n \text{ such that } \Psi(x^*) \leq \nu\,, \tag{2.1a}$$

where

$$B = \{x \in \mathbb{R}^n \,|\, l_i \leq x_i \leq u_i\,, \quad i = 1, \ldots, n\}\,, \tag{2.1b}$$

and the function $\Psi(\cdot)$ is as defined in (1.4), with the $\Psi^j: B \to \mathbb{R}$, $j \in \underline{l}$ either continuously differentiable or of the form (1.3), in which case we assume that the sets Y_j are compact intervals and that the functions $\phi^j(\cdot\,,\cdot)$ and their gradients $\nabla\phi^j(\cdot\,,\cdot)$, are locally Lipschitz continuous.

We will refer to x^* and ν, in (2.1a), as a *satisfycing solution* and the *satisfycing value*, respectively.

To construct our new algorithm, we begin with a simple multistart method for solving the global satisfycing problem **GSP**, which is an obvious adaptation of a standard multistart method for finding the global minimizer of $\Psi(\cdot)$ (see, e.g. [1, 2, 4, 6, 19]). We then modify this method by introducing into its termination tests estimates of the local minimum value to which the local search algorithm is converging. Although clustering schemes such as those in [3, 19, 20, 21] improve the performance of multistart global optimization methods, to simplify our analysis, we will not incorporate them in the algorithms below because they have exactly the same beneficial effect both on the simple multistart algorithm and our modification of it. However, we do expect these clustering techniques to be used in the final implementation of our method.

Let $A: B \to B$ denote the map defined by one iteration of a minimax algorithm. The simple multistart method for solving global satisfycing problems, below, is assumed to use a standard local search stopping rule, such as the step length, or value of an optimality criterion, dropping below a certain threshold.

MASTER ALGORITHM 2.1 (Simple Multistart Method for **GSP**).

```
Step 0: Set $i = 1$.
Step 1: Draw a random point $x_i$ from $B$. Set $z_0^i = x_i$ and $j = 0$.
Step 2: Compute $z_j^i = A(z_{j-1}^i)$.
Step 3: If $\Psi(z_j^i) \leq \nu$, then $z_j^i$ is a satifycing solution; stop.
        Else,
          If the local search stopping rule is satisfied, stop the local search, set
          $i = i + 1$ and go to Step 1;
          Else, replace $j$ by $j + 1$ and go to Step 2.    □
```

It is quite obvious that the simple multistart method is not very efficient because it may keep rediscovering the same unacceptable local minimizer x', (i.e. $\Psi(x') > \nu$). Clustering schemes, such as those in [3,19,20,21], reduce the occurrence of this undesirable phenomenon and can be used both in conjunction with Master Algorithm 2.1 as well as with the following one.

In the next section, we will describe an estimation scheme which uses the outcome of a certain number of iterations of the Pshenichnyi–Pironneau–Polak minimax algorithm to predict the value of $\Psi(\cdot)$ at the local minimizer to which the algorithm is converging. For the time being, we simply assume that we have such an estimation scheme associated with the local search algorithm, which, of course, need not be the Pshenichnyi–Pironneau–Polak minimax algorithm. Furthermore, we do not require that the estimation scheme be 100% accurate, in a sense to be made clear shortly.

For any $x \in B$ and positive integer k, let $\hat{\Psi}_{\min}(x, k)$ denote the *predicted* local minimum value, yielded by the estimation scheme on the basis of information produced in performing k iterations using the local algorithm A, starting from the initial point x, and let $\Psi_{\min}(x)$ denote the actual local minimum value to which the sequence $\{\Psi(A^i(x))\}_{i=0}^{\infty}$ converges (with $A^0(x) \overset{\Delta}{=} x$). Our new algorithm is as follows:

MASTER ALGORITHM 2.2. (Multistart Method for **GSP**, with Estimation Scheme)

Data: m_1, the minimum number of iterations in each local search required by the estimation scheme.

Step 0: Set $i = 1$.

Step 1: Draw a random point x_i from B. Set $z_0^i = x_i$ and $j = 0$.

Step 2: For $i = 1, 2, \ldots, m_1$, compute

$$z_j^i = A(z_{j-1}^i)\,. \tag{2.2}$$

Step 3: Estimate the local minimum value $\hat{\Psi}_{\min}(x_i, m_1)$.

Step 4: If $\hat{\Psi}_{\min}(x_i, m_1) > \nu$, stop the local search, set $i = i + 1$ and go to Step 1. Else, reinitialize the local search algorithm with $z_{m_1}^i$ and iterate until either a satisfycing solution is found or the local search stopping rule is satisfied.

Step 5: If a satisfycing solution was found, stop. Else, set $i = i + 1$ and go to Step 1. □

We will carry out our analysis under the following assumptions.

ASSUMPTION 2.1. (i) The random drawing is uniformly distributed on B.

(ii) The functions $\Psi^j(\cdot)$ in (1.3) can be evaluated without discretization of intervals.

(iii) The work associated with each iteration of the local algorithm $A(\cdot)$, expressed in function evaluation equivalents, is C units, where C is a constant.

(iv) The number of iterations used by one local search, starting from an initial point x_i until the stopping rule is satisfied, is m_2. □

The actual number of iterations used by one local search, starting from an initial point x_i until the stopping rule is satisfied, depends on the initial conditions in a manner that is difficult to model. Using an upper limit for m_2 is too conservative. In our analysis we will assume that m_2 is in fact equal to the average number of iterations observed. We will see from Table II that this assumption does not lead to incorrect final estimates.

