Global Optimization for Imprecise Problems

M.N. Vrahatis, D.G. Sotiropoulos and E.C. Triantafyllou
Department of Mathematics, University of Patras, GR–261.10, Patras, Greece
E-mail: vrahatis@math.upatras.gr

Abstract. A new method for the computation of the global minimum of a continuously differentiable real–valued function f of n variables is presented. This method, which is composed of two parts, is based on the combinatorial topology concept of the degree of a mapping associated with an oriented polyhedron. In the first part, interval arithmetic is implemented for a "rough" isolation of all the stationary points of f. In the second part, the isolated stationary points are characterized as minima, maxima or saddle points and the global minimum is determined among the minima. The described algorithm can be successfully applied to problems with imprecise function and gradient values. The algorithm has been implemented and tested. It is primarily useful for small dimensions ($n \leq 10$).

Key words: Generalized bisection method, interval arithmetic, characterization of stationary points, global optimization, imprecise problems, noisy functions

1. Introduction

Several methods for finding the extrema of a function $f: \mathcal{D} \subset \mathbb{R}^n \to \mathbb{R}$, where $\mathcal{D}$ is open and bounded, have been proposed with many applications in different scientific fields (mathematics, physics, engineering, computer science etc.). Most of them require precise function and gradient values. In many applications though, precise values are either impossible or time consuming to obtain. For example, when the function and gradient values depend on the results of numerical simulations, then it may be difficult or impossible to get very precise values. Or, in other cases, it may be necessary to integrate numerically a system of differential equations in order to obtain a function value, so that the precision of the computed value is limited [29, 48]. On the other hand, it is necessary, in many applications, to use methods which do not require precise values [48, 24], as for example in neural network training [31, 32, 30]. Furthermore, in many problems the values of the function to be minimized are computationally expensive [23]. Such problems are common in real life applications as in the optimization of parameters in chemical experiments or finite element calculations, where a single measurement (function evaluation) takes several hours or even days. With such applications in mind, we ask for robust methods which make good progress with the fewest possible number of function evaluations, while the work to select new evaluation points can be neglected.

Developments in Global Optimization, I.M. Bomze, T. Csendes, R. Horst and P.M. Pardalos (eds.), Chapter **3**, pp.37-54, Kluwer Academic Publishers (Nonconvex Optimization and Its Applications series, vol.18), Dordrecht, The Netherlands, 1997 [ISBN: 0-7923-4351-4].

In this contribution a new method is presented for the computation of the global minimum x^* of an n–dimensional real valued function f. The proposed algorithm makes use of interval arithmetic techniques [2, 34, 19] in order to isolate "roughly" the stationary points of f. Next, a criterion [54] is used to characterize the isolated stationary points as minima, maxima, or saddle points. This criterion implements topological degree theory and especially the concept and properties of the characteristic n–polyhedron [51, 44, 47], by which all calculations concerning the exact value of the topological degree are avoided, and, therefore, it can be applied to problems with imprecise values. Finally, the localized minima are computed and the global one is decided.

In Section 2 we give a discussion of optimization of noisy functions as well as a simulation of the influence of noise (proportional to a Gaussian distributed random number with zero mean and various variances). In Section 3 a detailed description of the new method and a model algorithm are presented, while in Section 4 numerical results are presented in tabular form. We finally end, in Section 5, with some concluding remarks and a short discussion for further research.

2. Optimization of noisy functions

The problem of optimization of noisy or imprecise (not exactly known) functions occurs in various applications, as for instance, in the task of experimental optimization. Also, the problem of locating local maxima and minima of a function from approximate measurement results is vital for many physical applications: In spectral analysis, chemical species are identified by locating local maxima of the spectra. In radioastronomy, sources of celestial radio emission and their subcomponents are identified by locating local maxima of the measured brightness of the radio sky. Elementary particles are identified by locating local maxima of the experimental curves [43].

The theory of local optimization provides a large variety of efficient and effective methods for the computation of an optimizer of a smooth function f. For example, Newton–type and quasi–Newton methods show superlinear convergence in the vicinity of a nondegenerate optimizer. However, these methods require the gradient or the Hessian, respectively, in contrast to other optimization procedures, like the simplex method [40, 33] or the direction set method of Powell (see [11, pp. 87–92]). Nevertheless, it is believed that a quasi–Newton method using a finite–difference approximation for the gradient is the most efficient one for the task of optimizing smooth functions when only function values are available.

In some applications, however, the function to be minimized is only known within some (often unknown and low) precision. This might be due to the fact that evaluation of the function means measuring some physical or chemical quantity or performing an (expensive) finite element calculation in order to solve partial differential equations. The function values obtained are corrupted by noise, namely stochastic measurement errors or discretization errors. This means that, although the underlying function is smooth, the function values available show a discontinuous behavior. Moreover, no gradient information is available.

For small variations in a neighborhood of a point the corresponding function values reflect the local behavior of the noise rather than that of the function. Thus, a finite–difference procedure to estimate the gradient, needed for a quasi–Newton method, fails.

