Journal of Global Optimization **14:** 157–179, 1999.

Global Interval Methods for Local Nonsmooth Optimization

CHRISTIANE GÖRGES and HELMUT RATSCHEK
Mathematisches Institut der Universität Düsseldorf, Germany

(Received 22 April 1997; accepted in revised form 14 January 1998)

Abstract. An interval method for determining local solutions of nonsmooth unconstrained optimization problems is discussed. The objective function is assumed to be locally Lipschitz and to have appropriate interval inclusions. The method consists of two parts, a local search and a global continuation and termination. The local search consists of a globally convergent descent algorithm showing similarities to ϵ-bundle methods. While ϵ-bundle methods use polytopes as inner approximations of the ϵ-subdifferentials, which are the main tools of almost all bundle concepts, our method uses axes parallel boxes as outer approximations of the ϵ-subdifferentials. The boxes are determined almost automatically with inclusion techniques of interval arithmetic. The dimension of the boxes is equal to the dimension of the problem and remains constant during the whole computation. The application of boxes does not suffer from the necessity to invest methodical and computational efforts to adapt the polytopes to the latest state of the computation as well as to simplify them when the number of vertices becomes too large, as is the case with the polytopes. The second part of the method applies interval techniques of global optimization to the approximative local solution obtained from the search of the first part in order to determine guaranteed error bounds or to improve the solution if necessary. We present prototype algorithms for both parts of the method as well as a complete convergence theory for them and demonstrate how outer approximations can be obtained.

Key words: Global optimization, Interval methods, Local nonsmooth optimization, Scientific computing

AMS subject classification: 90C30, 90C26, 65K05, 65G10

1. Introduction

We want to solve the unconstrained optimization problem

$$\min f(x)$$

where $f : \mathbf{R}^n \to \mathbf{R}$ is locally Lipschitz.

ϵ-bundle methods are a standard means for solving such problems (see for example [14, 23, 24, 26, 27, 28, 33, 34, 48, 49, 54], for mainly the nonconvex case). They were originally developed for convex functions and therefore related to the ϵ-subdifferential concept in the sense of convex analysis. The method was then generalized to locally Lipschitz continuous functions where the ϵ-subdifferential

was defined in the sense of Goldstein [6], that is

$$\partial_\epsilon f(x) = \text{conv}\{\partial f(y) : y \in B(x, \epsilon)\}. \tag{1}$$

($B(x, \epsilon)$ is the closed ball around x with radius ϵ and $\partial f(x)$ the subdifferential of f at x in the sense of Clarke, cf. [4] or Section 6.) ϵ-bundle methods consist in, firstly, a search for a descent direction and, secondly, a step length determination (line search), where trust region techniques are frequently used. The descent direction is chosen from a polytope being an inner approximation of the ϵ-subdifferential of the objective function. The line search underlies the usual conventions of the one-dimensional subproblem such as being exact or inexact, etc. The use of polytopes is, however, connected with two drawbacks: The one is that the polytopes become more and more complex during the numerical computation so that steps have to be incorporated to keep the number of vertices of the polytopes reasonably bounded. The other is that the approximation may become that poor that no appropriate descent direction is found and that it is necessary to update the polytope instead of determining the next descent direction (so-called *nullstep*).

The method we propose consists of two parts, i.e., a *local* und a *global* phase.

The *local* phase is an ϵ-bundle descent method where the ϵ-bundles are approximated by axes parallel parallelepipeds (called *boxes*, for brevity) which are built by interval arithmetic tools. *Global convergence* to an approximative stationary point can be proved.

The *global* phase uses interval arithmetic and branch and bound principles to detect whether the solution obtained by the local part so far, is in fact a sufficiently good approximation of a local minimizer and then to compute safe bounds for it, or otherwise, to improve that solution or eventually, to reject it. *Global convergence* is shown too.

Robinson [46] seemed to be the first who established such a 2-phase method by applying interval arithmetic to find safe error bounds for local minimizers, but for smooth problems only. His idea was to put a box around an approximate Karush–Kuhn–Tucker point which had been obtained by any local method, and then to apply an interval Newton based existence test to this box. Guaranteed error bounds were rendered when the test was successful. The success of the test was mainly depending on the right choice of the box size. Although the numerical results were satisfactory, no convergence proof was available.

One difference of our approach to the one of Robinson is that we use interval arithmetic in the local phase already, that is in order to approximate the ϵ-bundles from without. A second is that convergence for the global phase can be proved. Although the convergence can be very slow under worst case conditions, the efficiency is comparable with Robinson's approach.

We are not aware of any attempt where outer approximations for the ϵ-bundles in descent methods have been used. The notation of an *outer approximation*, however, is frequently used in connection with cutting plane or related techniques (e.g., [31]). Further, a large class of procedures for nonsmooth optimization is covered

by so-called *conceptual algorithms* [32]. Although similarities cannot be excluded, our approach does not fit into this class, since we do not need any assumption for a minimum descent of the steps, as is the case at conceptual algorithms.

Let us glance at the two phases. During the *local* phase, points x_k are generated iteratively in the following manner: Let, at the k-th iteration, X_k be the n-dimensional cube with midpoint x_k and edge length ϵ and let further be

$$\partial f(X_k) = \bigcup_{x \in X_k} \partial f(x). \tag{2}$$

Then the interval approach we propose differs from the standard ϵ-bundle approach in four points:

(i) ϵ-bundle methods use a polytope, say P_k, as an *inner* approximation of the ϵ-subdifferential $\partial_\epsilon f(x_k)$. At the interval approach, a box F_k' is used to approximate (2) from without. Note that the convex hull of (2) can be interpreted as an ϵ-subdifferential w.r.t. the maximum-norm (with a different value for ϵ, certainly).

(ii) In contrast to the polytopes, the boxes can be determined with tools of interval arithmetic. Hence, there is no need to determine any subgradient of the objective function, f, during the computation explicitly. (The subgradients are used to build up the polytopes.) If the interval tools are applied in the right manner, the overestimation by the box approximation is small and does have a negative impact to the method only in rare cases.

(iii) Generally, ϵ-bundle methods require to solve a quadratic subproblem at any step to compute the search direction. At the interval approach, the search direction is simply the argument of

$$\min\{\|y\| : y \in F_k'\} \text{ (shortly written as } \min \|F_k'\|),$$

where the norm denotes the Euclidean norm, cf. Remark 2 in Section 2.

(iv) Whereas some methodical and computational efforts are necessary at the ϵ-bundle approach to obtain a convergence behavior like

$$\min \|P_k\| \to 0 \quad \text{as } k \to \infty,$$

the convergence behavior

$$\min \|F_k'\| \to 0 \quad \text{as } k \to \infty,$$

which arises at the interval approach, is always satisfied without any further ado.

The termination of nonsmooth algorithms is generally initiated when

$$0 \in \partial f(x_k) \quad \text{or} \quad 0 \in \partial_\epsilon f(x_k)$$

or similar conditions hold. Therefore, ϵ-bundle algorithms are frequently caused to stop when

$$\min \|P_k\| \leqslant \delta \quad \text{for some } \delta > 0. \tag{3}$$

Now, if outer approximations are used, it is equally reasonable to stop the algorithm when

$$\min \|F_k'\| \leqslant \delta \tag{4}$$

occurs.