NOTATION: (i) Let B_1 and B_2 denote two disjoint attraction regions for the local search algorithm, such that in B_1 the local minimum values are below or equal to the satisfycing value ν, and in B_2 the local minimum values are above the satifycing value ν, i.e.,

$$B_1 \overset{\Delta}{=} \{x \in B \mid \Psi_{\min}(x) \leq \nu\} , \tag{2.3a}$$

$$B_2 \overset{\Delta}{=} \{x \in B \mid \Psi_{\min}(x) > \nu\} , \tag{2.3b}$$

(ii) Let $B_{11}(m_1)$ and $B_{12}(m_1)$ be the two disjoint subsets of B_1, obtained by partitioning of B_1 according to the predicted local minimum value obtained at the end of m_1 iterations of the local search, i.e.

$$B_{11}(m_1) \overset{\Delta}{=} \{x \in B_1 \mid \hat{\Psi}_{\min}(x, m_1) \leq \nu\} \tag{2.4a}$$

$$B_{12}(m_1) \overset{\Delta}{=} \{x \in B_1 \mid \hat{\Psi}_{\min}(x, m_1) > \nu\} . \tag{2.4b}$$

Similarly, let $B_{21}(m_1)$ and $B_{22}(m_1)$ be the two disjoint subsets of B_2, obtained by partitioning B_2 according to the predicted local minimum value, i.e.,

$$B_{21}(m_1) \overset{\Delta}{=} \{x \in B_2 \mid \hat{\Psi}_{\min}(x, m_1) \leq \nu\} , \tag{2.4c}$$

$$B_{22}(m_1) \overset{\Delta}{=} \{x \in B_2 \mid \hat{\Psi}_{\min}(x, m_1) > \nu\} . \tag{2.4d}$$

(iii) For $i = 1, 2$, let $a_i \overset{\Delta}{=} \mathrm{Vol}(B_i)/\mathrm{Vol}(B)$, where "Vol" denotes the volume of the set in $\mathbb{R}^n$, and for $i = 1, 2$ and $j = 1, 2$, let $a_{ij}(m_1) \overset{\Delta}{=} \mathrm{Vol}(B_{ij}(m_1))/\mathrm{Vol}(B)$.

(iv) Let p_s denote the probability that a satisfycing solution will be found in one try by Master Algorithm 2.1 (simple multistart method), and let p_e denote the probability that a satisfycing solution will be found in one try by Master Algorithm 2.2 (multistart method with estimation scheme), respectively.

(v) Let NI_s and NI_e denote the number of outer iterations (initializations of the local algorithm) required for solving problem **GSP** by Master Algorithm 2.1 and Master Algorithm 2.2, respectively. Both NI_s and NI_e are random variables.

(vi) Let NF_s and NF_e denote the number of function evaluations required for solving problem **GSP** by Master Algorithm 2.1 and Master Algorithm 2.2, respectively. Both NF_s and NF_e are random variables. □

Note that the number $a_1 = \mathrm{Vol}(B_1)/\mathrm{Vol}(B)$ is a strong indicator of the satisfycing problem difficulty, since the expected number of local searches for Master Algorithm 2.1 is $1/a_1$. Note also that Master Algorithm 2.1 will stop if and only if an initial point x_i, produced by the random drawing, is in B_1, and Master Algorithm 2.2 will stop if and only if an initial point x_i, produced by the random drawing, is in $B_{11}(m_1)$. Hence we obtain the following, rather obvious result.

PROPOSITION 2.1. (i) $p_s = a_1 = a_{11}(m_1) + a_{12}(m_1)$, $p_e = a_{11}(m_1)$.

(ii) *The probability distributions of* NI_s *and* NI_e *are geometric distributions and have the following form*:

$$Prob(NI_s = k) = p_s(1-p_s)^{k-1}\,, \quad k = 1, 2, \ldots, \tag{2.5}$$

$$Prob(NI_e = k) = p_e(1-p_e)^{k-1}\,, \quad k = 1, 2, \ldots, \tag{2.6}$$

(iii) *Hence*,

$$Prob(NI_s \leqslant k) = (1-(1-p_s)^k)\,, \quad k = 1, 2, \ldots, \tag{2.7a}$$

$$E(NI_s) = 1/p_s\,, \quad D(NI_s) = (1-p_s)/p_s^2\,, \tag{2.7b}$$

and

$$Prob(NI_e \leqslant k) = (1-(1-p_e)^k)\,, \quad k = 1, 2, \ldots, \tag{2.8a}$$

$$E(NI_e) = 1/p_e\,, \quad D(NI_e) = (1-p_e)/p_e^2\,. \tag{2.8b}$$

where $E(z)$ *denotes the expectation and* $D(z)$ *denotes the standard deviation of a random variable*, z. □

Thus, we conclude that both multistart Master Algorithms terminate after a finite number of reinitializations of the local search algorithm, with probability 1. We restate this result as:

THEOREM 2.1.

$$Prob(NI_s < \infty) = 1\,, \tag{2.9}$$

$$Prob(NI_e < \infty) = 1\,. \tag{2.10}$$

□

The calculation of the average number of function evaluations used by Master Algorithm 2.2 in solving **GSP** is complicated by the fact that when $x_i \in B_{12}(m_1) \cup B_{22}(m_1)$, only m_1 iterations of local search algorithm are performed, while in all the other cases m_2 iterations are performed, and the fact that Master Algorithm

2.2 will stop if and only if $x_{NI_e} \in B_{11}(m_1)$. For $i = 1, 2$ and $j = 1, 2$, let the random variable NI_{ij} be the total number of local searches originated from starting points in $B_{ij}(m_1)$. Clearly $NI_{11} = 1$. Hence, NI_e can be expressed as $1 + NI_{12} + NI_{21} + NI_{22}$. Therefore, $NF_e = Cm_1(NI_{12} + NI_{22}) + Cm_2(1 + NI_{21})$. Note that Master Algorithm 2.2 will stop in $k + 1$ local searches, where $k \geqslant 0$, if and only if the starting points in the first k local searches are not in $B_{11}(m_1)$ and the starting point in the last local search is in $B_{11}(m_1)$. Hence we get the following result:

PROPOSITION 2.2. (i) *The joint distribution of* $(NI_{12}, NI_{21}, NI_{22})$ *is of the form*

$$Prob(NI_{12} = i_{12}, NI_{21} = i_{21}, NI_{22} = i_{22})$$
$$= \frac{(i_{12} + i_{21} + i_{22})!}{i_{12}!i_{21}!i_{22}!} a_{11}(m_1)a_{12}(m_1)^{i_{12}}a_{21}(m_1)^{i_{21}}a_{22}(m_1)^{i_{22}} . \tag{2.11}$$

(ii) *The joint distribution of* $(NI_{12} + NI_{22}, NI_{21})$ *is of the form*

$$Prob(NI_{12} + NI_{22} = i, NI_{21} = i_{21})$$
$$= \frac{(i + i_{21})!}{i!i_{21}!} a_{11}(m_1)[a_{12}(m_1) + a_{22}(m_1)]^i a_{21}(m_1)^{i_{21}} . \tag{2.12}$$

