SIAM J. NUMER. ANAL.
Vol. 34, No. 3, pp. 922–938, 1997

SUBDIVISION DIRECTION SELECTION IN INTERVAL METHODS FOR GLOBAL OPTIMIZATION

T. CSENDES† AND D. RATZ‡

Abstract. The role of the interval subdivision selection rule is investigated in branch-and-bound algorithms for global optimization. The class of rules that allow convergence for the model algorithm is characterized, and it is shown that the four rules investigated satisfy the conditions of convergence. A numerical study with a wide spectrum of test problems indicates that there are substantial differences between the rules in terms of the required CPU time, the number of function and derivative evaluations and space complexity, and two rules can provide substantial improvements in efficiency.

Key words. global optimization, interval arithmetic, interval subdivision

AMS subject classifications. 65K05, 90C30

Abbreviated title: Subdivision directions in interval methods.

1. Introduction. Interval subdivision methods for global optimization [7, 21] aim at providing reliable solutions to global optimization problems

$$\min_{x \in X} f(x) \tag{1}$$

where the objective function $f : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable, and $X \subseteq \mathbb{R}^n$ is an n-dimensional interval. In many cases, only the globally optimal solutions are acceptable [4, 22], and the local minima are less important. No special problem structure is required: only inclusion functions of the objective function and its gradient are utilised [1]. Denote the set of compact intervals by $\mathbb{I} := \{[a, b] \mid a \leq b,\ a, b \in \mathbb{R}\}$ and the set of n-dimensional intervals (also called simply intervals or boxes) by $\mathbb{I}^n$. We call a function $F : \mathbb{I}^n \to \mathbb{I}$ to be an *inclusion function* of $f : \mathbb{R}^n \to \mathbb{R}$ in X, if $x \in Y$ implies $f(x) \in F(Y)$ for each interval Y in X. In other words, $f(X) \subseteq F(X)$, where $f(X)$ is the range of $f(x)$ on X. The inclusion function of the gradient of $f(x)$ is denoted by $F'(X)$.

There are several ways to build an inclusion function for a given optimization problem (e.g. by using the Lipschitz constant). Interval arithmetic [1, 6, 7, 21] is a convenient tool for constructing the inclusion functions, and one can get those for almost all functions that can be calculated by a finite algorithm (i.e. not only for given expressions).

It is assumed in the following that the inclusion functions have the *isotonicity* property, i.e. $X \subseteq Y$ implies $F(X) \subseteq F(Y)$, and that for all the inclusion functions

$$w(F(X^i)) \to 0 \ \text{ as } \ w(X^i) \to 0 \tag{2}$$

holds, where $w(X)$ is the width of the interval X ($w(X) = \max X - \min X$ if $X \in \mathbb{I}$, and $w(X) = \max_{i=1}^{n} w(X_i)$, if $X \in \mathbb{I}^n$).

The generality of the problem class and the modest requirement of the existence of the inclusion functions stress the importance of each improvement in the efficiency

† Department of Applied Informatics, József Attila University, Szeged, Árpád tér 2., Hungary (E-mail: csendes@inf.u-szeged.hu). The work has been supported by the Grants CIPA3510CT926141 of the COST Project of the EC, OTKA 2879/1991, and MKM 414/1994.

‡ Institut für Angewandte Mathematik, Universität Karlsruhe, Kaiserstraße 12, D–76128 Karlsruhe, Germany (E-mail: ae07@rz.uni-karlsruhe.de).

of the interval global optimization methods. After studying the effects of some accelerating tools [5], the present paper investigates the role of the selection of the interval subdivision direction.

2. Model algorithm and subdivision direction selection rules. First we give a simple model algorithm that has the most important common features of the interval subdivision methods for global optimization (cf. [2, 3, 5, 6, 7, 16, 21, 22]). No local search procedure is included. The Newton-like steps are also not built in, since these would need the inclusion of the Hessian. On the other hand, the cut-off and monotonicity tests are applied, because their usage does not require additional information on the problem. It would not make sense to skip these tests. Although cross-effects of the direction selection rules and the skipped steps are possible, the investigation of their numerical implication is subject of an other study.

The model algorithm is as follows:

Step 0 Set $Y = X$, and $y = \min F(X)$. Initialize the list $L = ((Y, y))$ and the cut-off level $z = \max F(X)$.

Step 1 Choose a coordinate direction $k \in \{1, 2, \ldots, n\}$.

Step 2 Bisect Y in direction k: $Y = V^1 \cup V^2$.

Step 3 Calculate $F(V^1)$ and $F(V^2)$, and set $v^i = \min F(V^i)$ for $i = 1, 2$ and $z = \min\{z, \max F(V^1), \max F(V^2)\}$.

Step 4 Remove (Y, y) from the list L.

Step 5 Cut-off test: discard the pair (V^i, v^i) if $v^i > z$ (where $i \in \{1, 2\}$).

Step 6 Monotonicity test: discard the remaining pair (V^i, v^i) if $0 \notin F'_j(V^i)$ for any $j \in \{1, 2, \ldots, n\}$, and $i = 1, 2$.

Step 7 Add the remaining pair(s) to the list L. If the list becomes empty, then STOP.

Step 8 Denote the pair with the smallest second element by (Y, y).

Step 9 If the width of $F(Y)$ is less than ε, then print $F(Y)$ and Y, STOP.

Step 10 Go to Step 1.

The interval Y, that is first set in Step 0, and then updated in Step 8, is called the *leading box*, and the leading box of the iteration number s is denoted by Y^s. Notice that the cut-off test does not have any effect on the convergence of the algorithm, it may just decrease the space complexity [5], the maximal length of list L.

The interval subdivision direction selection rule in Step 1 is the target of our present study. In the following, we describe shortly the four rules discussed. All the rules select a direction by using a merit function:

$$k := \min\left\{ j \mid j \in \{1, 2, \ldots, n\} \text{ and } D(j) = \max_{i=1}^{n} D(i) \right\} \tag{3}$$

where $D(i)$ is determined by the given rule.

Rule A. First the interval width oriented rule was applied [16, 21, 24], this chooses the coordinate direction with

$$D(i) := w(X_i). \tag{4}$$

This rule was justified by the idea that if the original interval is subdivided in a uniform way, then the width of the actual subintervals goes the quickest to zero. It has also been used for generating subdivision direction in other optimization procedures (e.g. [11]).

The algorithm with Rule A is convergent both with and without the monotonicity test (e.g. in [5] and in [21]). This rule allows a relatively simple analysis of the convergence speed (as in [21], Chapter 3, Theorem 6). The usual definition of Rule A does not specify one single coordinate direction if the maximum is achieved many times in (3). That is why we take the smallest one. For the sake of brevity, we call this interval subdivision direction selection rule in the sequel as Rule A.