Note that neither (3) nor (4) can guarantee that $0 \in \partial f(x_k)$ or $0 \in \partial_\epsilon f(x_k)$ is satisfied, and that δ is mainly a measure for the flatness of the subgradients under consideration. Following the standards of scientific computing, it is therefore necessary either to confirm that $0 \in \partial_\epsilon f(x_k)$ and that a local minimizer lies in X_k or to continue the search by checking for a local minimizer in boxes adjacent to X_k. These tasks together make up the contents of the *global* phase. Hence, in the first case, being the affirmative one, safe bounds for a local solution are obtained automatically. Local solution means, in fact, a local minimizer of f, and not only a stationary or saddle point. In the second case, the search for a solution will be continued, but no longer with ϵ-bundle related methods, but with *i*nterval *b*ranch and *b*ound and *m*onotonicity checking methods (we call them IBBM for brevity). The reason for dropping the ϵ-bundles is that the inaccuracy of the local search result was caused mainly by the use of approximations of subdifferentials. Therefore, this global phase will primarily rely on the function values and admit derivative information only secondarily in the so-called *monotonicity test*, which is a suitable means for nonsmooth functions too [39]. While interval based global optimization methods are usually applied to an initial box in which a solution is looked for, we have to modify these methods in such a manner that the search will proceed to adjacent boxes when no minimizer is found in the *interior* of the initial box (being the termination box of the local phase, mainly).

Since the basic features of IBBM are well known, cf. for example [3, 8, 18–21, 39, 41, 42, 44, 45], there is no need at all for their extensive treatment, and we restrict ourselves to exploring the links between the foregoing local phase and IBBM and abridging the main steps of IBBM. However, we have to treat those issues more detailed which are needed for the convergence proof of the global part. First attempts for solving nonsmooth problems with interval methods seem to be [21, 39, 53].

In Section 2, the algorithm for the local phase is established and in Section 3, the algorithm for the global continuation. The convergence proofs for the two algorithms are given in Section 4. In Sections 5 and 6, we discuss some techniques for constructing inclusions for functions and subdifferentials.

2. The local phase (how to get a local minimizer)

The aim of this section is a search for an approximative local minimizer of the unconstrained optimization problem

$$\min f(x)$$

where f is locally Lipschitz. To achieve this aim, we propose the following algorithm, which will be terminated by condition (4). The computation will then be continued by a global phase, which will either confirm that a local minimizer has been obtained and provide safe bounds for it or improve the solution (cf. the next section).

Let $\mathbf{I}$ be the set of real compact intervals, then $\mathbf{I}^n$ is the set of n-dimensional intervals resp. boxes. Let $E \in \mathbf{I}^n$ be that cube which has the zero vector as midpoint and ϵ as edge length.

ALGORITHM 2.1. (for obtaining an approximative local solution)
INPUT PARAMETERS:

$x_0 \in \mathbf{R}^n$ as starting vector,
$\epsilon > 0$ as width of the cubes needed as domain for the subdifferentials (default: 10^{-2} for average working areas for the iterates, x_k, that is, about $0 \leqslant \|x_k\| \leqslant 100$) and
$k = 0$ as counting index for the number of iterations.

Step 1 (Creating the boxes).
Set $X_k := x_k + E \in \mathbf{I}^n$;
determine $F_k' \in \mathbf{I}^n$ with $F_k' \supseteq \partial f(X_k)$ (as outer box inclusion of the set of subdifferentials over X_k).

Step 2 (Termination criterion).
If $\min \|F_k'\| \leqslant \delta$, then terminate and continue with Algorithm 3.1 for obtaining inclusions of the minimizer.

Step 3 (Choice of the descent direction).
Let $g_k \in F_k'$ minimize $\|F_k'\|$, that is, $\|g_k\| = \min \|F_k'\|$.
Set $d_k := -g_k/\|g_k\|$ (descent direction)

Step 4 (Step length determination).
Let t_k solve $\min f(x_k + t\, d_k)$ subject to $t > 0$.

Step 5 (Next iterate).
Set $x_{k+1} := x_k + t_k\, d_k$,
$k := k + 1$,
Go to Step 1.

REMARKS

1. As will be seen in Section 4, the termination criterion will hold under standard assumptions after a finite number of steps.

2. Since F_k' is an axes parallel box, the checking of the termination condition in Step 2 as well as the determination of g_k in Step 3 is a simple programming task.
3. The algorithm works equally well with inexact line search (compare [25]) under the usual conditions of type Armijo et al. (The proof of Theorem 4.1, which deals with the convergence of the algorithm with exact line search, remains valid for the inexact search version under some weak minimum step length assumptions.)
4. The algorithm works also for that kind of generalized locally Lipschitz functions where unbounded subdifferentials are admitted (cf. Section 5).
5. The main principle of the algorithm, that is the use of outer approximations of subdifferentials over a cube (or a box or a ball, etc.) is applicable to other methods for solving nonsmooth optimization problems too.
6. Steps 3 and 4 can also be carried out within a trust region environment. We do not discuss this variant since it has no influence to the use of outer approximations, which is the target of this paper.

3. The global phase (how to get safe error bounds)

Algorithm 2.1 terminates when an approximate of a local minimizer is found in the sense that the necessary condition for it is almost satisfied. That is, if X_k is the box where Algorithm 2.1 has been terminating, the chances that a local minimizer does already lie in X_k are excellent. But it is also possible that the solution is a bit outside this box, and, in a worst case situation, it cannot be excluded, that the box X_k contains an extremely flat function piece instead of a minimizer or that the zero with respect to condition (4) is caught only because the overestimation of the bundle by the box is too large. Therefore, in the sense of scientific or validated computation, a decision which of these cases applies is unavoidable.

The global phase, as described in this section, will either confirm that a solution lies in X_k or will initiate a search for a solution with global means as far as it can be justified economically. The heart of the global search could be any deterministic global optimization procedure, see for example [1, 3, 5, 8–11, 15–20, 38, 39, 41, 42, 44, 45, 52]. Since we do not know any method that is more robust, simpler and applicable in more general situations than interval based methods, we pick out the one we were calling IBBM in Section 1. This name is rather a working title than the official name of this method. We refer to [8, 41, 42] for a more detailed description and to [21, 39, 53] for an adaption to nonsmooth functions. We do not describe this method again in this paper, but it is necessary to explain precisely what we expect from this method and its purpose in order to formulate the algorithm in no uncertain terms:

Let $Y \in \mathbf{I}^n$ be a box (such as X_k of Algorithm 2.1) and f be the locally Lipschitz function as objective function of the minimization problem. Further we assume that to every subbox Z of Y, outer approximations $F \in \mathbf{I}$ and $F' \in \mathbf{I}^n$ of f and ∂f over

Z, resp., are available, such that

$$F = F(Z) \supseteq f(Z), \qquad F' = F'(Z) \supseteq \partial f(Z).$$

Under the *application of IBBM to f over Y*, we understand a computation which results in a list of subboxes $S_1, \ldots, S_s$ of Y with prescribed maximum width such that the global minimizers of f over Y lie in their union. The subboxes can be degenerated. Details can be found in the references cited.

It would also go too far to show where the monotonicity test flows in during the application. We only note that this test is one of the most effective tools of interval analysis. The test is designed to discover whether a function is strictly monotone over a box with respect to a coordinate direction. The conclusion in the affirmative case is that this function has no minimizers in the interior of this box. (An example can be found in Section 6, Example 6.4.)