□

THEOREM 2.2. *The expected values* $E(NF_s)$ *and* $E(NF_e)$ *satisfy the following relations*

$$E(NF_s) = Cm_2E(NI_s) = Cm_2/p_s = Cm_2/a_1$$
$$= Cm_2 \frac{(a_{11}(m_1) + a_{21}(m_1))}{(a_{11}(m_1) + a_{12}(m_1))} + Cm_2 \frac{(a_{12}(m_1) + a_{22}(m_1))}{(a_{11}(m_1) + a_{12}(m_1))} , \tag{2.13}$$

$$E(NF_e) = Cm_2 \frac{(a_{11}(m_1) + a_{21}(m_1))}{a_{11}(m_1)} + Cm_1 \frac{(a_{12}(m_1) + a_{22}(m_1))}{a_{11}(m_1)} , \tag{2.14}$$

$$\frac{E(NF_e)}{E(NF_s)} = \frac{a_1}{a_{11}(m_1)} [a_{11}(m_1) + a_{21}(m_1)) + \frac{m_1}{m_2} (a_{12}(m_1) + a_{22}(m_1))] . \tag{2.15}$$

Proof. (2.13) follows immediately from Proposition 2.1(i), (iii) and the fact that $NF_s = Cm_2NI_s$. Let us assume that the number of local searches started from B_{21} is i, and the number of local searches started from B_{12} and B_{22} is j, where $i + j = k$. Then, clearly, the probability of such an occurrence is $Prob(NI_{12} + N_{22} = j, N_{21} = i)$, and the number of function evaluations involved is $Cm_2(1 + i) + Cm_1j$. Therefore, we have that

$$\begin{aligned}E(NF_e) &= \sum_{k=0}^{\infty} \sum_{i+j=k} (Cm_2(1+i) + Cm_1 j) Prob(NI_{12} + NI_{22} = j, \quad NI_{21} = i) \\ &= \sum_{k=0}^{\infty} \sum_{i+j=k} (Cm_2(1+i) + Cm_1 j) \frac{(i+j)!}{i!j!} a_{11}(m_1) \\ &\quad \times (a_{12}(m_1) + a_{22}(m_1))^j a_{21}(m_1)^i \\ &= a_{11}(m_1) \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} \frac{(i+j)!}{i!j!} (Cm_2 + Cm_2 i + Cm_1 j) \\ &\quad \times (a_{12}(m_1) + a_{22}(m_1))^j a_{21}(m_1)^i . \end{aligned} \tag{2.16}$$

Making use of the fact that for all pairs (d_1, d_2) such that $-1 < d_1 + d_2 < 1$,

$$\sum_{i=0}^{\infty} \sum_{j=0}^{\infty} \frac{(i+j)!}{i!j!} d_1^j d_2^i = 1/(1 - d_1 - d_2) ,$$

and

$$\sum_{i=0}^{\infty} \sum_{j=0}^{\infty} i \frac{(i+j)!}{i!j!} d_1^j d_2^i = d_2/(1 - d_1 - d_2)^2 ,$$

and the fact that $a_{11}(m_1) + a_{12}(m_1) + a_{21}(m_1) + a_{22}(m_1) = 1$, we obtain from (2.16) that

$$\begin{aligned}E(NF_e) &= a_{11}(m_1) \Bigg[\frac{Cm_2}{(1 - a_{12}(m_1) - a_{22}(m_1) - a_{21}(m_1))} \\ &\quad + \frac{Cm_2 a_{21}(m_1)}{(1 - a_{12}(m_1) - a_{22}(m_1) - a_{21}(m_1) - a_{21}(m_1))^2} \\ &\quad + \frac{Cm_1(a_{12}(m_1) + a_{22}(m_1))}{(1 - a_{12}(m_1) - a_{22}(m_1) - a_{21}(m_1))^2} \Bigg] \\ &= Cm_2 \frac{(a_{11}(m_1) + a_{21}(m_1))}{a_{11}(m_1)} + Cm_1 \frac{(a_{12}(m_1) + a_{22}(m_1))}{a_{11}(m_1)} . \end{aligned} \tag{2.17}$$

□

REMARK 2.1. The average number of local searches used by Master Algorithm 2.1 (NI_s) is smaller than the average number of local searches used by Master Algorithm 2.2 (NI_e) because $a_1 \geqslant a_{11}(m_1)$. However, on the average, the total number of function evaluations used by Master Algorithm 2.1 can be larger or smaller than the number of function evaluations used by Master Algorithm 2.2, depending on (i) the difficulty of the problem **GSP** (the smaller the a_1, the more difficult the problem), (ii) the accuracy of the estimation scheme, (the closer $a_{11}(m_1)$ is to a_1 and $a_{22}(m_1)$ to a_2, the more accurate the estimation scheme), (iii) the smallness of the ratio of m_1 to m_2. It is quite obvious that Master Algorithm 2.2 is definitely better than Master Algorithm 2.1, in terms of the expected number of function evaluations, when (i) **GSP** is a hard problem, i.e., a_1 is quite small relative to 1, (ii) the estimation scheme is quite accurate, and (iii) the ratio of m_1 to m_2 is quite small.

To obtain a better understanding of the superiority of Master Algorithm 2.2 over Master Algorithm 2.1, we assume that our estimation scheme has a probability of correct prediction $\tau(m_1)$, i.e.,

$$a_{11}(m_1) = \tau(m_1)a_1\,, \quad a_{12}(m_1) = (1 - \tau(m_1))a_1\,, \tag{2.18a}$$

$$a_{21}(m_1) = (1 - \tau(m_1))a_2\,, \quad a_{22}(m_1) = \tau(m_1)a_2\,. \tag{2.18b}$$

After rearranging terms, we obtain that

$$\frac{E(NF_e)}{E(NF_s)} = \left[1 - \frac{m_1}{m_2}\right] \frac{[2\tau(m_1)a_1 - \tau(m_1) - a_1 + 1/(1 - m_1/m_2)]}{\tau(m_1)}\,. \tag{2.19}$$