The traditional method for optimizing noisy functions is the simplex or polytope method, originally given by Spendley, Hext and Himsworth, [40] and later developed by Nelder and Mead [33] (cf. [35, p.202], [11, p.18], [37, p.230]). This method surpasses other well–known optimization methods when dealing with the large noise case. However, this is not valid in the noiseless case [12, p.92]. The ability of this method to cope with noise is due to the fact that it proceeds solely by comparing the relative size of the function values. It does not use a local model of the function f and works without the assumption of continuity. Although this method has poor convergence properties (for a convergence proof of a modified version see [41]), yet it has been proved to be a useful method in many applications. The simplex method is assumed to have converged whenever the standard deviation of the function at the vertices of the current simplex is smaller than some prescribed small quantity. Deficiency of this method can occur when the current simplex is very "flat". This can be avoided by suitable variants (see for example [12, 41]). More sophisticated methods in this direction are discussed by Powell [38]. Also, a method which combines a simplex technique, similar to the method of Nelder and Mead, with the use of occasional quadratic models, estimated by some kind of finite differences, is given by Dixon [8].

Although a smooth function corrupted by noise is effectively discontinuous, one may use the smoothness of the underlying function and try to separate it from the added noise. A method of this type was suggested by Glad and Goldstein [13] who repeatedly estimated quadratic models by the method of least squares using $O(n^2)$ points in a fixed scaled pattern around the best point. Although this allows them to prove convergence of their algorithm, it makes each iteration unduly expensive. Instead of using extra function evaluations at points arranged in a fixed pattern, Karidis and Turns [23] suggested a method

based on using the previous function evaluations for the least square fit, thus improving an approach of Winfield [55], who uses interpolation (which is sensitive to noise).

Recently, an effective method which is comparable to quasi– Newton methods in the noiseless case and is much more robust than the method of Nelder and Mead in the noisy case, has be given by Elster and Neumaier [10]. It is based on the use of quadratic models and the restriction of the evaluation points to successively refined grids. This algorithm is particularly suitable for cases where the obtained function values are corrupted by (deterministic or stochastic) noise and the underlying function is smooth.

Another approach to handle noisy cases is to employ methods that require only the least amount of information regarding the function values, namely their signs. Recently, we have begun to study rootfinding and optimization methods which can be applied to problems with imprecise function and gradient values. To this end we have proposed methods for which the only computable information required is the algebraic signs of the considered functions [51, 44, 45, 14, 15, 16, 17]. For the unconstrained optimization case we have proposed a method based on a modified one–dimensional bisection rootfinding method which actually requires only the signs of the function values to be correct [48]. Also, we have proposed a method named dimension reducing for the computation of a local minimum [18]. This method can be applied to problems with imprecise function and gradient values and incorporates the advantages of Newton and nonlinear SOR algorithms. Specifically, although this method uses reduction to simpler one–dimensional nonlinear equations, it quadratically minimizes general functions.

To study the influence of the imprecise information (regarding the values of the objective function and the gradient), we simulate imprecisions with the following approach. Information about $f(x)$ is obtained in the form of $f^{\sigma}(x)$, where $f^{\sigma}(x)$ is an approximation to the true function value $f(x)$, contaminated by a small amount of noise η such as:

$$|f^{\sigma}(x) - f(x)| \leq \delta. \tag{1}$$

Specifically, the function values are obtained as:

$$f^{\sigma}(x) = f(x)(1+\eta), \qquad \eta \sim N(0, \sigma^2), \tag{2}$$

where $N(0, \sigma^2)$ denotes a Gaussian distributed random number with zero mean and variance σ^2, i.e., relative stochastic errors are used for the test problems. To obtain η we apply the method of Box and Muller [6] using various variances σ.

As it is generally known and many times confirmed, in the noisy case (with $\sigma = 0.01$), a quasi–Newton approach, using the standard

finite difference approximation for the gradient, breaks down [10]. For the behavior of optimization methods applied to noisy functions as well as studies of the performance of optimization methods by visualization see [3, 4].

3. Locating, characterizing and computing the stationary points of a function

The proposed method computes the global minimum x^* (and all the global minimizers) of a real–valued function f of n variables in the box $\mathbf{X}$ such that:

$$f(x^*) = \min_{x \in \mathbf{X}} f(x), \tag{3}$$

where the objective function $f : \mathbb{R}^n \to \mathbb{R}$ has continuous first and second derivatives and $\mathbf{X} \in \mathbb{I}^n$ is an n–dimensional interval.

Our method distinguishes the minima among all the stationary points by applying a new characterization criterion. The stationary points of a function $f(x)$ in the box $\mathbf{X}^0$ are the zeros of the set of equations:

$$g(x) = \nabla f(x) = \mathcal{O} = (0, 0, \ldots, 0), \tag{4}$$

where $\nabla f = \left(\frac{\partial f}{\partial x_1}, \ldots, \frac{\partial f}{\partial x_n}\right)$ denotes the gradient of f. Of course, Equation (4) is fullfiled for the global minimum as well as for local minima, maxima and saddle points.