Let us now connect IBBM with the target to find inclusions for the approximate minimizer which has been obtained by Algorithm 2.1. Assume that the procedure had stopped with a box X_k. We set $Y = X_k$. We remind that Y is a cube, and that its edge length is ϵ. Now, IBBM is applied to f over Y until the remaining subboxes, $S_1, \ldots, S_s$, are of maximum edge length $\epsilon/4$. Let Z be their box hull, that is, the smallest box that contains each of the subboxes $S_1, \ldots, S_s$. Clearly, Z is also a subbox of Y. By int Y the interior of Y (relatively to $\mathbf{R}^n$) is denoted. We denote the *midpoint* of a box Z by mid(Z) and the *width* of Z (maximum edge length of Z) by $w(Z)$. Then we distinguish three cases:

Case (i). $Z \subseteq$ int Y. Due to the properties of IBBM, Z contains a global minimizer of f over Y. Since Z lies in the interior of Y, a local minimizer, say x^*, of f over $\mathbf{R}^n$ lies in Z. Hence, the aim of the global phase of the computation is reached, and one can *terminate* the computation. (Since x^* will not be known in general, one can accept the midpoint of Z as approximate of x^*.) We have thus obtained the absolute error estimation

$$\|x^* - \text{mid}(Z)\|_\infty \leqslant w(Z)/2 < \epsilon/2 \tag{5}$$

where the norm used is the maximum norm. (If the absolute error width obtained is not satisfiable, IBBM can be applied to Z again with a prescribed smaller size of the final subboxes.)

Case (ii). $w(Z) \leqslant \epsilon/4$ and the edges of Y and Z intersect. This condition does not yet allow a decision whether int Y contains a global minimizer of $f \mid Y$ or not. Two constellations are possible:

(a) There is a global minimizer of $f \mid Y$, say x°, in the interior of Z so that x° is a local minimizer of f in any case.

(b) There is a global minimizer, x°, of $f \mid Y$ on the edge of Y. Certainly, x° could be a local minimizer of f too, but it is most likely that it is located

outside of Y in one of the boxes adjacent to Y. This second constellation is frequently indicated by the computational result that Z consists of a point or a degenerated box of lower dimension lying on the edge of Y. The degeneration had been caused by the successful application of the monotonicity test and the succeeding replacement of the whole box Y or of subboxes of Y by edges which might contain a global minimizer of $f \mid Y$.

In order to strive for a computational decision of which of the two constellations, (a) or (b) might be applicable, it makes sense to continue the search for minimizers around x°. Therefore it is reasonable to apply IBBM again, actually to that box that has $\mathrm{mid}(Z)$ as midpoint and edge length ϵ. (We will show in the next section that this case can occur only a finite number of times.) Nevertheless, the chances are not too bad that, already after a few steps of the type of case (ii), the computation turns back to case (i) and terminates with the error estimate (5).

Case (iii). None of the cases (i) or (ii) holds. In this case we suggest to terminate the computation since a continuation will, in general, be too expensive for solving local problems. The high costs of a continuation arise since it is necessary to switch to completely global methods outside the current box, Y. That is, one has to extend the search for minimizers probably to more than one adjacent boxes, and this extension will repeatedly be necessary, further, one has to store all these boxes in lists and to apply the global methods to the lists. (Such a situation can arise if the objective function is extremely flat, or if bulks of minimizers lie in the area under consideration, or if there are singular points from which several directions of locally steepest descent leave or if the influence of rounding errors increases too much.) In spite of the high numerical costs, it is possible to show that such a completely global procedure terminates after a finite number of steps, if standard assumptions are satisfied (see e.g., [41]). Nevertheless, if one really wants to obtain guaranteed inclusions for solutions, it is worthwhile to go through this case.

ALGORITHM 3.1. (for obtaining safe error bounds for the solution)

INPUT:
Box $Y := X_k \in \mathbf{I}^n$ which had been the current box at the termination of Algorithm 2.1, where $\epsilon = w(Y)$.

Step 1.
Apply IBBM to f over Y as explained above. The result is a list of (eventually degenerated) boxes $S_1, \ldots, S_s$ of maximum width $\epsilon/4$. Let Z be their box hull.

Step 2.
If $Z \subseteq \mathrm{int}\, Y$ then STOP (a local minimizer x^* of f lies in Z). *(Optionally:* If the error estimate (5) is not sharp enough, apply IBBM to $S_1, \ldots, S_s$ again with diminished values for ϵ.)

Step 3.
If $w(Z) \leqslant \epsilon/4$ and if the edges of Y and Z have a nonempty intersection, then set $Y := \mathrm{mid}(Z) + E$ and goto Step 1
else goto Step 4.

Step 4.
STOP (since the continuation of the computation tends to become uneconomical). *(Optionally:* continue the computation by enlarging the box Y and applying IBBM repeatedly, if a guaranteed inclusion of solutions is required.)

REMARK 3.1. If Step 3 is called up several times during the computation, it is reasonable to reexamine whether criterion $\min \|F'(Y)\| \leqslant \delta$ is still valid, where $F'(Y) \in \mathbf{I}^n$ is the actual outer approximation of $\partial f(Y)$, cf. the termination criterion in Step 2 of Algorithm 2.1. If this criterion is hurt, the necessary condition for Y to contain a local minimizer of f is no longer valid so that the computation of the global continuation becomes uneconomical and it would be more reasonable to return to Algorithm 2.1 with input box $X_0 := Y$ and to restart the local search. (Such a situation may occur if the iterates X_k of Algorithm 2.1 pass through points that have a saddle point behavior. First, condition $\min \|F'_k\| \leqslant \delta$ will apply in such cases and the computation will switch to the global method. It is plausible that then Step 3 of this algorithm would be called up several times, and since the descent becomes steeper and steeper, there is no chance for a soon leaving of Step 3. The costs will go up since the maximum distance from one iterate to the next is not larger than $\epsilon/2$. Clearly, Algorithm 2.1 would proceed reasonably in this situation.)

4. Convergence results

We show that Algorithm 2.1 as well as Algorithm 3.1 terminate after a finite number of iterations. The proofs need the standard assumption that the computation can be executed within a compact domain, i.e., that,

$$\text{for any } y \in \mathbf{R}^n, \text{ the set } \{x \in \mathbf{R}^n : f(x) \leqslant f(y)\} \text{ is bounded.} \tag{6}$$

The following lemma enlightens the descent situation at the k-th iteration of Algorithm 2.1.

LEMMA 4.1. *If* $0 \notin F'_k$, *then*
(i) d_k *is a (proper) descent direction of* f *for each* $x \in X_k$,
(ii) $x_{k+1} \notin X_k$.
Proof. A well-known theorem of convex analysis says, that for a compact convex set $G \subseteq \mathbf{R}^n$ and for any $g \in G$ the following equivalence holds,

$$g = \arg\min\{\|p\| : p \in G\} \iff p^t g \geqslant \|g\|^2 \quad \text{for all } p \in G$$

(cf. [30] or [13] for example).

We apply this theorem to $g_k = \arg\min\{\|p\| : p \in F'_k\}$, (cf. Step 3 of Algorithm 2.1, where $d_k = -g_k/\|g_k\|$ is set), and obtain

$$p^t g_k \geqslant \|g_k\|^2 \quad \text{for all } p \in F'_k,$$

and further, since $g_k \neq 0$ and $\partial f(X_k) \subseteq F_k'$,

$$p^t d_k \leqslant -\|g_k\| < 0 \quad \text{for all } p \in \partial f(X_k).$$

Consider any $x \in X_k$. Then it follows that

$$p^t d_k \leqslant -\|g_k\| < 0 \quad \text{for all } p \in \partial f(x), \tag{7}$$

which means that d_k is a proper descent direction of f at x (cf. [30], p. 78 for example). This proves (i).