Let us examine the equal cost contour of $E(NF_e)/E(NF_s)$ as a function of a_1 and $\tau(m_1)$, with m_1/m_2 fixed. The equal cost contour in $[0,1] \times [0,1]$, for $E(NF_e)/E(NF_s) = \gamma$, is a hyperbola defined by

$$[\tau(m_1) - 0.5][a_1 - (1 + \gamma^*)/2] = \frac{(\gamma - 1 - m_1/m_2)}{4(1 - m_1/m_2)}\,, \tag{2.20}$$

where $\gamma^* = \gamma/(1 - m_1/m_2)$. The equal cost contours, in $[0,1] \times [0,1]$, of $E(NF_e)/E(NF_s)$, for $m_1/m_2 = 0.2$, are shown in Figure 1. We conclude from this figure

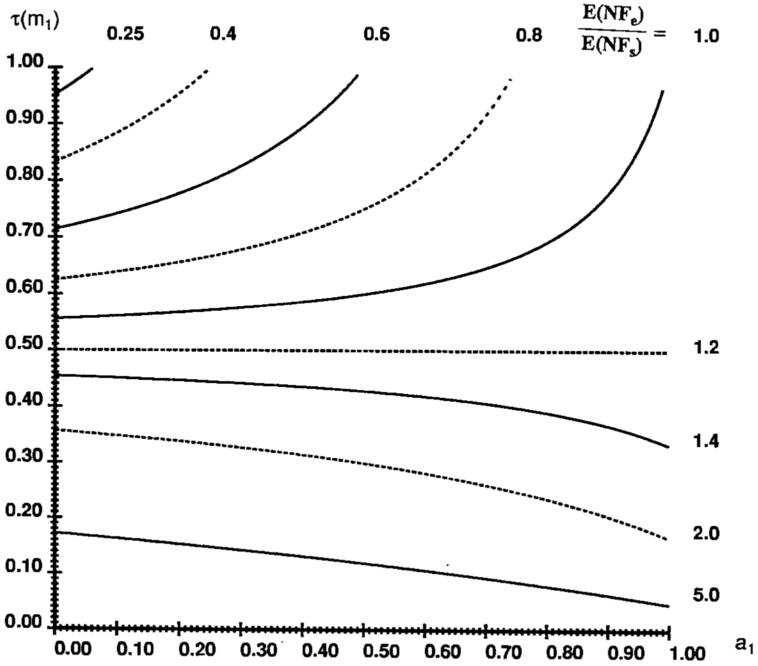

Fig. 1. Graph of equal cost contours of $E(NF_e)/E(NF_s)$.

that to obtain a benefit from using Algorithm 2.2, when the problem is easy, i.e., a_1 is near unity, we must have very good prediction as to which partition the starting point belongs to. However, as the problem becomes more difficult, i.e., a_1 becomes smaller, one can obtain considerable benefits from using Algorithm 2.2 even when the probability of correct prediction is only somewhat higher than 0.5.

3. Estimation Scheme for Linearly Convergent Algorithm

Now suppose that if $\{z_j\}_{j=0}^{\infty}$ is a sequence constructed by the local search algorithm in solving $\min_{z\in B}\Psi(z)$, then the corresponding sequence of costs $\{\Psi(z_i)\}_{i=0}^{\infty}$ is monotone decreasing and converges linearly to Ψ^*, a local minimum value, i.e., there exist constants $\theta^*\in(0,1)$ and $\beta^*>0$, such that

$$\Psi(z_i)-\Psi^*\leq\beta^*\theta^{*i}\,,\quad i=0,1,2,\ldots. \tag{3.1a}$$

We will develop a scheme for estimating the local minimum value Ψ^* and rate of convergence constants β^* and θ^*.

Suppose that the starting point $x\in B$ is given and, for $i=0,1,\ldots,m_1$, let $\Psi_i=\Psi(A^i(x))$, so that $\Psi_0>\Psi_1>\cdots>\Psi_{m1}$. The worst case situation corresponding to (3.1a) is given by the following equivalent equations:

$$\Psi_i-\Psi^*=\beta^*\theta^{*i}\,,\quad \text{for } i=0,1,\ldots,m_1\,. \tag{3.1b}$$

$$\Psi_{m_1}-\Psi^*=\theta^{*m_1-i}(\Psi_i-\Psi^*)\,,\quad \text{for } i=0,1,\ldots,m_1\,. \tag{3.1c}$$

Estimation of Ψ^* and θ^* by means of a least squares fit is not very satisfactory because (i) the resulting estimate $\hat{\Psi}$ of the local minimum value Ψ^* may be larger than Ψ_{m_1}, and (ii) the least squares fit problem is a nonlinear minimization problem. Hence we propose using the recursive process that we will now describe.[1]

Suppose that we have an estimate θ of the rate of convergence constant θ^*, then an estimate $\hat{\Psi}(\theta)$ of the local minimum value Ψ^* can be obtained by averaging the values given by (3.1c), viz:

$$\hat{\Psi}(\theta)=\frac{1}{m_1}\sum_{i=0}^{m_1-1}\frac{(\Psi_{m_1}-\theta^{m_1-i}\Psi_i)}{(1-\theta^{m_1-i})}\,. \tag{3.2}$$

On the other hand, if we have an estimate Ψ of the local minimum value Ψ^*, then we can use (3.1b) to set up a linear least squares fit problem to obtain estimates $\hat{\beta}(\Psi)$, $\hat{\theta}(\Psi)$, of the rate of convergence constants β^*, θ^*, as follows:

$$(\hat{\theta}(\Psi),\,\hat{\beta}(\Psi))=\operatorname{argmin}\sum_{i=0}^{m_1}[\log(\Psi_i-\Psi))-i\log(\theta)-\log(\beta)]^2\,. \tag{3.3}$$

Since, in the end, we only require an estimate of Ψ^* and since $\hat{\Psi}(\cdot)$, as defined by (3.2), does not depend on $\hat{\beta}$, we can discard $\hat{\beta}(\Psi)$, as defined by (3.3), and combine (3.2), (3.3) to define a map Θ: $(0, 1)\to(0, 1)$, with $\Theta(\theta) = \hat{\theta}(\hat{\Psi}(\theta))$. We propose to use any fixed point $\hat{\theta}$ of the map Θ as our estimate of the rate of convergence constant θ^*.