Our algorithm isolates and characterizes one by one all the stationary points. In cases where a stationary point is characterized as a minimum the algorithm computes it by applying a real valued generalized bisection method [51, 44, 47]. This bisection method is a global convergent method and can be applied for any box of arbitrary size which includes a stationary point. Thus, we only require a "rough" isolation of the stationary points. So, in order to roughly isolate these points we use an interval–Newton method which serves as an existence as well as a uniqueness test [26, 19]. We wish to avoid wasteful iterations of the interval–Newton method to isolate a stationary point of f which is not the global minimum. Since the interval–Newton method is costly in computational effort (it requires evaluation of the Jacobian of $g(x)$ and various algebraic procedures), we first apply a simple monotonicity test which may preclude the need of the above method. Specifically, if $\mathbf{X}$ is a sub–box of $\mathbf{X}^0$, we evaluate the components of g over $\mathbf{X}$ and, if the resulting interval vector does not include $\mathcal{O}$, then we conclude that there is no stationary point of f in $\mathbf{X}$. If the test of the gradient

is unsuccessful in deleting $\mathbf{X}$, we apply the interval–Newton method. Then System (4) is linearized to the form:

$$g'(\mathbf{X}_k)(\overline{\mathbf{X}}_k - \check{X}_k) = -g(\check{X}_k), \tag{5}$$

where $g'(\mathbf{X}_k)$ represents an appropriate interval extension of the Jacobian matrix of g over the box $\mathbf{X}_k$ and $\check{X}_k$ is the midpoint of the interval vector $\mathbf{X}_k$.

To simplify the above notations we denote $\mathbf{X}_k$ by $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)$, and set $\mathbf{A} = \{\mathbf{A}_{i,j}\}_{i,j=1}^{n}$ with $\mathbf{A}_{i,j}$ to be the interval in the ith row and jth column of $\mathbf{A} = g'(\mathbf{X})$, $g(\check{X}_k) = g = (\mathbf{g}_1, \mathbf{g}_2, \ldots, \mathbf{g}_n)$ and $\check{X}_k = M = (m_1, m_2, \ldots, m_n)$. With these notations Equation (5) can be written in the following form:

$$\mathbf{A}(\overline{\mathbf{X}}_k - M) = -g. \tag{6}$$

Then, applying an interval version of the Gauss–Seidel method, [27, 34], with starting guess $\mathbf{X}_k$ is a competitive way to compute the components of $\overline{\mathbf{X}}_k$ on a row–by–row basis according to the relation:

$$\overline{\mathbf{x}}_i = m_i - \left[\mathbf{g}_i + \sum_{\substack{j=1 \\ j \neq i}}^{n} \mathbf{A}_{ij}(\mathbf{x}_j - m_j) \right] \Big/ \mathbf{A}_{ii}. \tag{7}$$

The interval Gauss–Seidel linear solver possesses properties which allow it to function as an existence and uniqueness test and to couple the interval–Newton method to a bisection procedure [28]. These desirable characteristics of the Gauss–Seidel iteration follow from a proof presented in [34]. Applying the Gauss–Seidel linear solver we are able to obtain a root inclusion test to each sub–box by assigning the values "true", "false" or "unknown" as follows: The value true is assigned if, after applying Relation (7), it holds that $\overline{\mathbf{X}}_k \subseteq \mathbf{X}_k$, which means that System (4) has a unique solution in $\mathbf{X}_k$. If, on the other hand, $\overline{\mathbf{x}}_i \cap \mathbf{x}_i = \emptyset$ for any i, then there are no solutions to System (4) in $\mathbf{X}_k$, and the Gauss–Seidel method, acting as a root inclusion test, returns the value "false". Finally, if the Gauss–Seidel method returns neither "true" nor "false", then the root inclusion test is inconclusive, and the procedure returns the value "unknown". When the Gauss–Seidel method returns "unknown" either the value of $\mathbf{X}_k$ is set equal to that of $\overline{\mathbf{X}}_k$ and another Gauss–Seidel iteration is performed, or one of the coordinate intervals of $\mathbf{X}_k$ is bisected to form two sub–boxes. In this way, a bisection process enables an interval–Newton method to isolate all the stationary points of f in the initial interval $\mathbf{X}^0$, (for more details see [26]).

Using the above way we are able to maintain two kinds of lists: a list $\mathcal{L}$ of boxes $\mathbf{X}$ to be processed and a list $\mathcal{U}$ of boxes for which this stage of the algorithm cannot decide whether they contain a unique stationary point. These boxes will be processed at the next stage of the algorithm.

In the second phase of the algorithm, the isolated stationary points are characterized as minima, maxima or saddle points [54]. This procedure is based on the concept and properties of the characteristic n–polyhedron (CP) [51, 44, 47].

First, we give some preliminary concepts and theoretical background necessary for the description of the characterization procedure. Let us define a characteristic n–polyhedron by constructing the $2^n \times n$ matrix $\mathcal{M}_n$ whose rows are formed by all possible combinations of $-1,\ 1$, which is called n–*complete* matrix. Let $\mathit{\Pi}^n = \langle v^1, v^2, \ldots, v^{2^n} \rangle$ be an oriented n–polyhedron in $\mathbb{R}^n$ with 2^n vertices. The *matrix of signs associated with* ∇f *and* $\mathit{\Pi}^n$, denoted by $\mathcal{S}(\nabla f;\ \mathit{\Pi}^n)$, is the $2^n \times n$ matrix whose entries in the kth row are the corresponding coordinates of the vector:

$$\mathrm{sgn}\nabla f(v^k) = \Big(\mathrm{sgn}\nabla f_1(v^k), \mathrm{sgn}\nabla f_2(v^k), \ldots, \mathrm{sgn}\nabla f_n(v^k)\Big), \quad (8)$$

where sgn defines the well–known sign function. Then the n–polyhedron $\mathit{\Pi}^n = \langle v^1, v^2, \ldots, v^{2^n} \rangle$ in $\mathbb{R}^n$ is a *characteristic polyhedron (CP) relative to* ∇f if the matrix $\mathcal{S}(\nabla f;\ \mathit{\Pi}^n)$ is identical with the n–complete matrix $\mathcal{M}_n$ [51]. In other words, the signs of the components of ∇f at the 2^n vertices of $\mathit{\Pi}^n$ yield every combination of ± 1.