The assertion (ii) is obvious: If the next iterate, $x_{k+1} = x_k + t_k d_k$ (cf. Algorithm 2.1), would lie in X_{k+1}, then d_k would be a proper descent direction of f in x_{k+1} by (i), and t_k could not be a solution of the step length determination. □

Let now (x_k) be a not terminating sequence of iterates of Algorithm 2.1 and (g_k) be the related sequence of directions occurring in Step 3, that is, $g_k = \arg\min \|F_k'\|$.

THEOREM 4.1. *If the assumption (6) holds, then the sequence* (g_k) *converges to* 0.

Proof. Because of (6) there exists an accumulation point x^* of (x_k). Let $(x_k)_{k\in\mathcal{K}}$ be a subsequence with

$$\lim_{k\in\mathcal{K}} x_k = x^*.$$

Applying Lemma 4.1 we receive a step-length greater than $\epsilon/2$ in every step. From Lebourg's mean-value theorem (cf. [30], pp. 40, 41 for example) we get for all k

$$f\left(x_k + \frac{\epsilon}{2} d_k\right) - f(x_k) = \frac{\epsilon}{2} h^t d_k$$

where $h \in \partial f(u)$ with u on the line segment between x_k and $x_k + \frac{\epsilon}{2} d_k$. As $\|d_k\| = 1$, it is

$$u \in X_k \quad \text{and therefore} \quad \partial f(u) \subseteq F_k'.$$

By (7), we get the estimate

$$f(x_k + t_k d_k) - f(x_k) \leqslant f\left(x_k + \frac{\epsilon}{2} d_k\right) - f(x_k) \leqslant -\frac{\epsilon}{2}\|g_k\| \tag{8}$$

for the descent made in every step. This also shows the monotone decrease of $(f(x_k))$ which together with the continuity of the function f means that

$$\lim_{k\to\infty} f(x_k) = f(x^*)$$

for the whole sequence $(f(x_k))$. Let us now have a look at the sequence (g_k).

Assume that it does not converge to the zero-vector which gives the existence of a $\mu > 0$ such that

$$\forall k_0\, \exists k \geqslant k_0 : \|g_k\| \geqslant \mu$$

and so there exists a subsequence $(g_k)_{k\in\mathcal{K}_1}$ with

$$\|g_k\| \geqslant \mu \quad \text{for all } k \in \mathcal{K}_1 \tag{9}$$

The combination of our observations leads to

$$\begin{aligned} f(x^*) - f(x_0) &= \sum_{k=0}^{\infty} (f(x_{k+1}) - f(x_k)) \\ &\leqslant \sum_{k\in\mathcal{K}_1} (f(x_{k+1}) - f(x_k)) \\ &\leqslant -\frac{\epsilon}{2} \sum_{k\in\mathcal{K}_1} \|g_k\| \\ &\leqslant -\frac{\epsilon}{2} \sum_{k\in\mathcal{K}_1} \mu \end{aligned}$$

which is a contradiction, since the right side goes to $-\infty$. So the assertion of the convergence theorem must be true. □

As a direct consequence of Theorem 4.1 we can formulate

COROLLARY 4.1. *Algorithm 2.1 terminates after a finite number of iterations, if (6) holds.*

REMARK 4.1. As already mentioned this convergence result remains valid, when inexact line-search combined with a minimal step-length (cf. [25] for example) is used. Since a descent direction is determined in each step and $t_k \geqslant \epsilon/2$ in case of exact line-search we can obviously demand a constant minimal step-length $t_{\min} < \epsilon/2$ for the computation which then also replaces $\epsilon/2$ in our convergence proof.

THEOREM 4.2. *Algorithm 3.1 terminates after a finite number of iterations, if (6) holds.*

Proof. Let Y_0 be the initial box of the global continuation, let $Z_0 \subseteq Y_0$ be the box hull determined in Step 1, and assume that the if-clause of Step 3 holds for Z_0 and Y_0. Set $x_0 = \mathrm{mid}(Y_0)$ and $x_1 = \mathrm{mid}(Z_0)$. Then the distance between x_0 and x_1 can be estimated by

$$\|x_0 - x_1\|_\infty \geqslant 3\,\epsilon/8. \tag{10}$$

This is due to the facts that Z_0 touches the edge of Y_0 and that $w(Z_0) \leqslant \epsilon/4$. Let ξ_0 be a global minimizer of $f \mid Y_0$ then it follows that $\xi_0 \in Z_0$ and

$$f(\xi_0) < f(y) \quad \text{for all } y \in Y_0 \setminus Z_0. \tag{11}$$

(Otherwise, if there would exist an element $y \in Y_0 \setminus Z_0$ with $f(\xi_0) \geqslant f(y)$, then y would already lie in Z_0 due to the construction of an IBBM application.) According to the instruction of Step 3, a box Y_1 is provided for the next iteration,

$$Y_1 := x_1 + E.$$

The application of Step 1 to Y_1 yields a box hull, $Z_1 \subseteq Y_1$, and we assume again that the if-clause of Step 3 holds. If $x_2 = \mathrm{mid}(Z_1)$, then $w(Z_1) \leqslant \epsilon/4$ and $\|x_1 - x_2\|_\infty \geqslant 3\epsilon/8$. If $\xi_1 \in Z_1$ is a global minimizer of $f|Y_1$, we again obtain the relationship $f(\xi_1) < f(y)$ for all $y \in Y_1 \setminus Z_1$. Since Z_1 touches the edge of Y_1 and since $\|\xi_0 - x_1\|_\infty \leqslant \epsilon/4$, it is obvious that ξ_0 cannot lie in Z_1, so that $\xi_0 \in Y_1 \setminus Z_1$, and it follows

$$f(\xi_1) < f(\xi_0)$$

by (11).

We continue these iterations, and we get, for each k, that the inequality $\|x_k - x_{k+1}\|_\infty \geqslant 3\epsilon/8$ holds, where $x_k = \mathrm{mid}(Y_k)$ and $x_{k+1} = \mathrm{mid}(Z_k)$, and that for the global minimizers $\xi_k \in Z_k$ of $f \mid Y_k$, the strict inequality chain

$$f(\xi_k) < f(\xi_{k-1}) < \cdots < f(\xi_0) \tag{12}$$

is valid. In order to get a contradiction, we assume that this process would never terminate, i.e., that the if-clause of Step 3 would be valid at every iteration.

By (6), the area where the minimizers operate is compact. Since the distances from the box midpoints to the related minimizers is bounded too, the iterates x_k as well operate in a compact area. Hence, the sequence (x_k) has accumulation points, and we can figure out points of this sequence with arbitrarily small distance. Without restriction of the generality, we assume that x_0 and x_k are such points which satisfy

$$\|x_0 - x_k\|_\infty < \epsilon/8.$$

Note that $k = 1$ is excluded due to (10). Because of

$$3\epsilon/8 \leqslant \|x_0 - x_1\|_\infty \leqslant \|x_0 - x_k\|_\infty + \|x_1 - x_k\|_\infty$$

we obtain

$$\|x_1 - x_k\|_\infty > \epsilon/4.$$

Since x_1 and x_k are the midpoints of Z_0 and Z_{k-1}, respectively, and since the widths of Z_0 and Z_{k-1} are at most $\epsilon/4$, we obtain that

$$Z_0 \cap Z_{k-1} = \emptyset.$$

On the other hand, since $\|x_0 - x_k\|_\infty < \epsilon/8$ and $x_k = \mathrm{mid}(Z_{k-1})$, we get

$$Z_{k-1} \subseteq Y_0.$$

Hence $Z_{k-1} \subseteq Y_0 \setminus Z_0$ so that

$$f(\xi_0) < f(y) \quad \text{for all } y \in Z_{k-1}$$

due to (11). If we set $y = \xi_{k-1} \in Z_{k-1}$, we get

$$f(\xi_0) < f(\xi_{k-1}),$$

which contradicts (12). □

REMARK 4.2. Algorithm 3.1 has been slimmed a bit in order to make the proof of Theorem 4.2 as simple as possible. It is obvious that it can be designed more effective. For example, larger boxes could be admitted for Z_k. It is not even necessary to set $\mathrm{mid}(Z_k) = \mathrm{mid}(Y_{k+1})$.