PROPOSITION 3.1. *The functions* $\hat{\Psi}(\cdot)$ *and* Θ *have following properties*:
(i) *The function* $\hat{\Psi}(\cdot)$ *is strictly decreasing, with* $\hat{\Psi}(0) = \Psi_{m_1}$ *and* $\hat{\Psi}(1) = -\infty$.
(ii) *For any* $\theta \in (0, 1)$,

$$\log(\Theta(\theta)) = \sum_{i>m_1/2}^{m_1} \left[(i - m_1/2)\log\left(\frac{\Psi_i - \hat{\Psi}(\theta)}{\Psi_{m_1-i} - \hat{\Psi}(\theta)}\right)\right] \Big/ [m_1(m_1 + 1) \times (m_1 + 2)/12]\,. \tag{3.4}$$

(iii) *The function* $\Theta(\cdot)$ *is strictly increasing, with* $\lim_{\theta\to 0}\Theta(\theta) = 0$ *and* $\lim_{\theta\to 1}\Theta(\theta) = 1$.

Proof. (i) Since $\Psi_{m_1} < \Psi_i$ for $i = 0, 1, \ldots, m_1 - 1$, each term in the sum in (3.2) is strictly decreasing, and hence $\hat{\Psi}(\cdot)$ is strictly decreasing. Therefore $\hat{\Psi}(0) = \Psi_{m_1}$ and $\hat{\Psi}(1) = -\infty$.

(ii) The relation (3.4) follows directly from the expression for the solution of the linear least squares problem (3.3).

(iii) Since $\hat{\Psi}(\cdot)$ is strictly decreasing and $\Psi_i < \Psi_{m_1-1}$ for $i > m_1/2$, $(\Psi_i - \hat{\Psi}(\theta))/(\Psi_{m_1-i} - \hat{\Psi}(\theta))$ is strictly increasing for $i > m_1/2$. Making use of the fact that the logarithm is a strictly increasing function, we conclude that each term on the right side of the summation in (3.4) is strictly increasing. Therefore $\Theta(\cdot)$ is strictly increasing. It now follows from (i) and (ii) that $\log(\Theta(0+)) = -\infty$ and $\log(\Theta(1-)) = 0$. Hence $\Theta(0+) = 0$ and $\Theta(1-) = 1$. □

In view of the above established properties of the map $\Theta(\cdot)$, we can propose the following bisection scheme for locating a fixed point of the map $\Theta(\cdot)$, if it exists.[2]

ESTIMATION SCHEME 3.1.

Step 0: Select initial lower and uppers bound for the fixed point of $\Theta(\cdot)$, θ_l, θ_u, respectively (e.g., $\theta_l = 0.0001$ and $\theta_u = 0.9999$).

Step 1: If $\theta_u - \theta_l < 0.0001$, then accept $\bar{\theta} = (\theta_u + \theta_l)/2$ as the estimate of the rate of convergence constant θ^*, and $\hat{\Psi}(\bar{\theta})$ as the estimate of the local minimum value Ψ^*, and stop.
Else, go to Step 2.

Step 2: Set $\theta = (\theta_u + \theta_l)/2$. If $\Theta(\theta) > \theta$, set $\theta_l = \theta$. Else, set $\theta_u = \theta$. Go to Step 1. □

REMARK 3.1. If the initial lower and upper bounds on the fixed point are such that $\Theta(\theta_l) > \theta_l$ and $\Theta(\theta_u) < \theta_u$, then $\bar{\theta}$, the result of the above estimation scheme, is an approximate fixed point, say θ^*, which is stable in the sense that there is a neighborhood of θ^*, say (θ_1, θ_2), such that for any $\theta \in (\theta_1, \theta^*)$, $\theta < \Theta(\theta) < \theta^*$, while for $\theta \in (\theta^*, \theta_2)$, $\theta^* < \Theta(\theta) < \theta$. Although one can construct defining data, in

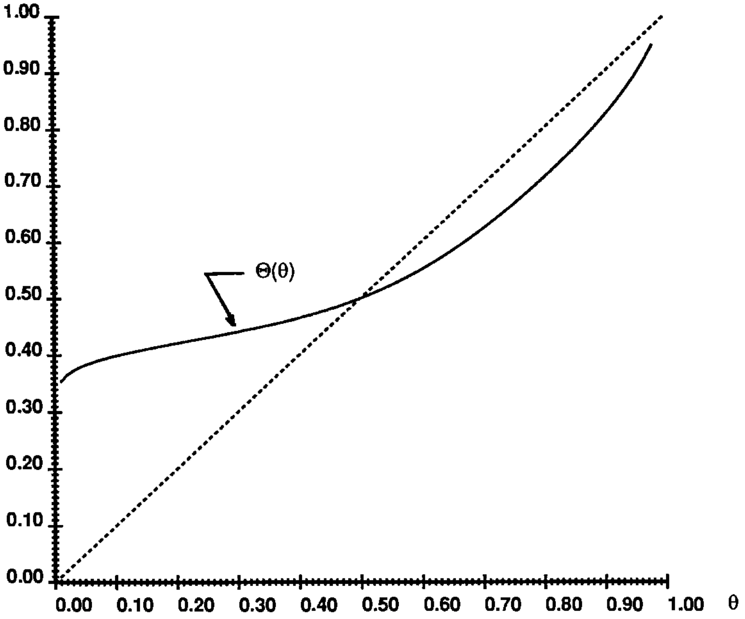

Fig. 2. Graph of the map $\Theta(\cdot)$ for Data set 1.

the form of a monotone decreasing sequence $\{\Psi_i\}_{i=0}^{m_1}$, for which $\Theta(\cdot)$ has no fixed point in $(0, 1)$, our numerical experience shows that in practice this is highly unlikely when the data is constructed by a linearly converging algorithm. Furthermore, in our numerical experience we have not encountered a case where $\Theta(\cdot)$ had more than one fixed point. □

To test the accuracy of the estimation scheme, we consider two sequences, and set $m_1 = 10$. The first sequence converges geometrically to zero and is defined by $\Psi_i = 10(0.5)^i$ for $i = 0, 1, 2, \ldots$, for the other sequence we assume that we only have the first 10 points: (68.0, 43.0, 36.0, 28.0, 21.0, 18.0, 16.0, 14.5, 13.0, 12.0, 11.5). Figures 2 and 3 show the graph of $\Theta(\cdot)$ for these two cases. We see that in both cases $\Theta(\cdot)$ has a fixed point, and that the fixed point of $\Theta(\cdot)$ associated with the first sequence is 0.5, which is the actual rate of convergence of this sequence. Figure 4 shows how accurately the linearly converging sequence, $\bar{\beta}(\bar{\theta})^i + \hat{\Psi}(\bar{\theta})$, approximates the sequence Ψ_i for the second test sequence.