If $\mathit{\Pi}^n$ is a CP then, under suitable assumptions on the boundary of $\mathit{\Pi}^n$, the value of the topological degree of ∇f at $\mathcal{O}$ relative to $\mathit{\Pi}^n$ is nonzero which implies the existence of a stationary point inside $\mathit{\Pi}^n$ (CP–criterion) [44, 45, 51]. It is important to note that the CP–criterion avoids all calculations concerning the topological degree since it requires not its exact value but only its nonzero value.

Next, we proceed with the description of the characterization procedure. From the first part of the algorithm we have a list of boxes each of which contains a unique stationary point x^{st} of f and so forms a characteristic n–polyhedron (CP), according to the CP–criterion. Each stationary point can be characterized according to the orientation of the vertices of the CP [54] by following the procedure outlined below.

First, the oriented CP $= \langle v^1, v^2, \ldots, v^{2^n} \rangle$ is transformed so that its n proper 1–simplexes with a common vertex are edges of the polyhedron with vertices formed by the rows of the following defined $2^n \times n$ matrix $\mathcal{R}$: If

$$x_j^{\min} = \min\{v_j^1, v_j^2, \ldots, v_j^{2^n}\}, \qquad x_j^{\max} = \max\{v_j^1, v_j^2, \ldots, v_j^{2^n}\} \quad (9)$$

are the minimum and maximum of all the jth components v_j^i of the CP vertices v^i, $i = 1, 2, \ldots, 2^n$, respectively, $\mathcal{G}$ is the rank–1 $2^n \times n$ matrix with elements in the jth column the value $x_j^{\min}$ and $\mathcal{B}$ is the $n \times n$ diagonal matrix having as ith element the difference $h_i = x_i^{\max} - x_i^{\min}$, then we define the matrix $\mathcal{R} = \mathcal{G} + \mathcal{M}_n^* \mathcal{B}$. For example for $n = 2$ we have:

$$\mathcal{R} = \mathcal{G} + \mathcal{M}_2^* \mathcal{B} = \begin{bmatrix} x_1^{\min} & x_2^{\min} \\ x_1^{\min} & x_2^{\min} \\ x_1^{\min} & x_2^{\min} \\ x_1^{\min} & x_2^{\min} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} h_1 & 0 \\ 0 & h_2 \end{bmatrix} = \begin{bmatrix} x_1^{\min} & x_2^{\min} \\ x_1^{\min} & x_2^{\max} \\ x_1^{\max} & x_2^{\min} \\ x_1^{\max} & x_2^{\max} \end{bmatrix}$$

In the sequel, the matrix $\mathcal{S}(\nabla f; \mathcal{R})$ is constructed and the following cases can be distinguished:

i) If 2^{n-1} rows of $\mathcal{S}(\nabla f; \mathcal{R})$ with the same sign in one of their columns are identical with the corresponding rows of $\mathcal{M}_n$, then x^{st} is characterized as a local minimum.

ii) If 2^{n-1} rows of $\mathcal{S}(\nabla f; \mathcal{R})$ with the same sign in one of their columns are identical with the corresponding rows of $-\mathcal{M}_n$, then x^{st} is characterized as a local maximum.

iii) Otherwise, if these rows are not identical either with the corresponding rows of $\mathcal{M}_n$ or $-\mathcal{M}_n$, then x^{st} is characterized as a saddle point.

This procedure makes use only of the algebraic sign of ∇f, while derivatives of ∇f or approximations of them are not required. Thus it can be applied to problems with imprecise values.

Next, the algorithm chooses those points characterized as minima and computes all of them within a given accuracy to obtain the global minimum. To this end, it uses a generalized bisection method which requires only the signs of the gradient values to be correct and thus it can be applied to imprecise problems. Also, it is a globally convergent method and can be applied to non–differentiable continuous functions [44, 45, 51]. Furthermore, this method has been successfully applied to imprecise and difficult problems (see for example [9, 46, 49, 50, 52]).

This generalized bisection method, used in combination with the CP–criterion outlined above, bisects a CP in such a way that the new refined n–polyhedron is also a CP. To do this, one computes the midpoint of a proper 1–simplex (edge) of Π^n and uses it to replace that vertex of Π^n for which the vector of their signs are identical (see [44, 45, 51, 47] for details and extensions). Finally, the number B of characteristic bisections of the edges of a Π^n required to obtain a new

refined CP, $\Pi_\star^n$, whose longest edge length, $\Delta(\Pi_\star^n)$, satisfies $\Delta(\Pi_\star^n) \leq \varepsilon$, for some $\varepsilon \in (0,1)$, is given by:

$$B = \left\lceil \log_2 \left(\Delta(\Pi^n)\, \varepsilon^{-1} \right) \right\rceil , \tag{10}$$

where the notation $\lceil \cdot \rceil$ refers to the smallest integer, which is not less than the real number quoted (see [44] for a proof).