Rule B. Hansen described an other rule (initiated by G. W. Walster) [7]. The direct aim of this heuristic direction selection rule is to find the component for which

$$\begin{aligned} W_i \quad = \quad & \max_{t \in X_i} f\left(m(X_1), \ldots, m(X_{i-1}), t, m(X_{i+1}), \ldots, m(X_n)\right) \\ & - \min_{t \in X_i} f\left(m(X_1), \ldots, m(X_{i-1}), t, m(X_{i+1}), \ldots, m(X_n)\right) \end{aligned}$$

is the largest (where $m(X_i) = (\min X_i + \max X_i)/2$ is the midpoint of the interval X_i). The factor W_i that should reflect how much f varies as x_i varies over X_i is then approximated by $w(F_i'(X))w(X_i)$. The latter is not an upper bound for W_i (cf. [7] page 131 and Example 2 in Section 3 of the present paper), yet it can be useful as a merit function.

The Rule B selects the coordinate direction, for which (3) holds with

$$D(i) := w(F_i'(X))w(X_i). \tag{5}$$

It should be noted that the model algorithm represents only one way how Rule B was applied in [7]. The subdivision was, e.g., carried out also for many directions in a single iteration step.

Rule C. The next rule of our investigation was defined by Ratz [23]. The underlying idea was to minimize the width of the inclusion:

$$w(F(X)) = w(F(X) - F(m(X))) \approx w(F'(X)(X - m(X))) =$$

$$\sum_{i=1}^{n} w\left(F_i'(X)(X_i - m(X_i))\right).$$

Obviously, that component is to be chosen for which $w(F_i'(X)(X_i - m(X_i)))$ is the largest. Thus, Rule C can also be formulated with (3) and

$$D(i) := w(F_i'(X)(X_i - m(X_i))). \tag{6}$$

The important difference between (5) and (6) is that in Rule C the width of the multiplied intervals is maximized and not the multiplied widths of the respective intervals (and these are in general not equal). After a short calculation, the right-hand side of (6) can be written as $\max\{|\min F_i'(X)|, |\max F_i'(X)|\}w(X_i)$ (cf. [1, 17, 21]). This corresponds to the "*maximum smear*" (used as a direction selection merit function solving systems of nonlinear equations [13]) for the case $f : \mathbb{R}^n \to \mathbb{R}$.

It is easy to see that the Rules B and C give the same merit function value if and only if either $\min F_i'(X) = 0$ or $\max F_i'(X) = 0$. It is worth mentioning, that $\max\{|\min F_i'(X)|, |\max F_i'(X)|\}$ is the best possible estimation of the Lipschitz constant of the (one-dimensional) function $f(x)$ with fixed variables x_j, $j = 1, 2, \ldots, i -$

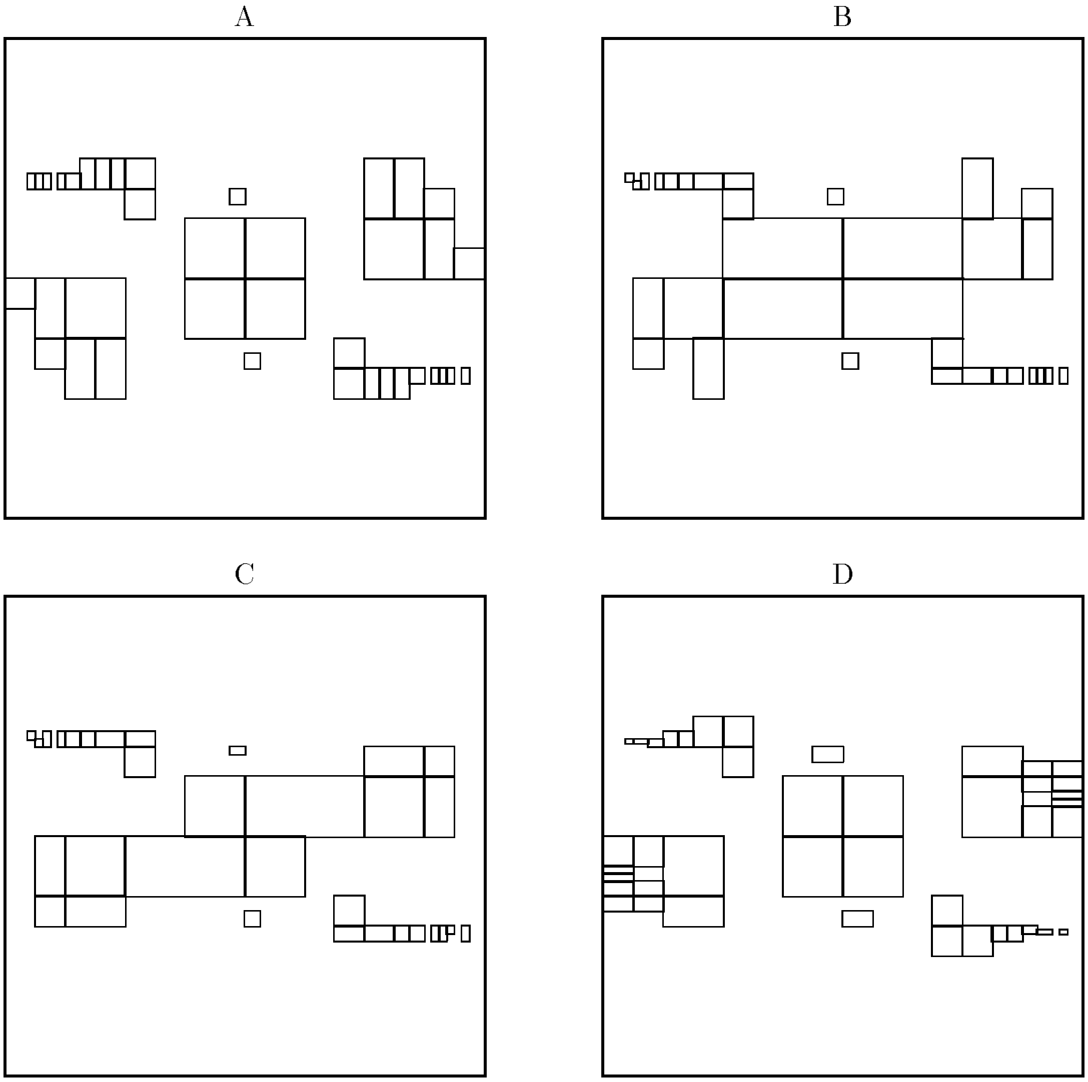

FIG. 1. *Remaining subintervals after 100 iteration steps of the model algorithm with the direction selection rules A, B, C, and D for the Six-Hump-Camel-Back problem.*

$1, i+1, \ldots, n$ for the current box X on the basis of the available inclusion function information. This formulation shows how the model algorithm with the direction selection rule C can be related to Lipschitzian partition methods for global optimization [19, 20].