4. Numerical Results

To obtain statistical information needed to compare Master Algorithm 2.1 with Master Algorithm 2.2, we carried out a set of computations to determine the range of the various quantities which determine their relative effectiveness. Our test problems include seven "classical" global optimization problems described in

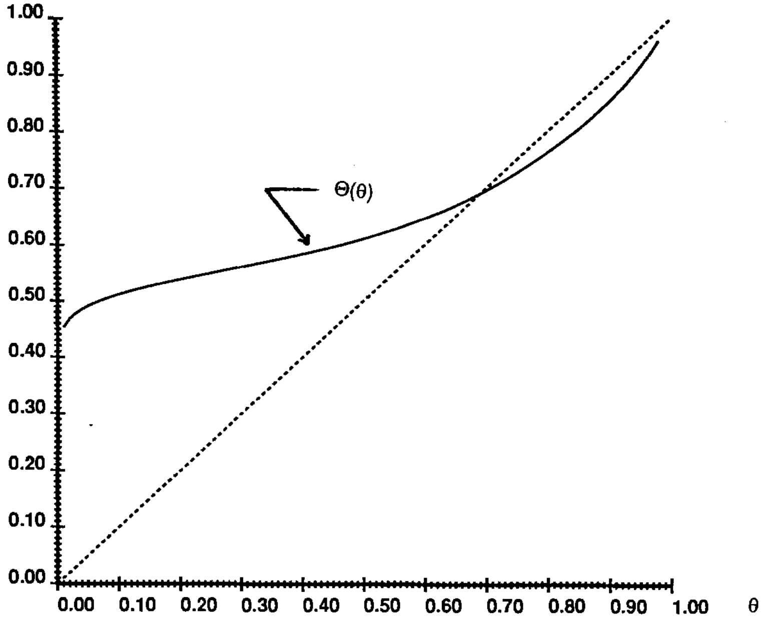

Fig. 3. Graph of the map Θ(·) for Data Set 2.

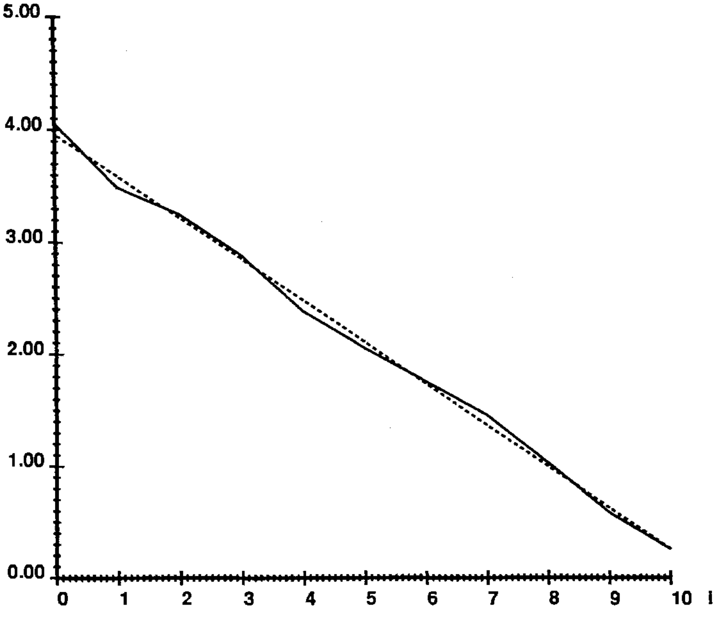

Fig. 4. Plot of log $(\psi_i - \hat{\psi}(\bar{\theta}))$ v/s i.

[6] and eighteen test problems described in [9]. The satisfycing values were chosen to be only slightly larger than the global minimum values, so as to make the satisfycing problems fairly difficult. A summary of the essential features of these test problems is shown in Table I, where the number of variables n, the number of local minima n^*, the global minimum value M (rounded to four digits), and the satisfycing values are given in the second, third, fourth, and fifth column, respectively.

In our numerical experiments, we used the local iteration algorithm $A(\cdot)$ map defined by a straightforward extension to problems with box constraints, of the Pshnichnyi–Pironneau–Polak algorithm. For problems of the form

$$\min_{x \in B} \max_{j \in \underline{l}} \max_{j \in \underline{l}} \Psi^j(x) , \tag{4.1}$$

with the $\Psi^j(\cdot)$ continuously differentiable, this algorithm computes search directions $h(x)$ according to the rule

$$h(x) \overset{\Delta}{=} \operatorname*{argmin}_{h \in B - x} \max_{j \in \underline{l}} \Psi^j(x) + \langle \nabla\Psi^j(x), h \rangle + \tfrac{1}{2}\|h\|^2 , \tag{4.2a}$$

and uses an Armijo type step size rule of the form

$$\lambda(x) \overset{\Delta}{=} \max_{k \in \mathbb{N}} \{\beta^k \mid \Psi(x + \beta^k h(x)) - \Psi(x)) \leq \beta^k \alpha\theta(x)\} , \tag{4.2b}$$

where $\mathbb{N} \overset{\Delta}{=} \{0, 1, 2, \ldots\}$, α, $\beta \in (0, 1)$ are fixed parameters and the optimality