Using the above procedures we propose the following model algorithm:

Algorithm 1. (Locating, characterizing and computing the stationary points of a function)

1. Initialize list $\mathcal{L}$ by placing the initial search region $\mathbf{X}^0$ into it.
2. **Do while** $\mathcal{L} \neq \emptyset$;
   - 2.1 Remove the first box $\mathbf{X}$ from $\mathcal{L}$;
   - 2.2 Do one of the following:
     - (a) Reject $\mathbf{X}$ (when the range of ∇f over $\mathbf{X}$ does not contain zero);
     - (b) Use an interval–Newton method to do one of the following:
       - i) reduce the size of $\mathbf{X}$ ($\mathbf{X} \leftarrow \overline{\mathbf{X}} \cap \mathbf{X}$) ;
       - ii) discard $\mathbf{X}$ ($\overline{\mathbf{X}} \cap \mathbf{X} = \emptyset$);
       - iii) verify that a unique stationary point exists in $\mathbf{X}$ ($\overline{\mathbf{X}} \subseteq \mathbf{X}$);
     - (c) If a unique stationary point is isolated, then characterize it. If it is characterized as a minimum then compute it within the accuracy ε, otherwise discard the box $\mathbf{X}$. Get the next box from the list $\mathcal{L}$;
     - (d) In the case where the root inclusion test is inconclusive then subdivide $\mathbf{X}$ to make it more likely to succeed at rejecting, reducing, or verifying uniqueness and put the corresponding sub–boxes into list $\mathcal{L}$;

end{while}

end{algorithm}

4. Numerical Applications

The above procedures were implemented using a new portable Fortran program named IMINBIS, which has been applied to several test functions. For the rough isolation of the stationary points we have applied

the interval–Newton/bisection package INTBIS due to Kearfott and Novoa [28].

IMINBIS was tested on a HP–715 workstation as well as on a PC IBM compatible. Our experience is that the algorithm behaves predictably and reliably. The results were satisfactory without any redundant function evaluations. Some typical computational results are given below. For the following problems, the reported parameters are:

— n dimension,

— $\mathbf{X}^0$ starting box,

— σ the value of the variance of the simulated noise,

— x^{st} an n–dimensional stationary point,

— IFC the total number of interval function calls to determine the range of the function,

— IJC the total number of interval Jacobian calls,

— IFNC the total number of interval function calls to do the interval–Newton method,

— RFE1 the total number of real function evaluations for the localization portion of our algorithm,

— RFE2 the total number of real function evaluations for the characterization portion of our algorithm,

— RFE3 the total number of real function evaluations for the computation of the minima within the accuracy $\varepsilon = 10^{-8}$.

Here we exhibit results from the following test cases.

Example 1. *Himmelblau function*, [20]. In this case f is given by:

$$f(x) = \left(x_1^2 + x_2 - 11\right)^2 + \left(x_1 + x_2^2 - 7\right)^2.$$

This function has the following stationary points:

a) four global minima

$$\begin{aligned} x_1^{\mathrm{st}} &= (3, 2), \\ x_2^{\mathrm{st}} &= (3.584428340330, -1.848126526964), \\ x_3^{\mathrm{st}} &= (-3.779310253378, -3.283185991286), \\ x_4^{\mathrm{st}} &= (-2.805118086953, 3.131312518250), \end{aligned}$$

b) one maximum $x_5^{\mathrm{st}} = (-0.270844590667,\ -0.923038556480)$, and

Table I. Himmelblau function, $n = 2$.

x_i^{st}	RFE1	RFE2	RFE3	Characterization
x_1^{st}	4	2	58	minimum
x_2^{st}	4	2	66	minimum
x_3^{st}	4	2	45	minimum
x_4^{st}	4	2	63	minimum
x_5^{st}	4	2	0	maximum
x_6^{st}	4	1	0	saddle
x_7^{st}	4	1	0	saddle
x_8^{st}	4	1	0	saddle
x_9^{st}	4	1	0	saddle
$\sigma = 0.01$,	IFC = 107,	IJC = 89,	IFNC = 89	

c) four saddle points

$$
\begin{aligned}
x_6^{\mathrm{st}} &= (0.086677504555, 2.884254701175),\\
x_7^{\mathrm{st}} &= (-3.073025750764, -0.081353044288),\\
x_8^{\mathrm{st}} &= (3.385154183607, 0.073851879838),\\
x_9^{\mathrm{st}} &= (-0.127961346731, -1.953714980244).
\end{aligned}
$$

Executing the implemented program with the initial box $\mathbf{X}^0 = [-5, 5] \times [-5, 5]$, one characterizes all the above stationary points and computes the four minima $x_1^{\mathrm{st}}, \ldots, x_4^{\mathrm{st}}$, all of which have the same function value. For this example the elapsed CPU time on a HP–715 was 0.33 seconds. In Table I we exhibit the corresponding results obtained by IMINBIS.

Example 2. *Kearfott function*, [25]. The objective function f is given by:

$$f(x) = \left(x_1^2 + x_2^2 - 2\right)^2 + \left(x_1^2 - x_2^2 - 1\right)^2.$$

This function has the following nine stationary points:

a) four global minima $x_1^{\mathrm{st}} = (-\sqrt{1.5}, -\sqrt{0.5})$, $x_2^{\mathrm{st}} = (-\sqrt{1.5}, \sqrt{0.5})$, $x_3^{\mathrm{st}} = (\sqrt{1.5}, \sqrt{0.5})$, $x_4^{\mathrm{st}} = (\sqrt{1.5}, -\sqrt{0.5})$,

b) one maximum $x_5^{\mathrm{st}} = (0, 0)$, and

c) four saddle points $x_6^{\mathrm{st}} = (\sqrt{1.5}, 0)$, $x_7^{\mathrm{st}} = (-\sqrt{1.5}, 0)$, $x_8^{\mathrm{st}} = (0, \sqrt{0.5})$, $x_9^{\mathrm{st}} = (0, -\sqrt{0.5})$.