5. About the interval tools needed

In this and the next section, we give some hints and examples how to find appropriate inclusions of the objective functions and the subdifferentials. The reader is referred to [40] or [51] for a more extensive treatment. See also the example in [21]. Further, we restrict ourselves to programmable functions to avoid considerations, which are rather sophisticated than applicable. Programmable functions are nothing more than factorable functions in the sense of McCormick [29], that obey the additional condition that a computer program for determining the function values can explicitly be written in some common programming language. Thus, we call a function *f programmable* if f can be built up from the arithmetic, the logical and the comparison operators and some collection of standard transcendental functions (like sin, cos, power, squareroot, exp, log, etc.). Specially, given an argument x, the function value, $f(x)$, can be computed with a finite number of operations. All the functions dealt with in this section are assumed to be programmable.

Let $g : \mathbf{R}^n \to \mathbf{R}^m$ be a single-valued or multivalued function. Then we call an interval-valued (case $m = 1$) or box-valued function (case $m > 1$),

$$G : \mathbf{I}^n \to \mathbf{I}^m$$

an *inclusion function* of g, if

$$g(Y) \subseteq G(Y) \quad \text{for any } Y \in \mathbf{I}^n.$$

We need inclusion functions of the objective function $f : \mathbf{R}^n \to \mathbf{R}$ (single-valued) and of the subdifferential $\partial f : \mathbf{R}^n \to \mathbf{R}^n$ (multivalued). A real number $\alpha > 0$ is called *order* of the inclusion function G for g if

$$w(G(Y)) - w(g(Y)) = O(w(Y)^\alpha) \quad \text{for } Y \in \mathbf{I}^n.$$

In case of $m > 1$, the *width* of $g(Y)$ is understood as the width of the *box hull* of $g(Y)$, that is the smallest box which includes $g(Y)$.

The order is a mainly asymptotic measure of the overestimation of $g(Y)$ by $G(Y)$. The consideration of the order is not extremely important for our algorithm, since the box sizes do not vary too much during the computation.

A straight way to obtain inclusion functions is the use of so-called natural interval extensions, which were introduced by Moore [35] and count to the most important tools of interval analysis. They are implemented in almost every software package of interval arithmetic like PASCAL-XSC, C-XSC, FORTRAN-XSC, etc. The features of natural interval extensions can be sketched in the following way:

Let $g : \mathbf{R}^n \to \mathbf{R}^m$ be a programmable function and $g(x)$ be a function expression (for example, a program for computing $g(x)$ in dependence of x). Then the *natural interval extension* of $g(x)$ to Y, where Y is a box or a box variable over $\mathbf{I}^n$, is that expression that occurs from the expression $g(x)$, if the variable x is replaced by Y and if the result of this substitution is interpreted (and executed) as an interval arithmetic expression. If cases occur, one has to take care that, after the replacement, the cases do not hurt the inclusion property and the program remains complete (cf. Example 5.1). If standard transcendental functions occur in the expression, we may think of the range of this function over the interval as natural interval extension. This is not a presumptuous concept, since the monotonicity behavior of such functions is well-known and the range can be determined by cutting up the underlying interval into the monotonicity areas of the function.

For example, the natural interval extension of the function expression

$$f(x) = x_1 + \sin x_2$$

is

$$F(Y) = Y_1 + \sin Y_2,$$

where $\sin Y_2$ is the range of sin over Y_2. The order of this inclusion is 1. It is possible to raise the order by admitting more involved expressions for $f(x)$ and to approximate f by mean value, Taylor, interpolation formulas, etc. (see, for example, [40]) or to apply other principles. Such improvements of the order are made possible by a curiosity of interval arithmetic, which is, that different expressions for one and the same function can lead to different inclusion functions.

In this paper, we only provide a few principles for a more or less fast and convenient application of interval arithmetic, and it is best to consider a few examples. If $A, B \in \mathbf{I}^m$, then $A \vee B$ shall denote the interval or box hull of A and B, that is, the smallest interval or box in $\mathbf{I}^m$ containing A and B.

EXAMPLE 5.1. Let

$$\begin{aligned} f(x) &= x \sin x \quad \text{if } x \geqslant 0, \\ &= x \cos x \quad \text{if } x \leqslant 0. \end{aligned}$$

The straight inclusion function of f is

$$\begin{aligned} F(Y) &= Y \sin Y && \text{if } Y \geqslant 0, \\ &= Y \cos Y && \text{if } Y \leqslant 0, \\ &= (Y_1 \sin Y_1) \vee (Y_2 \cos Y_2) && \text{if } 0 \in \operatorname{int} Y \end{aligned}$$

where $Y = Y_1 \cup Y_2,\ Y_1 \geqslant 0,\ Y_2 \leqslant 0$. One can observe, that a direct replacement of x by Y in the expression for $f(x)$ will not lead to a complete inclusion function. Particularly one has to be aware that the two cases $Y \geqslant 0$ and $Y \leqslant 0$ do not cover all possibilities (as is the case for real numbers), and that one has to add a third case, $0 \in \operatorname{int} Y$, to make the decision complete. This is done best by choosing the union of the inclusions of the first two single cases, and, in order not to leave the interval environment, to accept the interval hull of the union as inclusion. In the added case, $0 \in \operatorname{int} Y$, we had split the interval Y in its positive and in its negative part to get a smaller inclusion. In situations, where Y is small or where a splitting is inopportune or impossible, the splitting of Y can simply be dropped and the whole argument Y be taken as argument (as far as possible) for both branches of the function. At the example, this leads to the inclusion

$$Y(\sin Y \vee \cos Y) \quad \text{if} \quad 0 \in \operatorname{int} Y.$$

An example, where such a splitting is not possible, is the following one:

EXAMPLE 5.2. Let

$$\begin{aligned} f(x, y) &= x - y && \text{if } x + y \geqslant 1, \\ &= (x - y)^2 && \text{otherwise.} \end{aligned}$$

A suitable inclusion function is

$$\begin{aligned} F(X, Y) &= X - Y, && \text{if } X + Y \geqslant 1, \\ &= (X - Y)^2, && \text{if } X + Y < 1, \\ &= (X - Y) \vee (X - Y)^2 && \text{otherwise.} \end{aligned}$$

EXAMPLE 5.3. Let

$$f(x) = \max\{f_i(x) : i = 1, \ldots, k\}$$

with programmable functions $f_i : \mathbf{R}^n \to \mathbf{R}$, $i = 1, \ldots, k$. Let further $F_i : \mathbf{I}^n \to \mathbf{I}$ be an inclusion function for f_i, $i = 1, \ldots, k$. An inclusion function $F : \mathbf{I}^n \to \mathbf{I}$ of f can be constructed as follows: Let $Y \in \mathbf{I}^n$ be given.

(i) Choose an inclusion $F_s(Y)$, $s \in \{1, \ldots, k\}$, with maximum upper bound, that is,

$$\operatorname{ub} F_s(Y) \geqslant \operatorname{ub} F_i(Y), \quad i = 1, \ldots, k,$$

where ub stands for upper bound.