Rule D. The fourth rule is derivative-free like Rule A, and reflects the machine representation of the inclusion function $F(X)$ (see [6]). It is again defined by (3) and by

$$D(i) := \begin{cases} w(X_i) & \text{if } \ 0 \in X_i, \\ w(X_i)/\min\{|x_i| \mid x_i \in X_i\} & \text{otherwise.} \end{cases} \tag{7}$$

This rule (called Rule D) may decrease the excess width $w(F(X)) - w(f(X))$ of the inclusion function that is caused in part by the floating point computer representation

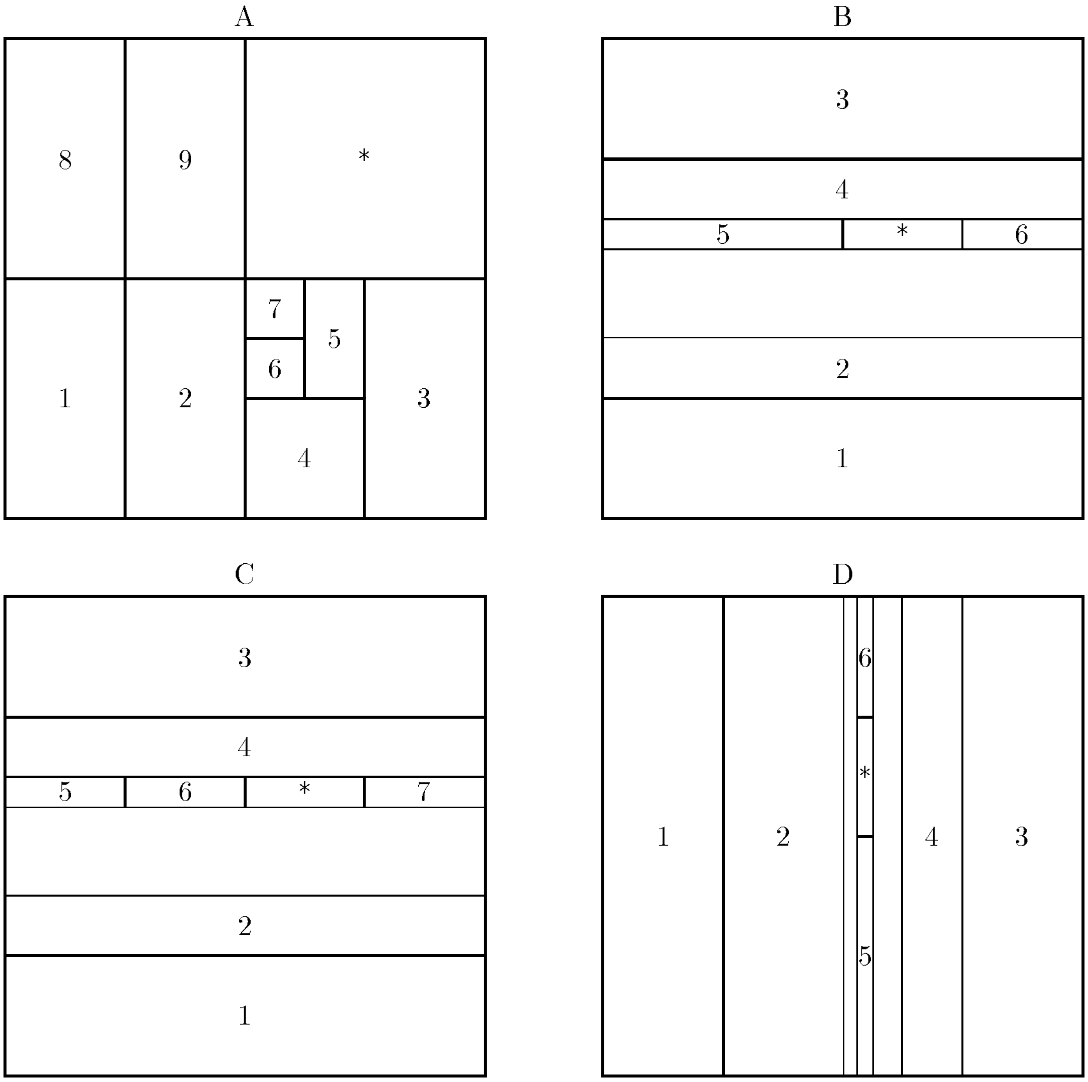

FIG. 2. *Remaining subintervals after 10 iteration steps of the model algorithm with the direction selection rules A, B, C, and D for the EX1 problem.*

of real numbers. Consider the case when the component widths are of similar order, and the absolute value of one component is dominant. The subdivision of the latter component may result in a worse inclusion, since the raster points of the representable numbers are sparser in this direction.

Typical distributions of subintervals are shown in Figure 1 for the discussed direction selection rules A, B, C, and D, respectively. The Six-Hump-Camel-Back standard global optimization test problem [4, 21, 23] was solved with the model algorithm (initial box $[-2.0, 2.0]^2$). The figures show the situations after 100 iterations. The numbers of subintervals are 38, 31, 32 and 42, respectively. These figures reflect the space complexity of the related procedures to a certain extent. The direction selection rule A tends to form square-like boxes, while Rule D produces elongated intervals as the magnitudes of the coordinates differ. Rules B and C generate similar sets of subintervals reflecting the utilised derivative information, too. The sets of subintervals closely fit the respective level sets, and the differences are mainly due to the overestimating inclusion functions. Since the global minimizer points are in the remaining subintervals, the uncertainty in the place of the global minimum has been deceased substantially. Figure 1 shows little about the efficiency of the involved algorithms — it will be addressed in a later section.

A test problem, called EX1 has been constructed to show the differences caused by the different direction selection rules: the simple function $0.1((1-x_1)^2+\sin(10x_1)) + (11-x_2)^2+\sin(10x_2)$ is to be minimized on the initial box of $[0.0, 2.0] \times [10.0, 12.0]$. The sine terms were added to inhibit a fast convergence due to the monotonicity property. Figure 2 provides the sets of subintervals after just 10 iterations with the respective model algorithms. The subboxes denoted by * are those selected for the next subdivision, and the figures in the subintervals indicate their age: the intervals with 1 are the oldest among the remaining boxes. Rule A again tends to form square-like boxes, while the others produce elongated subintervals. The direction preferred by Rule D is different from that chosen often by Rules B and C. Notice that the model algorithm with Rule A was unable to delete a subinterval from the initial box, while the greatest volume decreases were due to Rules B and C.

3. Convergence and the direction selection rules. For the next theoretical study, we define the sequence of intervals that can be produced by the model algorithm, and we specify a property of the direction selection rules that can ensure convergence for our algorithm. With the exception of Rule A [21], no similar convergence investigation has been published.

Assume that the direction selection rules decide using exclusively the information provided by X, $F(X)$ and $F'(X)$ for the interval X to be subdivided. This assumption is valid for Rules A to D. The strategy to utilize all information collected in earlier iteration steps may increase the space complexity substantially.

DEFINITION 3.1. *We call an infinite sequence of intervals $(Y^s)_{s=0}^{\infty}$ an* infinite subdivision sequence *of Y, if $Y^0 = Y$, and if for each nonnegative integer s the box Y^{s+1} is given as $Y_j^{s+1} = Y_j^s$ for $j = 1, \ldots, i-1, i+1, \ldots, n$, and either $Y_i^{s+1} = [\min Y_i^s, m(Y_i^s)]$, or $Y_i^{s+1} = [m(Y_i^s), \max Y_i^s]$, where i is the direction selected by the given rule with Y^s, $F(Y^s)$ and $F'(Y^s)$.*

It is easy to see that if X is not discarded by the monotonicity test and $\varepsilon = 0$, then the set of leading boxes $(Y^s)_{s=0}^{\infty}$ contains at least one infinite subdivision sequence. The set $(Y^s)_{s=0}^{\infty}$ contains only infinite subdivision sequences and finite sequences of subintervals that end with a box Y for which $0 \notin F_j'(Y)$ for some $j \in \{1, 2, \ldots, n\}$. The latter finite sequences do not affect the convergence of the procedure.