Table II. Kearfott function, $n = 2$.

x_i^{st}	RFE1	RFE2	RFE3	Characterization
x_1^{st}	4	2	58	minimum
x_2^{st}	4	2	57	minimum
x_3^{st}	4	2	51	minimum
x_4^{st}	4	2	52	minimum
x_5^{st}	4	2	0	maximum
x_6^{st}	4	1	0	saddle
x_7^{st}	4	1	0	saddle
x_8^{st}	4	1	0	saddle
x_9^{st}	4	1	0	saddle
$\sigma = 0.01$,	IFC = 38,	IJC = 32,	IFNC = 32	

Running IMINBIS with the initial box $\mathbf{X}^0 = [-2, 2] \times [-2, 2]$, we characterize all the above stationary points and compute the four global minima $x_1^{\mathrm{st}}, \ldots, x_4^{\mathrm{st}}$. In Table II we exhibit the obtained results.

Example 3. *Extended Kearfott function*, [25, 44]. In this case f is given by

$$f(x) = \sum_{i=1}^{n} f_i^2(x),$$

where:

$$f_i(x_1, x_2, \ldots, x_n) = {x_i}^2 - x_{i+1}, \quad i = 1, 2, \ldots, n-1,$$
$$f_n(x_1, x_2, \ldots, x_n) = {x_n}^2 - x_1.$$

This function, for $n = 4$, has the following stationary points:

a) two global minima $x_1^{\mathrm{st}} = (0, 0, 0, 0)$, $x_2^{\mathrm{st}} = (1, 1, 1, 1)$, and

b) one saddle point $x_3^{\mathrm{st}} = (0.5, 0.5, 0.5, 0.5)$,

Executing the implemented program with the initial box $\mathbf{X}^0 = [-2, 2]^4$, one characterizes all the above stationary points and computes the two global minima after 2.32 seconds of CPU time on an HP–715. In Table III we exhibit the corresponding results obtained by IMINBIS.

Example 4. *Quadratic function*, [48]. The objective function f is given by:

$$f(x_1, x_2, \ldots, x_n) = x_1^2 + x_2^2 + \cdots + x_n^2 - r,$$

Table III. Extended Kearfott function, $n = 4$.

x_i^{st}	RFE1	RFE2	RFE3	Characterization
x_1^{st}	16	8	739	minimum
x_2^{st}	16	8	781	minimum
x_3^{st}	16	1	0	saddle
$\sigma = 0.01$,	IFC = 2610,	IJC = 1627,	IFNC = 1627	

with $f(x^*) = -r$ at the global minimum $x^* = (0, \ldots, 0)$. In Table IV we exhibit the results obtained for Example 4 starting with the box $\mathbf{X}^0 = [-99.99, 100]^n$ for various dimensions and variances.

From the above results we observe that our method is independent of the value of the variance σ. This is also true for all of the considered examples.

5. Concluding remarks

A new method for the computation of the global minimum of an n–dimensional real valued function f is presented. Our method computes all the stationary points if required. In the case of the computation of the global minimum it avoids the redundant computation effort needed for various other stationary points by using a new characterization criterion.

The proposed algorithm makes use of interval arithmetic techniques in order to isolate the stationary points of f "roughly". Next, the characterization criterion is used in order to characterize these isolated points as minima, maxima, or saddle points. This criterion can be applied to problems with imprecise values. Finally, the localized minima are computed by using a generalized bisection method and the global minimizers are determined. The only computable information which is required by this bisection method is the algebraic signs of the function values and, thus, it can be applied to imprecise problems.

Certainly it is known that the number of stationary points can be large (in fact it is shown in [36] that it can be exponential in the problem dimension). There exist, however, interesting cases where the number of stationary points is small or moderately large where our method performs very well. Nevertheless, the same restriction holds for any other method which globally optimizes through the discovery of all the stationary points. The advantage of our method compared

Table IV. Quadratic function.

n	σ	RFE1	RFE2	RFE3
3	0.01	8	4	21
	0.05	8	4	21
	0.10	8	4	20
4	0.01	16	8	21
	0.05	16	8	21
	0.10	16	8	20
5	0.01	32	16	21
	0.05	32	16	21
	0.10	32	16	20
6	0.01	64	32	21
	0.05	64	32	21
	0.10	64	32	20
7	0.01	128	64	21
	0.05	128	64	21
	0.10	128	64	20
8	0.01	256	128	21
	0.05	256	128	21
	0.10	256	128	20
9	0.01	512	256	21
	0.05	512	256	21
	0.10	512	256	20
10	0.01	1024	512	21
	0.05	1024	512	21
	0.10	1024	512	20
11	0.01	2048	1024	21
	0.05	2048	1024	21
	0.10	2048	1024	20
12	0.01	4096	2048	21
	0.05	4096	2048	21
	0.01	4096	2048	20
	IFC = 1,	IJC = 1,	IFNC = 1	

to those, is of course the better performance in case of noisy functions and the characterizing process which is clearly more efficient than the traditional methods. More issues on the complexity of these and related problems can be found in [21, 22, 36, 42].