(ii) Let $F(Y)$ be the union of those among the intervals $F_i(Y)$ which satisfy

$$F_s(Y) \cap \operatorname{int} F_i(Y) \neq \emptyset, \quad i = 1, \ldots, k.$$

Then $F(Y) \supseteq f(Y)$ is evident.

EXAMPLE 5.4. The function $f : [0, 1] \to \mathbf{R}$ defined as

$$\begin{aligned} f(x) &= x \log x \quad \text{if } x > 0, \\ &= 0 \qquad\quad \text{if } x = 0. \end{aligned}$$

is locally Lipschitz at the point 0 in a generalized sense as the derivative has the improper value $-\infty$. Therefore, 0 is the bottleneck at an attempt to find inclusions $F(Y)$ if $0 \in Y$. The direct way for finding an inclusion would yield $F(Y) = Y \log Y$ if $0 \in Y$. If $Y = [0, y]$, then $F(Y) = [0, y] \log [0, y] = [0, y)(-\infty, \log y) = (-\infty, 0]$ provided, infinite interval arithmetic is applied. This interval, however, cannot be accepted as a suitable result, since the overestimation is not only too large, but also completely unrealistic. Hence we have to condemn this direct way. Another idea is based on the observation that the monotonicity areas of f can be determined easily, so that one could provide the exact range of f over any interval $Y \subseteq [0, 1]$. This would be, however, only an exceptional situation, since we assume the possibility of a direct range determination at standard transitive functions only. Thus we don't intend to present such a concept as a typical procedure either. A practicable way would be to investigate the monotonicity of f in some small neighborhood of the branching point, 0. This could be done by hand at the preparation phase of the problem or computationally as part of the program. In this latter case the monotonicity test can be executed automatically with interval tools. Only the knowledge of inclusions of the derivative or the subdifferential is necessary (cf. Remark 6.2 later on), which does not require additional computations, since the computation of these inclusions is a definite part of the algorithm. The monotonicity test would indicate that f is strictly monotonically decreasing in a sufficient small neighborhood of 0. (Note that an infinite interval arithmetic must be used for approximating $\partial f(Y)$ which is due to the generalized local Lipschitz property of f at 0, cf. Example 6.4.) Therefore, the following inclusion would do it (dropping parts of the knowledge our brain has):

$$\begin{aligned} F(Y) &= Y \log Y \text{ if } Y > 0, \\ &= [y \log y,\ 0] \text{ if } Y = [0, y] \leqslant 1/e, \\ &= [y_1 \log y_1,\ 0] \vee (Y_2 \log Y_2) \text{ if } Y = [0, y_1] \cup Y_2,\ Y_2 = [y_1,\ y_2], \end{aligned}$$

(where $Y \leqslant 1/e$ cannot be guaranteed, but Y has been split so that at least $y_1 \leqslant 1/e$ is ensured).

Example 5.2 and the alternative of the inclusion in Example 5.1 as well as Example 5.4 show the general recipe how to get the inclusions for involved situations

where a splitting of the argument box is hardly possible. That is, if the function is built up from several cases, say k cases, let

$$f(x) = f_i(x) \text{ in case } i, \quad \text{for } i = 1, \ldots, k.$$

We further assume that the functions f_i $(i = 1, \ldots, k)$ are built up from arithmetic operations and standard transitive functions only. (This is no real restriction. It means only, that, if a branch itself consists of subcases, the branch itself is already split.) Let further $F_i(Y)$ be an inclusion of $f_i(Y)$. If f_i is not completely defined on Y, take a sufficiently large subdomain instead of Y. Then let $F(Y)$ be the box hull of the union $\cup_i F_i(Y)$ where i runs through all cases which cannot be excluded to hold for some $x \in Y$.

6. Outer approximations of the subdifferentials

In this section we show how to construct outer approximations for the subdifferentials. In order to avoid the discussion of a very sophisticated theory which is still under construction we restrict ourselves to those subdifferentials which arise from the examples of Section 5. It should then be not too difficult to transfer the basic principles to other examples and problems.

Inclusions for the subdifferentials can be gained in a similar manner as for nonsmooth objective functions. For the realization of the idea, Clarke's subdifferential version [4] is suitable,

$$\partial f(x) = \text{ conv } \{\lim \nabla f(x_\nu) : x_\nu \to x,\ x_\nu \notin \Omega \cup T\}$$

where Ω is the set of all points where f is not differentiable, and T is any set of Lebesgue measure zero. Ω is also of Lebesgue measure zero due to a theorem of Rademacher (cf. [4]). The key to a practicable procedure is that the box hull of $\{\nabla f(x) : x \in Y \setminus \Omega\}$ is an inclusion of $\partial f(Y)$. This is due to the fact that a box hull is a closed set and contains all the existing limits occurring in the subdifferential. Since f and hence ∇f are programmable, the set $Y \setminus \Omega$ will be controllable, in general. If no further information is available, we put together the subdifferential inclusions over Y of all possible cases as was done before at the determination of inclusions for f. This means in the worst case, that a finite number of cases occur,

$$\nabla f(x) = \nabla f_i(x) \text{ in case } i, \text{ for } i = 1, \ldots, k.$$

We then simply set $F'(Y)$ as the box hull

$$\nabla f_1(Y \setminus \Omega_1) \vee \ldots \vee \nabla f_k(Y \setminus \Omega_k),$$

where Ω_i is the set of points for that f_i is not differentiable $(i = 1, \ldots, k)$. Since ∇f_i is built up from arithmetic operations and standard transitive functions, the domain of definition, $Y \setminus \Omega_i$, is usually built up from intervals and can be held under

control. Matters can be simplified, if generalized subdifferentials are admitted that contain the improper values $\pm\infty$.

We will return to the four examples and realize the concept just discussed in order to obtain inclusions for the subdifferentials.

EXAMPLE 6.1. The derivative and subdifferential of the function in Example 5.1 is

$$\begin{aligned} f'(x) &= \sin x + x\cos x \quad \text{if } x > 0, \\ &= \cos x - x\sin x \quad \text{if } x < 0, \\ \partial f(x) &= [0,1] \qquad\qquad \text{if } x = 0. \end{aligned}$$

Accordingly, the inclusion function of derivative and subdifferential, which is obtained via the natural interval extension of these expressions is

$$\begin{aligned} F'(Y) &= \sin Y + Y\cos Y && \text{if } Y > 0, \\ &= \cos Y - Y\sin Y && \text{if } Y < 0, \\ &= (\sin Y_1 + Y_1\cos Y_1) \vee (\cos Y_2 - Y_2 \sin Y_2) && \text{if } 0 \in Y \end{aligned}$$

where $Y = Y_1 \cup Y_2,\ Y_1 \geqslant 0,\ Y_2 \leqslant 0$. Again, if the splitting of Y in the case $0 \in Y$ is not opportune or not possible, one simply can take

$$F'(Y) = (\sin Y + Y\cos Y) \vee (\cos Y - Y \sin Y) \text{ if } 0 \in Y$$

without being too crude if Y is small.