DEFINITION 3.2. *We call a direction selection rule* balanced*, if for all intervals X, for all isotone inclusion functions $F(X)$ and $F'(X)$ having property (2), and for each infinite subdivision sequence of X that is a subsequence of the leading boxes $(Y^s)_{s=0}^{\infty}$, the sequence of directions generated by the given rule contains each k of the possible directions $1, 2, \ldots, n$ for which $w(X_k) > 0$ an infinite number of times.*

The name of this property reflects the fact that even though such rules do not necessarily deliver the directions in a uniform way, yet for each direction the distance between two appearances is a finite number of iteration steps.

Denote the set of accumulation points of the sequence $(Y^s)_{s=0}^{\infty}$ by A, the global minimum of $f(x)$ on X by f^*, and the set of global minimizer points of $f(x)$ on X by

X^*. Recall that the inclusion functions $F(X)$ and $F'(X)$ are assumed to be isotone and the property (2) holds for them. Set the stopping criterion parameter ε to zero for the sake of the convergence investigation.

The monotonicity test may discard subintervals containing global minimizer points if they are on the boundary of X. Since the main point in the present study is to investigate the impact of the direction selection rules on the convergence of the model algorithm, we assume that there exists a stationary point $x^* \in X$ for which $f(x^*) = f^*$, and that $w(X) > 0$ (since otherwise the solution requires no search and thus no subdivision).

THEOREM 3.1. *The model algorithm converges in the sense that* $\lim_{s\to\infty} w(Y^s) = 0$ *if and only if the interval subdivision direction selection rule is balanced.*

Proof. 1. Assume that the direction selection rule has the required property, and the model algorithm produces the infinite sequence of intervals $(Y^s)_{s=0}^{\infty}$ for a global optimization problem given by X, F and F'. The sequence $(Y^s)_{s=0}^{\infty}$ has to contain at least one subsequence $(Y^{s_l})_{l=0}^{\infty}$ which is an infinite subdivision sequence of Y. By construction, $(Y^s)_{s=0}^{\infty}$ contains a union of certain infinite subdivision sequences, and some additional boxes that belong to subdivision sequences that were terminated by the monotonicity test. The latter boxes do not affect the value of $\lim_{s\to\infty} w(Y^s)$. Now the property of balancedness holds for each infinite subdivision sequence $(Y^{s_l})_{l=0}^{\infty}$ incorporated in $(Y^s)_{s=0}^{\infty}$, and it implies $\lim_{l\to\infty} w(Y^{s_l}) = 0$. Hence, we have proven that $\lim_{s\to\infty} w(Y^s) = 0$.

2. Assume now that there exists an infinite subdivision sequence for the actual direction selection rule such that an $i \in \{1, 2, \ldots, n\}$ with $w(X_i) > 0$ appears only a finite number of times in the generated sequence of directions. For such an i obviously $\lim_{s\to\infty} w(Y_i^s) > 0$, and thus $\lim_{s\to\infty} w(Y^s) > 0$. $\square$

THEOREM 3.2. *Assume that the interval subdivision direction selection rule is balanced. Then the model algorithm converges to global minimizer points in the sense that* $\lim_{s\to\infty} F(Y^s) = f^*$, $A \neq \emptyset$ *and* $A \subseteq X^*$.

Proof. Having a stationary point $x^* \in X$ with $f(x^*) = f^*$, $0 \in F_i'(Z)$ $(i = 1, 2, \ldots, n)$ holds for each subinterval Z containing it. Such subintervals cannot be deleted by the monotonicity test. Consequently, x^* is in the union of the boxes in the list L in each iteration. As $(Y^s)_{s=0}^{\infty}$ is then an infinite sequence of intervals inside X, the set of accumulation points A cannot be empty. The inclusion property of $F(X)$ and the definition of the leading box imply that $f^* \in F(Y^s)$ for each iteration number $s \in \{0, 1, \ldots\}$.

Now $\lim_{s\to\infty} w(Y^s) = 0$ according to Theorem 3.1. This equation and the property (2) imply $\lim_{s\to\infty} w(F(Y^s)) = 0$. With $f^* \in F(Y^s)$ for each $s = 0, 1, \ldots$ we get $\lim_{s\to\infty} F(Y^s) = f^*$, and thus also $A \subseteq X^*$. $\square$

One direction of the assertions of Theorem 3.1 and Theorem 3.2 is a generalization of some convergence results in [21] for the model algorithm with the studied class of direction selection rules.

The opposite direction of the statements in Theorem 3.2 is not always true: $A \neq \emptyset$ e.g. holds also if the direction selection rule is not balanced. Notice also that $f^* \in F(Y^s)$ was proven without using any special property of the direction selection rule.

THEOREM 3.3. *Assume that the model algorithm converges for a given problem (1) to global minimizer points in the sense that* $\lim_{s\to\infty} F(Y^s) = f^*$, *and thus* $A \subseteq X^*$. *Then either the algorithm proceeds on the problem like an algorithm with a balanced direction selection rule, or there exists a box* $\hat{X} \subseteq X$ *such that* $f(x) = f^*$ *for all* $x \in \hat{X}$, *and* $w(\hat{X}_i) > 0$, $i \in \{1, 2, \ldots, n\}$ *for all coordinate directions that are selected only a finite number of times.*

Proof. Assume that there exists no $\hat{X}$ with the property defined in Theorem 3.3. Then $w(F(Y)) \geq w(f(Y)) > 0$ for all intervals $Y \subseteq X$ with $w(Y) > 0$ for which there exists an $x \in Y$ such that $f(x) = f^*$. Then $\lim_{s\to\infty} F(Y^s) = f^*$ implies $\lim_{s\to\infty} w(Y^s) = 0$, and with Theorem 3.1 we obtain that each direction $i \in \{1, 2, \ldots, n\}$ for which $w(X_i) > 0$ is selected an infinite number of times. □

The main result of Theorem 3.3 is that with the exception of problems for which a box $\hat{X}$ as defined above exists, the direction selection rule must be balanced to ensure convergence to global minimizer points.

COROLLARY 3.4. *The subdivision direction selection Rules A and D are balanced, and thus the model algorithm converges to global minimizer points with each of these rules.*

Proof. Assume for Rules A and D that a global optimization problem is given by X, F and F'. Let $(Y^s)_{s=0}^{\infty}$ be an infinite subdivision sequence of this problem, and let $i \in \{1, 2, \ldots, n\}$ be a coordinate direction for which $w(X_i) > 0$.

Consider first the model algorithm with the direction selection Rule A. Then $D(i)$ is positive according to (4), and i is chosen as the next subdivision direction after $(\lfloor \log_2(\max_{j=1}^{n}(w(Y_j^s))/w(Y_i^s)) \rfloor + 1)^{n-1}$ steps at the latest ($\lfloor . \rfloor$ denotes the largest integer that is not greater than the argument). This expression gives 1 if $w(Y_i^s)$ is the largest, and it is finite for all compact intervals (with $w(Y_i^s) > 0$). Having i selected as a subdivision direction after a finite number of iteration steps, i must appear an infinite number of times for each infinite subdivision sequence.