Table I. Data for statistical analysis of Master Algorithms 2.1 and 2.2

Problem	n	n^*	M	v	$a_{11}(m_1)$	$a_{12}(m_1)$	$a_{21}(m_1)$	$a_{22}(m_1)$	m_1/m_2
SQRIN5	4	5	−10.15	−10.0	0.156	0.000	0.808	0.036	0.175
SQRIN7	4	7	−10.40	−10.0	0.704	0.001	0.234	0.060	0.200
SQRIN10	4	10	−10.54	−10.0	0.795	0.000	0.145	0.060	0.205
HARTMAN3	3	4	−3.998	−3.5	0.785	0.105	0.000	0.110	0.166
HARTMAN6	6	4	−3.322	−3.0	0.985	0.000	0.000	0.015	0.045
RCOS	2	3	0.3979	1.0	0.933	0.067	0.000	0.000	0.163
GOLDPR	2	4	3.000	4.0	0.153	0.423	0.053	0.373	0.058
LEVY1	1	3	7.000	8.0	0.780	0.000	0.000	0.220	0.259
LEVY2	1	19	14.5	−14.0	0.398	0.000	0.000	0.602	0.267
LEVY3	2	760	−186.7	−180.0	0.070	0.001	0.025	0.904	0.174
LEVY4	2	760	−186.7	−180.0	0.011	0.000	0.052	0.937	0.176
LEVY5	2	760	−186.7	−180.0	0.015	0.000	0.051	0.934	0.176
LEVY6	2	6	−1.032	−0.5	0.782	0.000	0.005	0.213	0.287
LEVY7	2	25	0.000	0.25	0.416	0.022	0.008	0.553	0.060
LEVY8	3	125	0.000	0.25	0.363	0.080	0.016	0.540	0.038
LEVY9	4	625	0.000	0.25	0.329	0.121	0.018	0.532	0.030
LEVY10	5	10^5	0.000	0.25	0.312	0.136	0.026	0.527	0.026
LEVY11	8	10^8	0.000	0.25	0.286	0.164	0.034	0.506	0.023
LEVY12	10	10^{10}	0.000	0.25	0.284	0.202	0.040	0.474	0.022
LEVY13	2	900	0.000	0.25	0.058	0.002	0.016	0.924	0.021
LEVY14	3	2700	0.000	0.25	0.010	0.012	0.014	0.964	0.016
LEVY15	4	71000	0.000	0.25	0.002	0.006	0.014	0.978	0.014

function $\theta(x)$ is defined by

$$\theta(x) \overset{\Delta}{=} \min_{h \in B-x} \max_{j \in \underline{l}} \Psi^j(x) + \langle \nabla\Psi^j(x), h \rangle + \tfrac{1}{2}\|h\|^2 \,. \tag{4.2c}$$

Hence for any $x \in B$,

$$A(x) = x + \lambda(x)h(x) \,. \tag{4.3}$$

We used the scheme proposed in Section 3, with $m_1 = 10$, to estimate the values of $\Psi(x) = \max_{j \in \underline{l}} \Psi^j(x)$ at the local minimizers.

For each test problem, several hundred local searches were started from randomly generated points, using a uniform distribution on B. Due to computing time constraints, the maximum number of local searches per test problem was limited to 2500. In order to obtain sufficient data for our estimates, we performed $N = \min\{2500,\ 100 \times \text{number of known local minima}\}$ local searches for each problem. Since the local searches were continued until the stopping rule, $|\theta(x)| \leq 10^{-5}$, was satisfied, it was possible to determine whether a starting point x_i was in B_1 or B_2.

To estimate the $a_{ij}(m_1)$, we used the formula

$$a_{ij}(m_1) = \frac{\text{number of initial points tried in } B_{ij}(m_1)}{\text{total number of local searches}} \,. \tag{4.4}$$

We obtained an estimate of m_2 by averaging the number of iterations used in the local searches.

Columns 6–10 of Table I show the estimates of $a_{11}(m_1)$, $a_{12}(m_1)$, $a_{22}(m_1)$, and the ratio of m_1 to m_2. All the numbers in the last five columns of Table I were rounded to 3 digits after the decimal point.

For Table II, we computed the estimates of $E(NF_e)/E(NF_s)$ from the estimates of the $a_{ij}(m_1)$, Table I, and of m_2, according to (2.15). The actual values of $E(NF_e)/E(NF_s)$ were computed by applying the Master Algorithms 2.1 and 2.2 N times to each of the satisfycing problems in our tables, where N was as used for computing the estimates of these quantities. The values in the last four columns of Table II were calculated from the data in Table I. Referring to Table II, we observe as follows:

(i) The estimated values of $E(NF_e)/E(NF_s)$ are very close to the actual values of $E(NF_e)/E(NF_s)$, except for Levy 7–Levy 12, where the actual values are substantially smaller (i.e., actual performance much better) than the estimated values.

(ii) With the exception of SQRIN5, the classical test problems in [6] are very easy, i.e., $a_1 = a_{11}(m_1) + a_{12}(m_1)$ is quite large, while most of Levy's problems are quite difficult.

(iii) For the hard problems, the expected number of function evaluations used by Master Algorithm 2.1 is about 4 times the expected number of function evaluations used by Master Algorithm 2.2 (our new algorithm). On easy problems Master Algorithm 2.2 is only slightly better than Master Algorithm 2.1.

Table II. An evaluation of the relative effectiveness of Master Algorithms 2.1 and 2.2

Problem	Actual $\frac{E(NF_e)}{E(NF_s)}$	Estimated $\frac{E(NF_e)}{E(NF_s)}$	a_1	a_{11}/a_1	a_{22}/a_2	$a_{11} + a_{22}$
SQRIN5	0.958	0.970	0.156	1.000	0.043	0.192
SQRIN7	0.965	0.953	0.706	0.998	0.204	0.764
SQRIN10	0.970	0.952	0.795	1.000	0.293	0.855
HARTMAN3	0.859	0.931	0.890	0.882	1.000	0.895
HARTMAN6	0.998	0.986	0.985	1.000	1.000	1.000
RCOS	1.010	1.012	1.000	0.933	1.000	0.933
GOLDPR	0.845	0.945	0.575	0.265	0.877	0.525
LEVY1	0.801	0.837	0.780	1.000	1.000	1.000
LEVY2	0.575	0.559	0.398	1.000	1.000	1.000
LEVY3	0.327	0.255	0.071	0.989	0.973	0.974
LEVY4	0.307	0.228	0.011	1.000	0.947	0.948
LEVY5	0.313	0.230	0.015	1.000	0.948	0.949
LEVY6	0.697	0.848	0.782	1.000	0.977	0.995
LEVY7	0.273	0.484	0.438	0.950	0.985	0.970
LEVY8	0.166	0.492	0.443	0.819	0.971	0.904
LEVY9	0.146	0.502	0.450	0.731	0.967	0.861
LEVY10	0.149	0.509	0.448	0.696	0.955	0.448
LEVY11	0.160	0.539	0.470	0.609	0.955	0.792
LEVY12	0.173	0.580	0.480	0.585	0.921	0.758
LEVY13	0.078	0.097	0.060	0.973	0.983	0.982
LEVY14	0.085	0.088	0.022	0.446	0.974	0.974
LEVY15	0.127	0.147	0.008	0.200	0.986	0.980