EXAMPLE 6.2. The plain natural interval extension of the expressions of the three function branches as defined in Example 5.2 leads to the following inclusion function, $F' : \mathbf{I}^2 \to \mathbf{I}^2$, of the subdifferential function ∂f of f:

$$\begin{aligned} F'(X,Y) &= (1,-1) && \text{if } X + Y > 1, \\ &= 2(X - Y,\ -X + Y) && \text{if } X + Y < 1, \\ &= (1,-1) \vee 2(X - Y, -X + Y) && \text{if } 1 \in X + Y. \end{aligned}$$

EXAMPLE 6.3. Let $F_i' : \mathbf{I}^n \to \mathbf{I}^n$ be an inclusion function of ∂f_i for the function f_i, $i = 1, \ldots, k$, as defined in Example 5.3. Then, for $i = 1, \ldots, k$, the inclusion $F_i'(Y)$ need not be part of the whole inclusion $F'(Y)$, if it is guaranteed that f_i has no essential influence to the objective function f over Y, as was the case already at the construction of $F(Y)$ (cf. Example 5.3). Therefore, an inclusion $F'(Y)$ of $\partial f(Y)$ can be composed as the box hull of those inclusions $F_i'(Y)$, for which

$$F_s(Y) \cap \operatorname{int} F_i(Y) \neq \emptyset, \quad i \in \{1, \ldots, k\}.$$

EXAMPLE 6.4. The derivative of f is

$$f'(x) = \log x + 1, \text{ if } x \in (0,1].$$

If the extended real axis is admitted for derivative values, we get

$$f'(0) = \partial f(0) = -\infty.$$

A realistic inclusion function of f' is

$$F'(Y) = \log Y + 1.$$

If an infinite interval arithmetic is included in the software package, the evaluation of $F'(Y)$ can be arranged automatically by the program without any correction or help by the user's hand. (cf. for example [41, 43]) Further, if $0 \leqslant Y < 1/e$, then $F'(Y) < 0$, which indicates strict monotonicity. This information can be used to obtain better inclusion functions of the objective function, f, cf. Example 5.4.

REMARK 6.1. Theoretical results on the order of inclusion for nonsmooth functions do not yet exist. Nevertheless, it can be seen from our examples (Examples 5.1–5.4, 6.1–6.3) that the overestimation tends to zero as the width of the argument box Y does. Also for more involved functions it is frequently possible to choose the inclusions $F(Y)$ and $F'(Y)$ in such a manner that the overestimation tends to zero. The reason is that f and ∂f depend on branches of arithmetic operations and standard transitive functions and that appropriate inclusions of order one ore two for these branches can be found. (Since $F'(Y)$ is always a box, but $\partial f(Y)$ just a set of rather an arbitrary shape, the overestimation by $F'(Y)$ is understood w.r.t. the box hull of $\partial f(Y)$ and not w.r.t. $\partial f(Y)$ itself.)

REMARK 6.2. A very fortune side effect of the computation of inclusions of $\partial f(Y)$ is that they can immediately be applied to the so-called *monotonicity test*, which is one of the most powerful tools in interval methods for solving optimization problems. The test says, that f is strictly monotone over Y w.r.t. the ith coordinate direction when $0 \notin \partial_i f(Y)$. Here, ∂_i denotes the ith component of the subdifferential. If strict monotonicity is indicated, int Y cannot contain any local or global minimizer of f. This phenomenon can be utilized for several strategies for eliminating subboxes from the search for solutions of the problem domain (cf. for example [41, 42]).

REMARK 6.3. *Influence of rounding errors.* It is well-known that the standard implementations and software packages containing interval arithmetic offer convenient options to control all kind of rounding errors (directed rounding, inward rounding, outward rounding, etc.). Therefore, we abstained from touching the rounding aspect in this paper.

7. Numerical results

We chose three well-known examples which showed a typical performance of our method. We compare our statistics with the computations of Lemaréchal, Schramm,

and Mäkelä–Neittaanmäki as they are published in [30]. However, satisfactory comparisons are not easy to establish. The reasons are, firstly, that the obtained approximations of the minimizers were not reported. Secondly, the extense of a 'function evaluation' was not quite clear. (In our statistics, it can be the evaluation of functions as well as of gradients and subgradients over a point or over a box.) The third reason depends on the selfvalidation philosophy, which underlies almost all interval arithmetic methods and our method too. Therefore, it is not enough to determine approximate solutions, but also to compute guaranteed error bounds for them, or at least to indicate in the output of the computation that such bounds could not be found within the computation time or costs limits predescribed by the user. Our algorithm terminates if either an exact solution can be guaranteed to lie in the current box or it might be unreasonable to continue the computation since the global phase had to be repeated several times.

In the following examples, the termination parameter, δ, was set to 10^{-15}, and the box width, ϵ, to 2×10^{-6}.

EXAMPLE 7.1. *(Crescent).* The objective function is given by

$$f(x) = \max\{x_1^2 + (x_2 - 1)^2 + x_2 - 1, -x_1^2 - (x_2 - 1)^2 + x_2 + 1\},$$

the global minimizer is $x^* = (0, 0)$, and the global minimum is $f^* = 0$. The function is not smooth at x^*. Starting point is $x_0 = (-1.5, 2)$. Lemaréchal, Schramm, and Mäkelä-Neittaanmäki needed 31, 24, and 32 iterations, respectively, and 93, 27, and 33 function evaluations, respectively, in order to get an approximate solution x^+ (which was not reported) having a function value of about 10^{-6}. We needed 7 iterations with 58 function evaluations till the local phase was terminated. The final box at the local phase was

$$X^* = ([-9.9999, 10],\ [-10.2895,\ 9.71]) \times 10^{-6}$$

and $[-3.086881,\ 3.028974] \times 10^{-5}$ the final approximation of the minimum.

The global phase which was following the local phase needed 12 function evaluations to confirm that X^* was, in fact, containing a solution.

EXAMPLE 7.2. *(DEM).* The objective function is given by

$$f(x) = \max\{5x_1 + x_2,\ -5x_1 + x_2,\ x_1^2 + x_2^2 + 4x_2\},$$

the global minimizer is $x^* = (0, -3)$, and the global minimum is $f^* = -3$. The function is not smooth at x^*. Starting point is $x_0 = (1, 1)$. Lemaréchal, Schramm, and Mäkelä–Neittaanmäki needed 10, 9 and 7 iterations, respectively, and 33, 13, and 8 function evaluations, respectively, in order to get a solution x^+ (which was not reported) having function value -3. We needed 3 iterations with 16 function evaluations till the local phase was terminated. The final box of the local phase was

$$X^* = ([-1.885,\ 0.1151] \times 10^{-5},\ [-3.000011,\ -2.999991])$$

and $[-3.0001052,\ -2.9998967]$ the final approximation of the minimum.

The global phase which was following the local phase needed 12 function evaluations to confirm that X^* was, in fact, containing a solution.

EXAMPLE 7.3. *(QL).* The objective function is given by

$$f(x) = \max\{x_1^2 + x_2^2,\ x_1^2 + x_2^2 - 40x_1 - 10x_2 + 40,\ x_1^2 + x_2^2 - 10x_1 - 20x_2 + 60\},$$

the global minimizer is $x^* = (1.2,\ 2.4)$, and the global minimum is $f^* = 7.2$. The function is not smooth at x^*. Starting point is $x_0 = (-1,\ 5)$. Lemaréchal, Schramm, and Mäkelä–Neittaanmäki needed 12, 12, and 17 iterations, respectively, and 30, 17, and 18 function evaluations, respectively, in order to get a solution x^+ (which is not reported) having almost the exact function value. We needed 9 iterations with 105 function evaluations till the local phase was terminated. The final box of the local phase had midpoint (1.06182, 2.46921) and edge length ϵ and was the final approximation of the minimum. The termination was initiated automatically after several repetitions of the global phase steps were called up.