Consider now the model algorithm with Rule D, and assume again that a coordinate direction i with $w(X_i) > 0$ is chosen only a finite number of times. Then $D(i)$ is constant and positive (cf. (7)) after a finite number of steps. Consider only that part of the sequence $(Y^s)_{s=1}^{\infty}$, for which $D(i)$ is constant and positive. Let j be another index that is selected an infinite number of times. Although $D(j)$ is not necessarily monotonously decreasing, $D(j) \to 0$ for Y^s as $s \to \infty$. Then there exists an s_0 with $D(j) < D(i)$ for $s > s_0$, and this is a contradiction. □

COROLLARY 3.5. *Either the subdivision direction selection Rules B and C choose each direction* $i \in \{1, 2, \ldots, n\}$ *for which* $w(X_i) > 0$ *an infinite number of times, and thus the model algorithm converges to global minimizer points with each of these rules, or the algorithm converges to a subinterval of* X *with a positive width that contains only global minimizer points.*

Proof. Assume again for Rules B and C that a global optimization problem is given by X, F and F'. Let $(Y^s)_{s=0}^{\infty}$ be an infinite subdivision sequence of this problem, and let $i \in \{1, 2, \ldots, n\}$ be a coordinate direction for which $w(X_i) > 0$.

Consider the model algorithm with the direction selection Rule B. Now $D(i) = 0$ means that $w(F_i'(Y^s)) = 0$, i.e. either $F_i'(Y^s) \neq 0$, and Y^s is discarded by the monotonicity test, or $F_i'(Y^s) = 0$, so that $f(x)$ is independent of x_i. In the latter case, no subdivision is required with respect to i. Hence we can assume $D(i) > 0$ without loss of generality. Assume that i is selected only a finite number of times. Then $w(Y_i^s)$ remains constant and positive after a finite number of steps. Let j be another index that is chosen an infinite number of times. Then $w(F_i'(Y^s))w(Y_i^s) < w(F_j'(Y^s))w(Y_j^s)$, and with $w(Y_j^s) \to 0$ one obtains $w(F_i'(Y^s)) \to 0$. Thus the global minimum f^* must be attained at each point of a result interval $\lim_{s=0}^{\infty} Y^s = Y^*$, since an interval Y^s with $0 \notin F_i'(Y^s)$ would have been discarded by the monotonicity test. Thus either the algorithm converges to a subinterval of X that contains only global minimizer points, or each direction is selected an infinite number of times.

Now consider the model algorithm with the subdivision direction selection Rule C, and assume that j is such a coordinate direction that is selected an infinite number of times. Then $Y_j^s - m(Y_j^s)$ converges to zero during the iteration. Having $F_j'(Y^s)$ bounded and isotone, $D(j) \to 0$ as $s \to \infty$. Now for each $i \in \{1, 2, \ldots, n\}$ for which $w(X_i) > 0$, the merit function $D(i)$ is nonnegative in the beginning. Then $D(i) \to 0$ for Y^s, $s \to \infty$, and either the direction i is selected infinitely many times, or $\lim_{s=1}^{\infty} F_i'(Y^s) = 0$ because of $w(Y_i^s) > 0$ for each s. In the latter case each point of the interval $\lim_{s=0}^{\infty} Y^s = Y^*$ is a global minimizer point. ∎

EXAMPLE 1. We want to find global minimizer points of problem (1) with $f(x) = x_1^2 x_2^4$ on the interval $X = [0.0, 1.0]^2$. We use the range functions as inclusion functions, so $F(X) = f(X) = X_1^2 X_2^4$, $F_1'(X) = 2X_1X_2^4$, and $F_2'(X) = 4X_1^2X_2^3$. Using the direction selection Rule C with the model algorithm, $D(1) = w(F_1'(Y^0)(Y_1^0 - m(Y_1^0))) = w([0.0, 2.0][-0.5, 0.5]) = w([-1.0, 1.0]) = 2$ and $D(2) = w(F_2'(Y^0)(Y_2^0 - m(Y_2^0))) = w([0.0, 4.0][-0.5, 0.5]) = 4$ for $Y^0 = X$. Thus the second coordinate is selected for subdivision, and the next leading box is $Y^1 = [0.0, 1.0] \times [0.0, 0.5]$. Y^1 yields $D(1) = 0.125$ and $D(2) = 0.25$. The procedure converges to the interval $Y^* = [0.0, 1.0] \times [0.0, 0.0]$ without a single subdivision in the first coordinate. According to the comment after the definition of Rule C, the merit function values and the selected directions are the same with Rule B, i.e. the same result interval is obtained by the inclusion of Rule B.

EXAMPLE 2. Our algorithm with Rule B may become non-convergent if we remove the monotonicity test. Consider e.g. the function $f(x) = x_1 + x_2^2$. With $F_1' = [1.0, 1.0]$, Rule B chooses always direction $k = 2$ for all intervals with $w(X_i) > 0$, $i = 1, 2$. In this case $\lim_{s\to\infty} \min F(Y^s) \neq \lim_{s\to\infty} \max F(Y^s)$, where Y^s is again the leading box Y in the iteration number s. Although the probability to have this phenomenon in real-life problems is small, yet it is worth to note this behavior that differs from those of the other rules.

The aimed problem class is obviously too wide to allow meaningful theoretical comparisons between the studied subdivision rules. The next section shows results of extensive numerical testing.

4. Numerical experiences. The numerical tests were carried out on an IBM RISC 6000-580 workstation, coded in Fortran-90 with an implemented interval arithmetic package handling the outside rounding necessary for the inclusion functions. The authors thank R. B. Kearfott and W. V. Walter for their kind help in providing the interval arithmetic package [12] and the necessary modules. The inclusion functions were produced by the natural interval extension that fulfils the assumptions made in Section 1: the isotonicity and property (2). In this straightforward way, to transform a subroutine calculating a real function to the interval version, one simply has to write a new statement to include the interval module, change the data types from real to interval, and rename some function calls. This procedure is much simpler, quicker and less error prone than the earlier one in FORTRAN-77, when all the operations were transformed to function calls on new data structures. More sophisticated inclusion functions (like in [10] or [21]) would result in better efficiency figures at the cost of additional calculations or preliminary reformulations on the involved functions.

The inclusions for the gradients were calculated componentwise, and in this way some of the component calculations could be skipped if the monotonicity test showed that the objective function was monotonous in a variable. On the other hand, we could not make use of the possible joint computations for many gradient components. The code for the gradients was calculated symbolically, that is neither automatic nor numerical differentiation was used. The effects of using alternative ways of the gradient inclusion are the subject of a future study.