To study the influence of the imprecise information regarding the values of the objective function and gradient, we simulated imprecisions by taking approximations to the true function and gradient values by contaminating them with a small amount of noise. We took this noise

to be proportional to a Gaussian distributed random number with zero mean and various specific variances.

In the case of the computation of global minimizers our algorithm can be more accelerated by parallel techniques [5] and/or using several special tests such as cut–off test, concavity test, local search procedures, as well as special subdivision direction rules [7, 39].

When the total number of the stationary points of f is a priori known, our algorithm can be more efficient. One way to estimate this number is the usage of degree computational techniques [25]. For this purpose one can apply Picard's theorem and compute the value of the topological degree of the extended Picard's function [24]. For the computation of this value Aberth's method [1], which is an adaptation of Kearfott's method [28], can be utilized. Thus, by using the information about the total number of the stationary points of f as an upper bound to the number of sub–boxes to be investigated, the initial box is divided into smaller sub–boxes. In this way, the given region is covered by a set of small boxes where the range of values of $f(x)$ is more precise. An analysis and results for these approaches will appear in a future publication.

Furthermore, preliminary investigations suggest that our techniques can be efficiently applied to the "clustering problem" by solving the corresponding system of nonlinear algebraic and/or transcendental equations, which have several nearly equal, but distinct zeros. Results on this direction will be reported in [53].

References

1. Aberth O., Computation of topological degree using interval arithmetic, and applications, *Math. Comp.*, **62,** 171–178 (1994).
2. Alefeld G. and Herzberger J., *Introduction to Interval Computations*, Translated by J. Rokne, Academic Press, New York, (1983).
3. Androulakis G.S. and Vrahatis M.N., OPTAC: A portable software package for analyzing and comparing optimization methods by visualization, *J. Comput. Appl. Math.*, **75,** (1996), in press.
4. Androulakis G.S., Vrahatis M.N. and Grapsa T.N., Studying the performance of optimization methods by visualization, *Sys. Anal. Mod. Simul.*, in press.
5. Berner S., *A parallel method for verified global optimization*, In: Scientific Computing and Validated Numerics, Alefeld G., Frommer A. and Lang B. eds., Akademie Verlag, Berlin, 200-206 (1996).
6. Box G.E.P. and Muller M.E., A note on the generation of random normal deviates, *Ann. Math. Statistics*, **29,** 610–611 (1958).
7. Csendes T. and Ratz D., Subdivision direction selection in interval methods for global optimization, *SIAM J. Numer. Anal.*, in press.
8. Dixon L.C.W., ACSIM: an accelerated constrained simplex technique, *Computer Aided Design*, **5,** 22–32 (1973).
9. Drossos L., Ragos O., Vrahatis M.N. and Bountis T.C., Method for computing long periodic orbits of dynamical systems, *Phys. Rev. E*, **53,** 1206–1211 (1996).

10. Elster C. and Neumaier A., A grid algorithm for bound constrained optimization of noisy functions, *IMA J. Numer. Anal.*, **15,** 585–608 (1995).
11. Fletcher R., Practical Methods of Optimization, Wiley, New York, (1987).
12. Gill P.E., Murray W. and Wright M.H., Practical Optimization, Academic Press, London, (1981).
13. Glad T. and Goldstein A., Optimization of functions whose values are subject to small errors, *BIT*, **17,** 160–169 (1977).
14. Grapsa T.N. and Vrahatis M.N., The implicit function theorem for solving systems of nonlinear equations in $\mathbb{R}^2$, *Inter. J. Computer Math.*, **28,** 171–181 (1989).
15. Grapsa T.N. and Vrahatis M.N., A dimension–reducing method for solving systems of nonlinear equations in $\mathbb{R}^n$, *Inter. J. Computer Math.*, **32,** 205–216 (1990).
16. Grapsa T.N., Vrahatis M.N. and Bountis T.C., Solving systems of nonlinear equations in $\mathbb{R}^n$ using a rotating hyperplane in $\mathbb{R}^{n+1}$, *Inter. J. Computer Math.*, **35,** 133–151 (1990).
17. Grapsa T.N. and Vrahatis M.N., A new dimension–reducing method for solving systems of nonlinear equations, *Inter. J. Computer Math.*, **55,** 235–244 (1995).
18. Grapsa T.N. and Vrahatis M.N., A dimension–reducing method for unconstrained optimization, *J. Comput. Appl. Math.*, **66,** (1996), in press.
19. Hansen E.R., *Global Optimization using Interval Analysis*, Marcel Dekker, Inc., New York, (1992).
20. Himmelblau D.M., *Applied nonlinear programming*, McGraw–Hill, New York, (1972).
21. Horst R. and Pardalos P.M. (editors), *Handbook of Global Optimization*, Kluwer Academic Publishers, Dordrecht, The Netherlands, (1995).
22. Horst R., Pardalos P.M. and Thoai N.V., *Introduction to Global Optimization*, Kluwer Academic Publishers, Dordrecht, The Netherlands, (1995).
23. Karidis J.P. and Turns S.R., Efficient optimization of computationally expensive objective functions, *IBM Research Report* RC10698 (#47972), Yorktown Heights, New York, (1984).
24. Kavvadias D.J. and Vrahatis M.N., Locating and computing all the simple roots and extrema of a function, *SIAM J. Sci. Comput.*, **17,** No. 5, (1996), in press.
25. Kearfott R.B., An efficient degree–computation method for a generalized method of bisection, *Numer. Math.*, **32,** 109–127 (1979).
26. Kearfott R.B., Some tests of generalized bisection, *ACM Trans. Math. Software*, **13,** 197–220 (1987).
27. Kearfott R.B., Preconditioners for the interval Gauss–Seidel method, *SIAM J. Numer. Anal.*, **27,** 804–822 (1990).
28. Kearfott R.B. and Novoa M., INTBIS, a portable Interval Newton/bisection package, *ACM Trans. Math. Software*, **16,** 152–157 (1990).
29. Kupferschmid M. and Ecker J.G., A note on solution of nonlinear programming problems with imprecise function and gradient values, *Math. Program. Stud.*, **31**, 129–138 (1987).
30. Magoulas G.D., Vrahatis M.N. and Androulakis G.S., Effective backpropagation training with variable stepsize, *Neural Networks*, in press.
31. Magoulas G.D., Vrahatis M.N., Grapsa T.N. and Androulakis G.S., Neural network supervised training based on a dimension reducing method, *Ann. Math. Artif. Intel.*, in press.
32. Magoulas G.D., Vrahatis M.N., Grapsa T.N. and Androulakis G.S., A training method for discrete multilayer neural networks, *Ann. Math. Artif. Intel.*, in press.
33. Nelder J.A. and Mead R., A simplex method for function minimization. *Computer J.*, **7,** 308–313 (1965).