References

1. Androulakis, I.P., Maranas, C.D. and Floudas, C.A. (1995), αBB: A global optimization method for general constrained nonconvex problems, *Journal of Global Optimization* 7: 337–363.
2. Auslender, A., et al. (eds.) (1981), *Lecture Notes in Control and Information Sciences*, Vol. 30. Springer Verlag, Berlin.
3. Berner, S. (1995), *Ein paralleles Verfahren zur verifizierten globalen Optimierung*. Ph.D. Thesis, Wuppertal.
4. Clarke, F.H. (1983), *Optimization and Nonsmooth Analysis*, SIAM. New York.
5. Evtushenko, Y.G. and Potapov, M.A. (1994), Deterministic Global Optimization, in [50], pp. 481–500.
6. Goldstein, A.A. (1977), Optimization of Lipschitz continuous functions, *Mathematical Programming* 13: 14–22.
7. Giannessi, F. (ed.) (1992), *Nonsmooth Optimization: Methods and Applications*. Gordon and Breach, Philadelphia.
8. Hansen, E. (1992), *Global Optimization Using Interval Analysis*. Dekker, New York.
9. Hansen, P., Jaumard, B. and Lu, Ski-Hin (1991), An analytical approach to global optimization, *Mathematical Programming* 52: 227–254.
10. Hansen, P., Jaumard, B. and Xiong, J. (1993), Decomposition and interval arithmetic applied to the global minimization of polynomial and rational functions, *Journal of Global Optimization* 3: 421–437.
11. Hansen, P. and Jaumard, B. (1995), Lipschitz optimization, in [15], pp. 407–493.
12. Herzberger, H. (ed.) (1994), *Topics in Validated Computations*. Elsevier, Amsterdam.
13. Hiriart-Urruty, J.-B. and Lemaréchal, C. (1993), *Convex Analysis and Minimization Algorithms I*. Springer Verlag, Berlin.
14. Hiriart-Urruty, J.-B. and Lemaréchal, C. (1993), *Convex Analysis and Minimization Algorithms II*. Springer Verlag, Berlin.

15. Horst, R. and Pardalos, P.M. (eds.) (1995), *Handbook of Global Optimization*. Kluwer, Dordrecht.
16. Horst, R., Pardalos, P.M. and Thoai, N.V. (1995), *Introduction to Global Optimization*. Kluwer, Dordrecht.
17. Horst, R. and Tuy, H. (1990), *Global Optimization: Deterministic Approaches*. Springer Verlag, Berlin.
18. Jansson, C. (1994), On self-validating methods for optimization problems, in [12], pp. 381–438.
19. Jansson, C. and Knüppel, O. (1995), A branch and bound algorithm for bound constrained optimization problems without derivatives, *Journal of Global Optimization* 7: 297–331.
20. Kearfott, R.B. (1996), A review of techniques in the verified solution of constrained global optimization problems, in [22], pp. 23–59.
21. Kearfott, R.B. (1996), Interval extensions of non-smooth functions for global optimization and nonlinear systems solvers, *Computing* 57: 149–162.
22. Kearfott, R.B. and V. Kreinovich, (eds.) (1996), *Applications of Interval Computations*. Kluwer, Dordrecht.
23. Kiwiel, K.C. (1985), *Methods of Descent in Nonsmooth Optimization*. Springer Verlag, Berlin.
24. Kiwiel, K.C. (1992), A restricted step proximal bundle method for nonconvex nondifferentiable optimization, in [7], pp. 175–188.
25. Lemaréchal, C. (1981), A view of line-searches, in [2], pp. 59–78.
26. Lemaréchal, C. (1989), Nondifferentiable optimization, in [37], pp. 529–572.
27. Lemaréchal, C. (1992), Langrangian decomposition and nonsmooth optimization: Bundle algorithm, prox iteration, augmented langrangian, in [7], pp. 201–216.
28. Lemaréchal, C. and Zowe, J. (1994), A condensened introduction to bundle methods in nonsmooth optimization, in [50], pp. 357–382.
29. McCormick, G.P. (1983), *Nonlinear Programming*. Wiley, New York.
30. Mäkelä, M. and Neittaanmäki, P. (1992), *Nonsmooth Optimization – Analysis and Algorithms with Applications to Optimal Control*. World Scientific, Singapore.
31. Mayne, D.Q. and Polak, E. (1984), Outer approximation algorithm for nondifferentiable optimization problems, *Journal of Optimization Theory and Applications* 42: 19–30.
32. Mayne, D.Q., Polak, E. and Wardi, Y. (1983), On the extension of constrained optimization algorithms from differentiable to nondifferentiable problems, *SIAM Journal of Control and Optimization* 21: 179–203.
33. Mifflin, R. (1982), A Modification and an extension of Lemaréchal's algorithm for nonsmooth minimization, *Mathematical Programming Study* 17: 77–90.
34. Mifflin, R. (1992), Ideas for developing a rapidly convergent algorithm for nonsmooth minimization, in [7], pp. 228–239.
35. Moore, R.E. (1966), *Interval Analysis*. Prentice-Hall, Englewood Cliffs, N.J.
36. Moore, R.E. (ed.) (1988), *Reliability in Computing: The Role of Interval Mathematics*. Academic Press, San Diego.
37. Nemhauser, et al. (1989), *Handbooks in Operation Research & Management Science, Vol. I: Optimization*. Elsevier Science Publishers, Amsterdam.
38. Pintér, J.D. (1996), *Global Optimization in Action*. Kluwer, Dordrecht.
39. Ratschek, H. (1988), Some recent aspects of interval algorithms for global optimization, in [36], pp. 325–339.
40. Ratschek, H. and Rokne, J. (1984), *Computer Methods for the Range of Functions*. Horwood, Chichester.
41. Ratschek, H. and Rokne, J. (1988), *New Computer Methods for Global Optimization*. Horwood, Chichester.
42. Ratschek, H. and Rokne, J. (1995), Interval methods, in [15], pp. 751–828.
43. Ratschek, H. and Voller, R. (1990), Unconstrained optimization over unbounded domains, *SIAM Journal of Control and Optimization* 28: 528–539.

44. Ratz, D. (1992), *Automatische Ergebnisverifikation bei globalen Optimierungsproblemen*. Ph.D. thesis, Karlsruhe.
45. Ratz, D. and Csendes, T. (1995), On the selection of subdivision directions in interval branch-and-bound methods for global optimization, *Journal of Global Optimization* 7: 183–207.
46. Robinson, S.M. (1973), Computable error bounds for nonlinear programming, *Mathematical Programming* 5: 235–242.
47. Schittkowski, K. (ed.) (1985), *Computational Mathematical Programming*. Springer Verlag, Berlin.
48. Schramm, H. and Zowe, J. (1992), A version of the bundle idea for minimizing a non-smooth function: Conceptual idea, convergence analysis, numerical results. *SIAM Journal of Optimization* 2: 121–152.
49. Shor, N.Z. (1985), *Minimization Methods for Nondifferentiable Functions*. Springer Verlag, Berlin.
50. Spedicato, E. (ed.) (1994), *Algorithms for Continuous Optimization: The State of Art*. Kluwer, Dordrecht.
51. Stahl, V. (1995), *Interval Methods for Bounding the Range of Polynomials and Solving Systems of Linear Equations*. Ph.D. thesis, Linz.
52. Törn, A.A. and Žilinkas, A. (1989), *Global Optimization*. Springer Verlag, Berlin.
53. Wolfe, M.A. and Zhang, L.S. (1994), An interval algorithm for nondifferentiable global optimization, *Applied Mathematics and Computation* 63: 101–122.
54. Zowe, J. (1985), Nondifferentiable optimization, in [47], pp. 323–356.