The stopping criterion parameter ε was set to 0.01 in each test. The list L was implemented as a simple array. The list was not fully ordered, the program just kept track of the three first list members. This implementation can be efficient for short lists, while problems of large memory complexity require data structures like the AVL-trees or other search trees [9, 11]. The implementation of the list can affect the required CPU time, but not the number of objective function and derivative evaluations. The memory complexity is invariant regarding the data structure in terms of the maximal number of items to be saved, but the size of the data structure can be decreased using the available information [11].

The numerical tests involved the set of standard global optimization problems (definitions in [25], further numerical results in [4, 5, 10, 23]) and the set of test problems studied in Hansen's book (descriptions in [7], additional test results in [10, 23]). In some cases, where slight alterations were found in the problem definitions (in the expression or in the search region), the first published versions were chosen. The test program and the related input files can be found at ftp.jate.u-szeged.hu per anonymous ftp in the directory /pub/math/optimization/article. Since the original Schwefel 3.7 problem was very easy to solve, we multiplied the original objective function by 10^4. This change can be interpreted as solving the original problem with $\varepsilon = 10^{-6}$.

These problems were completed by three new ones. The first one, called EX1 was defined in Section 2. EX2 is a simplified real life parameter estimation problem [8, 18]:

$$f(x) = \sum_{i=1}^{6} \left| f_i - \left(x_1 + \frac{x_2}{\omega_i^{x_3}} - \imath \left(\omega_i x_4 - \frac{x_5}{\omega_i^{x_3}} \right) \right) \right|^2,$$

where the f_i-s are $5.0 - 5.0\imath$, $3.0 - 2.0\imath$, $2.0 - \imath$, $1.5 - 0.5\imath$, $1.2 - 0.2\imath$ and $1.1 - 0.1\imath$, and $\omega_i = \pi i/20$, for $i = 1, 2, ..., 6$. The initial interval is $[0.0, 1.0]^2 \times [1.1, 1.3] \times [0.0, 1.0]^2$.

EX3 is a hard global optimization problem for stochastic algorithms [4]: $f(x) = \sum_{i=1}^{4} x_i^6(sin(1/x_i) + 2)$ if $\Pi_{i=1}^{4} x_i \neq 0$, and $f(x) = 0$ otherwise. The initial box for EX3 is $[-2.0, 2.0]^4$.

4.1. Numerical test results. All the test problems were solved. The problem names in the Tables reflect the first definitions, Si stands for Shekel-i, Hi for Hartman-i, GP for Goldstein-Price, SHCB for Six Hump Camel Back, RCOS for Branin RCOS, RB for Rosenbrock, THCB for Three Hump Camel Back, Li for Levi No. i, and SCij for Schwefel No. i.j.

Tables 1 to 4 contain the efficiency measures provided solving the test problems. The second column contains the dimension of the problem, and the efficiency measures obtained for Rules B, C and D are also expressed as percentages of the respective figure for Rule A. In the last lines, that computational effort is shown which is necessary to solve the whole set of test problems, or to solve the subset of 6 problems with significant complexity (denoted by /sig.). The latter subset (problems H6, GP, L3, L5 SC12 and EX2) required about 99% of the computational burden. The percentages in these lines show how much effort is needed with the actual rule compared to the value by Rule A.

This is the anticipated ratio of improvement (if smaller than 100%) solving a large set of problems similar to the studied one. The average of the percentages (AoP) reflects the relative computational burden one can expect on a single problem if the given rule is used instead of Rule A according to the statistical information provided by the set of test problems.

Table 1 contains the CPU time values required for the test problems with the four direction selection rules. The standard time unit (STU, 1000 evaluations of the non-interval Shekel-5 function) was on the used workstation 0.0036 Sec. The large CPU times measured in STU are in part due to the interval implementation (cf. [14]) and the overhead of the list manipulation. The CPU values are in general proportional to the number of objective function (NFE) and derivative evaluations (NDE). The exceptions are the cases with high memory complexity.

According to the CPU times, Rules B and C are better choices than Rules A and D. On the basis of the numerical study made, we can expect 7% and 6% improvements, respectively, in the computation time if we use Rule B or C instead of Rule A, while Rule D causes about a ninefold increase. Completing a large set of problems similar to the test problems, Rule B needs 90% less, Rule C 91% less, and Rule D about eleven times more CPU time. For the subset of hard test problems, similar tendencies can be seen with larger differences on individual problems. The few bigger CPU time values for the new rules in Table 1 are basically due to the larger number of objective function and derivative evaluations.

Table 2 shows the number of objective function evaluations necessary to solve the test problems. In practical applications, this measure together with the number of derivative evaluations is more important than the required CPU time, because the computation of the involved functions are usually longer than those of the test problems (see e.g. [15, 22]). According to the test results, 7% improvement can be expected if Rules B or C are applied instead of Rule A, and Rule D causes 102% higher number of function evaluations. The sum of the numbers of function evaluations (and also that of the derivatives) must be interpreted with care, because the complexities of the test problems are different. When a similar set of problems is to be solved, the anticipated improvements are as high as 72% for Rule B and 74% for Rule C, while Rule D means about five times more function evaluations. In the case of the subset of hard test problems, the changes are -30%, -31% and $+644\%$ for a single problem, and -72%, -74% and $+426\%$ for a similar set of problems.

Table 3 gives the number of partial derivative evaluations. As mentioned earlier, the inclusions of the gradients were calculated componentwise, i.e. NFE multiplied by the dimension of the problem is an upper bound on the NDE. There is a remarkable stability in the NDE/(n NFE) values: they are between 80 and 99%, and the most of them even lie between 85 and 95%. The only exception is the problem Schwefel Nr. 1.2, for which this ratio is between 55 and 63%, with much larger differences with the used direction selection rules as usual. The monotonicity test deletes those subintervals on which the objective function proves to be monotonous, thus the ratio of such subintervals compared to the total number of generated subintervals cannot be high. This can be an explanation for the relative stability of the NDE/(n NFE) values, since the number of derivative evaluations can then be less than n for the deleted subintervals.

According to the test results, 7% improvement can be expected again if Rules B or C are applied instead of Rule A. Rule D causes 98% more derivative evaluations.

When a similar set of problems is to be solved, the anticipated improvements are as high as 75% for Rule B and 77% for Rule C, while Rule D means about three times more derivative evaluations. In the case of the subset of hard test problems, the respective changes are −28%, −30% and +619% for a single problem, and −75%, −77% and +221% for a similar set of problems.

Table 4 provides the minimal list lengths necessary to solve the test problems with the given direction selection rules. The joint space complexity of the whole set of test problems is characterized by the maximal value for a rule. Since the results are identical for the subset of hard problems, this line is skipped for them.

According to the test results, a list of length 68 714 is enough to solve the set of test problems with Rule A, while the list lengths for the other rules were 13 898, 12 855 and 486 382, respectively. These mean −80%, −81% and +608% differences. The average list length required was 2 062 with Rule A, 640 with Rule B (−69%), 616 with Rule C (−70%), and 15 645 with Rule D (+659%). The average of the percentages were 101%, 103% and 234%, respectively. The differences in performance on the hard problems were similar: the average list length was 13 278 with Rule A, 4 038 with Rule B (30%), 3 876 with Rule C (29%), and 101 567 with Rule D (765%). The average of the percentages for the hard problems were 93%, 95% and 954%, respectively.