34. Neumaier A., *Interval methods for systems of equations*, Cambridge University Press, Cambridge, (1990).
35. Nocedal J., *Theory of algorithms for unconstrained optimization*, In: Acta Numerica 1991, Iserles A., ed., Cambridge University Press, Cambridge, 199–242, (1991).
36. Pardalos P.M. and Rosen J.B., *Constrained Global Optimization*, Lectures Notes in Computer Science, **268,** Springer–Verlag, New York, (1987).
37. Powell M.J.D., *A review of algorithms for nonlinear equations and unconstrained optimization*, In: ICIAM 1987 Proceedings, McKenna J. and Temam R., eds., SIAM, Philadelphia, 220–232, (1988).
38. Powell M.J.D., A direct search optimization method that models the objective and constraint functions by linear interpolation, *Numerical Analysis Reports*, DAMTP 1992/NA5, Department of Applied Mathematics and Theoretical Physics, University of Cambridge, England, April (1992).
39. Ratz D. and Csendes T., On the selection of subdivision directions in interval branch and bound methods for global optimization, *J. Global Optimization*, **7,** 183–207 (1995).
40. Spendley W., Hext G.R. and Himsworth F.R., Sequential application of simplex designs in optimization and evolutionary operation, *Technometrics*, **4,** 441–461 (1962).
41. Torczon V., On the convergence of the multidimensional search algorithm, *SIAM J. Optimization*, **1,** 123–145 (1991).
42. Traylor B. and Kreinovich V., A bright side of NP–hardness of interval computations: interval heuristics applied to NP–problems, *Reliable Computing*, **1,** 343–359 (1995).
43. Villaverde K. and Kreinovich V., A linear–time algorithm that locates local extrema of a function of one variable from interval measurement results, *Interval Computations*, **No. 4,** pp. 176–194 (1993).
44. Vrahatis M.N., Solving systems of nonlinear equations using the nonzero value of the topological degree, *ACM Trans. Math. Software*, **14,** 312–329 (1988).
45. Vrahatis M.N., CHABIS: A mathematical software package for locating and evaluating roots of systems of non–linear equations, *ACM Trans. Math. Software*, **14,** 330–336 (1988).
46. Vrahatis M.N., An efficient method for locating and computing periodic orbits of nonlinear mappings, *J. Comp. Phys.*, **119,** 105–119 (1995).
47. Vrahatis M.N., *A generalized bisection method for large and imprecise problems*, In: Scientific Computing and Validated Numerics, Alefeld G., Frommer A. and Lang B. eds., Akademie Verlag, Berlin, 186–192, (1996).
48. Vrahatis M.N., Androulakis G.S. and Manoussakis M.E., A new unconstrained optimization method for imprecise function and gradient values, *J. Math. Anal. Appl.*, **197,** 586–607 (1996).
49. Vrahatis M.N. and Bountis T.C., *An efficient method for computing periodic orbits of conservative dynamical systems*, In: Hamiltonian Mechanics: Integrability and Chaotic Behavior, Seimenis J. ed., Plenum Press, New York, 261–274, (1994).
50. Vrahatis M.N., Bountis T.C. and Kollmann M., Periodic orbits and invariant surfaces of 4–D nonlinear mappings, *Inter. J. Bifurc. Chaos*, accepted for publication.
51. Vrahatis M.N. and Iordanidis K.I., A rapid generalized method of bisection for solving systems of nonlinear equations, *Numer. Math.*, **49,** 123–138 (1986).
52. Vrahatis M.N., Servizi G., Turchetti G. and Bountis T.C., A procedure to compute the fixed points and visualize the orbits of a 2D map, *CERN SL/93–06 (AP)*, (1993).
53. Vrahatis M.N. and Sotiropoulos D.G., Solving the root–clustering problem with certainty, (working title), manuscript in preparation.

54. Vrahatis M.N. and Triantafyllou E.C., Locating, characterizing and computing the stationary points of a function, *Reliable Computing*, **2,** 187–194 (1996).
55. Winfield D., Function minimization by interpolation in a data table, *J. Inst. Math. Appl.*, **12,** 339–348 (1973).