(iv) The last three columns of Table II enable us to evaluate the reliability of our Estimation Scheme 3.1 on the problems tested. We note that $a_{11}(m_1)/a_1$ is the fraction of times we were able to establish correctly than an initial $x \in B_1$ is in fact in B_1 ; $a_{22}(m_1)/a_1$ is the fraction of times we were able to establish correctly that an initial $x \in B_2$ is in fact in B_2; and finally, $a_{11}(m_1) + a_{22}(m_1)$ is the fraction of times we correctly identified whether an initial point x is in B_1 or in B_2. We see that our prediction success rate, averaged over the test problems, is well over 80%, and that it is particularly good on Levy's problems. □

5. Conclusions

We have presented a new multistart method for solving global satisfycing problems. The novel feature of this method is the utilization of the fact that many minimax algorithms converge linearly, which makes it possible to estimate the local minimum value to which costs generated by a minimax algorithm converge. Hence, when the projected local minimum value exceeds the satisfycing value, the local search is aborted. Our statistical data indicate that the new method is much superior to an obvious adaptation of a classical multistart, global optimization method, whenever the satisfycing problem is difficult and our local minimum estimation scheme is fairly accurate. Our numerical experience show that in most cases our estimation scheme is indeed quite accurate and hence the new method should prove a valuable addition to the decision maker's toolbox.

Acknowledgements

We are indebted to an anonymous referee for pointing out an error in our original version of Proposition 2.2 and for providing us with a correct set of formulae.

The research reported herein was sponsored in part by the National Science Foundation grant ECS-8713334, the Air Force Office of Scientific Research contract AFOSR-86-0116 and the State of California MICRO Program grant 532410-19900.

Notes

[1] Our estimation scheme seems to have elements in common with acceleration schemes for linearly converging sequences, see, e.g. [11].

[2] Another way of finding a fixed point of $\Theta(\cdot)$ is to generate a sequence θ_0, θ_1, $\theta_2, \ldots$ numerical experiments show that this approach is very slow due to the fact that $\Theta(\theta) - \theta$ is quite flat around the fixed point.

References

1. F. Archetti and F. Schoen (1984), A Survey on the Global Optimization Problem: General Theory and Computational Approaches, *Annals of Operations Research* **1**, 87–110.
2. B. Betro and F. Schoen (1987), Sequential Stopping Rules for the Multistart Algorithm in Global Optimization, *Math. Prog.* **38**, 271–286.
3. C. G. E. Boender, A. H. G. Rinnooy Kan, G. T. Timmer, and L. Stougie (1982), A Stochastic Method for Global Optimization, *Math. Prog.* **22**, 125–140.
4. C. G. E. Boender and A. G. H. Rinnooy Kan (1982), Bayesian Stopping Rules for Multistart Global Optimization Methods, *Math. Prog.* **37**, 59–80.
5. R. H. Byrd, C. L. Dert, A. H. G. Rinnooy Kan, and R. B. Schnabel (1990), Concurrent Stochastic Methods for Global Optimization, *Math. Prog.* **46**, 1–30.
6. L. C. Dixon and G. P. Szego, eds. (1978), *Towards Global Optimization II*, North-Holland, Amsterdam.
7. W. Feller (1961), *Introduction to Probability Theory and Its Applications*, Wiley, NY.
8. L. He and E. Polak (1990), Effective Diagonalization Strategies for the Solution of a Class of Optimal Design Problems, *IEEE Trans. Auto. Control* **35**, 258–267.
9. A. V. Levy, A. Montalvo, S. Gomez, and A. Calderon (1981), Topics in Global Optimization, *Numerical Analysis*, Lecture notes in Math. 909, pp. 18–33.
10. J. G. March and H. A. Simon (1958), *Organizations*, Wiley, New York.
11. A. C. Matos (1990), Acceleration Methods Based on Convergence Tests, *Num. Math.* **58**, 329–340.
12. O. Pironneau and E. Polak (1972), On the Rate of Convergence of Certain Methods of Centers, *Mathematical Programming* **2**(2), 230–258.
13. E. Polak (1971), *Computational Methods in Optimization: A Unified Approach*, Academic Press, New York.
14. E. Polak (1987), On the Mathematical Foundations of Nondifferentiable Optimization in Engineering Design, *SIAM Review* **29**(1), 21–91.
15. E. Polak (1989), Basics of Minimax Algorithms, in F. H. Clarke, V. F. Dem'yanov, and F.

Giannessi eds., *Nonsmooth Optimization and Related Topics*, pp. 343–367, Plenum Press, New York.

16. E. Polak and L. He (1992), Rate Preserving Discretization Strategies for Semi-Infinite Programming and Optimal Control, *SIAM J. Control & Optimization* **30**(3), 548–572.
17. E. Polak and L. He (1991), Finite-Termination Schemes for Solving Semi-Infinite Satisfycing Problems, *J. Optimization Theory and Applications* **70**(3), 429–466.
18. B. N. Pshenichnyi and Yu. M. Danilin (1975), *Numerical Methods in External Problems* (*Chislennye metody v ekstremal'nykh zadachakh*), Nauka, Moscow.
19. A. H. G. Rinnooy Kan and G. T. Timmer (1987), Stochastic Global Optimization Methods Part I: Clustering Methods *Math. Prog.* **39**, 27–56.
20. A. H. G. Rinnooy Kan and G. T. Timmer (1987), Stochastic Global Optimization Methods Part II: Multi Level Methods, *Math. Prog.* **39**, 57–78.
21. A. Torn (1977), Cluster Analysis Using Seed Points and Density-Determined Hyperspheres with an Application to Global Optimization, *IEEE Trans. on Systems, Man and Cybernetics* **7**(8), 610–615.
22. A.P. Wierzbicki (1982), A Mathematical Basis for Satisfycing Decision-Making, *Mathematical Modeling* **3**(5), 391–405.
23. R. Zielinski (1981), A Statistical Estimate of the Structure of Multi-Extremal Problems, *Math. Prog.* **21**, 348–356.