Two dominant behaviors can be recognized mainly in Tables 2 and 3, but also in a smaller extent in Tables 1 and 4. For about half of the test problems, the differences caused by the changing the subdivision direction selection rules are moderate. In a smaller subset of test problems, Rule B, and especially Rule C provide a much more efficient solution than Rule A, while Rule D is the worst in this sense. The effects described in the previous paragraphs are even stronger for this second subset of problems. The remaining test problems (about 10%) show various other patterns.

4.2. Statistical evaluation. We used the nonparametric Wilcoxon signed rank test to study the effects of the algorithmic changes to the efficiency measures CPU time, number of objective function and derivative evaluations and space complexity. The normality test failed for each group of data (columns A, B, C and D of the Tables 1 – 4, $P = 0.05$).

The changes in the required CPU time that occurred with the substitution of Rule A by Rules B, C and D, respectively, are greater than would be expected by chance; there are statistically significant changes ($P = 0.037$, $P = 0.044$, and $P = 0.012$). The differences between other pairs (e.g. Rule B vs. Rule C) are not significant. That is, just the substitutions of Rule A cause significant changes in the CPU times.

Regarding the number of objective function evaluations, only the change due to the padding Rule B by Rule C is not significant ($P = 0.406$), all the others are significant. The same substitutions provide significant differences in the number of derivative evaluations, and the P value for the nonsignificant case (for Rules B and C) is 0.587. In other words, the transition inside the pair Rule B and Rule C causes no significant difference in the NFE and NDE values, while each transition between the other pairs provides a significant change. No subdivision rule substitution caused a significant difference in the memory complexity.

The same statistical study was repeated for the smaller data set of the harder problems. The only cases where statistically significant differences could be found ($P = 0.031$ for each) were the ones between Rules B and D; and between Rules C and D in terms of the number of objective function evaluations.

5. Summary and conclusions. Compared to stochastic methods, the interval methods for global optimization are able to provide guarantied reliability solutions — at the cost of sometimes substantially higher computational and space complexity. The present study aimed to investigate the possibilities to improve the efficiency while keeping the reliability. A property of the interval subdivision rules (balanced) was defined that ensures convergence for the studied model algorithm: both $\lim_{s\to\infty} w(Y^s) = 0$, and $\lim_{s\to\infty} F(Y^s) = f^*$. We showed that the opposite direction is also true with some trivial exceptions. It was proved that Rules A and D are balanced, and thus the related interval global optimization algorithms are convergent in both senses. For some problems Rules B and C do not fulfil the requirements of balancedness, yet the algorithms converge also in such cases to the global minimum, and the result sets are positive width intervals the points of which are all global minimizers.

Summarizing the numerical experiences, we can conclude that Rules B, C, and in certain cases also Rule D may be successful alternatives to Rule A. According to our test results, Rule C is definitely the best choice as a direction selection rule, closely followed by Rule B. The poor overall performance achieved with Rule D is in part due to the fact that there are no huge differences in the magnitudes of the variables in the set of test problems: neither in the initial intervals, nor in the global minimizer point coordinates. With early recognition of the problem type, one can save substantial amount of computational effort by using one of the latter rules. For some problems, the application of a new rule can result in dramatic improvements in the efficiency measures — or they can even make it possible to solve a problem due to decreased memory complexity. Some important features of the discussed algorithmic improvements must be highlighted again: they do not require additional information on the problems, and they provide those improvements on a very wide problem class.

REFERENCES

[1] G. ALEFELD AND J. HERZBERGER, *Introduction to Interval Computations*, Academic Press, New York, 1983.

[2] H. P. BENSON, *On the convergence of two branch-and-bound algorithms for nonconvex programming problems*, J. Optim. Theory and Appl., 36 (1982), pp. 129–134.

[3] O. CAPRANI, H. HANSEN AND K. MADSEN, *Combining real and interval methods for global optimization*, Paper presented at the SCAN-93 Conference in Vienna, 1993.

[4] T. CSENDES, *Nonlinear parameter estimation by global optimization — efficiency and reliability*, Acta Cybernetica, 8 (1988), pp. 361–370.

[5] T. CSENDES AND J. PINTÉR, *The impact of accelerating tools on the interval subdivision algorithm for global optimization*, European J. of Operational Research, 65 (1993), pp. 314–320.

[6] R. HAMMER, M. HOCKS, U. KULISCH AND D. RATZ, *Numerical Toolbox for Verified Computing I.*, Springer-Verlag, Berlin, 1993.

[7] E. HANSEN, *Global optimization using interval analysis*, Marcel Dekker, New York, 1992.

[8] Z. HANTOS, B. SUKI, T. CSENDES, B. DARÓCZY AND S. NAGY, *Modeling of low-frequency pulmonary impedance in dogs*, J. of Appl. Physiol., 68 (1990), pp. 849–860.

[9] R. HORST, M. NAST AND N. V. THOAI, *A New LP-Bound in Multivariate Lipschitz Optimization: Application to Unconstrained and Linearly Constrained Problems and to Systems of Inequalities*, Forschungsbericht Nr. 94-08, Universität Trier, 1994.

[10] C. JANSSON AND O. KNÜPPEL, *A global minimization method: the multi-dimensional case*, Report 92.1, Technische Universität Hamburg-Harburg, 1992.

[11] D. R. JONES, C. D. PERTTUNEN AND B. E. STUCKMAN, *Lipschitzian Optimization without the Lipschitz Constant*, J. of Opt. Theory and Appl., 79 (1993), pp. 157-181.

[12] R. B. KEARFOTT, *A FORTRAN-90 Environment for Research and Prototyping of Enclosure Algorithms for Constrained and Unconstrained Nonlinear Equations*, accepted for publication

in the ACM T. on Mathematical Software.

[13] R. B. KEARFOTT AND M. NOVOA, *INTBIS, a Portable Interval Newton/Bisection Package*, ACM T. on Mathematical Software, 16 (1990), pp. 152–157.

[14] R. B. KEARFOTT AND Z. XING, *An interval step control for continuation methods*, SIAM J. Numer. Anal., 31 (1994), pp. 892–914.

[15] B. P. KRISTINSDOTTIR, Z. B. ZABINSKY, T. CSENDES AND M. E. TUTTLE, *Methodologies for tolerance intervals*, Interval Computations, 3 (1993), pp. 133–147.

[16] R. E. MOORE, *Interval Analysis*, Prentice Hall, Engelwood Cliffs, 1966.

[17] A. NEUMAIER, *Interval Methods for Systems of Equations*, Cambridge University Press, Cambridge, 1990.

[18] F. PETÁK, Z. HANTOS, Á. ADAMICZA AND B. DARÓCZY, *Partitioning of pulmonary impedance: modeling vs. alveolar capsule approach*, J. Appl. Physiol., 75 (1993), pp. 513–521.

[19] J. PINTÉR, *Extended univariate algorithms for n-dimensional global optimization*, Computing, 36 (1986), pp. 91–103.

[20] ———, *Convergence qualification of adaptive partition algorithms in global optimization*, Mathematical Programming, 56 (1992), pp. 343–360.

[21] H. RATSCHEK AND J. ROKNE, *New Computer Methods for Global Optimization*, Ellis Horwood, Chichester, 1988.

[22] ———, *Experiments using interval analysis for solving a circuit design problem*, J. Global Optimization, 3 (1993), pp. 501–518.

[23] D. RATZ, *Automatische Ergebnisverifikation bei globalen Optimierungsproblemen.* Dissertation, Universität Karlsruhe, 1992.

[24] S. SKELBOE, *Computation of Rational Interval Functions*, BIT, 4 (1974), pp. 87–95.

[25] A. TÖRN AND A. ŽILINSKAS, *Global Optimization*, Springer-Verlag, Berlin, 1989.

TABLE 1
CPU time in seconds required by the four methods for the solution of the test problems

Problem		Direction selection rules						
Name	n	A	B	(B/A)	C	(C/A)	D	(D/A)
S5	4	0.17	0.16	(94%)	0.17	(100%)	0.17	(100%)
S7	4	0.25	0.24	(96%)	0.24	(96%)	0.25	(100%)
S10	4	0.35	0.35	(100%)	0.35	(100%)	0.36	(103%)
H3	3	0.48	0.22	(46%)	0.21	(44%)	1.07	(223%)
H6	6	5.84	3.66	(63%)	3.09	(53%)	16.30	(279%)
GP	2	558.57	449.94	(81%)	474.15	(85%)	182 203.62	(32 620%)
SHCB	2	0.07	0.07	(100%)	0.06	(86%)	0.09	(129%)
RCOS	2	0.04	0.03	(75%)	0.03	(75%)	0.05	(125%)
RB	2	0.01	0.01	(100%)	0.01	(100%)	0.01	(100%)
THCB	2	0.06	0.05	(83%)	0.05	(83%)	0.07	(117%)
L1	1	0.05	0.05	(100%)	0.05	(100%)	0.05	(100%)
L3	2	6.91	4.76	(69%)	4.76	(69%)	7.21	(104%)
L5	2	2.71	2.09	(77%)	2.09	(77%)	2.77	(102%)
L8	3	0.04	0.04	(100%)	0.04	(100%)	0.04	(100%)
L9	4	0.06	0.06	(100%)	0.06	(100%)	0.06	(100%)
L10	5	0.09	0.09	(100%)	0.09	(100%)	0.09	(100%)
L11	8	0.22	0.22	(100%)	0.22	(100%)	0.22	(100%)
L12	10	0.34	0.34	(100%)	0.34	(100%)	0.34	(100%)
L13	2	0.02	0.02	(100%)	0.02	(100%)	0.02	(100%)
L14	3	0.04	0.03	(75%)	0.03	(75%)	0.04	(100%)
L15	4	0.06	0.05	(83%)	0.06	(100%)	0.06	(100%)
L16	5	0.08	0.08	(100%)	0.08	(100%)	0.08	(100%)
L18	7	0.15	0.14	(93%)	0.14	(93%)	0.16	(107%)
SC12	50	105.14	114.60	(109%)	116.14	(110%)	105.56	(100%)
Beale	2	0.11	0.16	(145%)	0.17	(155%)	0.10	(91%)
SC31	3	0.04	0.04	(100%)	0.04	(100%)	0.04	(100%)
SC31P	3	0.04	0.04	(100%)	0.04	(100%)	0.04	(100%)
Booth	2	0.02	0.02	(100%)	0.02	(100%)	0.02	(100%)
Box	3	0.15	0.29	(193%)	0.29	(193%)	0.15	(100%)
Kowalik	4	0.72	0.31	(43%)	0.31	(43%)	0.14	(19%)
Powell	4	0.17	0.17	(100%)	0.18	(106%)	0.16	(94%)
Matyas	2	0.09	0.09	(100%)	0.09	(100%)	0.09	(100%)
SC37	5	0.02	0.02	(100%)	0.02	(100%)	0.02	(100%)
SC37	10	0.04	0.04	(100%)	0.04	(100%)	0.04	(100%)
SC37	30	0.21	0.21	(100%)	0.21	(100%)	0.21	(100%)
SC32	3	0.02	0.02	(100%)	0.02	(100%)	0.02	(100%)
EX1	2	0.02	0.02	(100%)	0.02	(100%)	0.02	(100%)
EX2	5	18 343.47	1 379.06	(8%)	1 080.00	(6%)	30 842.71	(168%)
EX3	4	0.07	0.07	(100%)	0.07	(100%)	0.07	(100%)
Sum		19 026.94	1 957.86	(10%)	1 684.00	(9%)	213 182.52	(1 120%)
AoP				(93%)		(94%)		(943%)
Sum / sig.		19 022.64	1 954.11	(10%)	1 680.23	(9%)	213 178.17	(1 121%)
AoP / sig.				(68%)		(67%)		(5 562%)

Table 2
Number of function evaluations required by the four methods for the solution of the test problems

Problem		Direction selection rules						
Name	n	A	B	(B/A)	C	(C/A)	D	(D/A)
S5	4	89	89	(100%)	89	(100%)	89	(100%)
S7	4	95	95	(100%)	95	(100%)	95	(100%)
S10	4	97	97	(100%)	95	(98%)	97	(100%)
H3	3	405	183	(45%)	163	(40%)	887	(219%)
H6	6	1 621	1 057	(65%)	893	(55%)	4 505	(278%)
GP	2	112 287	98 565	(88%)	101 997	(91%)	4 176 639	(3 720%)
SHCB	2	345	309	(90%)	305	(88%)	437	(127%)
RCOS	2	101	75	(74%)	75	(74%)	139	(138%)
RB	2	59	49	(83%)	49	(83%)	59	(100%)
THCB	2	265	237	(89%)	239	(90%)	367	(138%)
L1	1	301	301	(100%)	301	(100%)	301	(100%)
L3	2	2 777	1 961	(71%)	1 953	(70%)	2 917	(105%)
L5	2	1 089	853	(78%)	847	(78%)	1 123	(103%)
L8	3	45	45	(100%)	45	(100%)	45	(100%)
L9	4	59	59	(100%)	59	(100%)	59	(100%)
L10	5	73	73	(100%)	73	(100%)	73	(100%)
L11	8	117	117	(100%)	117	(100%)	117	(100%)
L12	10	145	145	(100%)	145	(100%)	145	(100%)
L13	2	45	41	(91%)	41	(91%)	45	(100%)
L14	3	65	59	(91%)	59	(91%)	65	(100%)
L15	4	85	79	(93%)	77	(91%)	85	(100%)
L16	5	105	97	(92%)	95	(90%)	105	(100%)
L18	7	145	133	(92%)	133	(92%)	145	(100%)
SC12	50	1 125	1 111	(99%)	1 111	(99%)	1 125	(100%)
Beale	2	291	447	(154%)	467	(160%)	265	(91%)
SC31	3	61	61	(100%)	61	(100%)	61	(100%)
SC31P	3	61	61	(100%)	61	(100%)	61	(100%)
Booth	2	121	107	(88%)	117	(97%)	97	(80%)
Box	3	91	179	(197%)	177	(195%)	95	(104%)
Kowalik	4	153	71	(46%)	71	(46%)	31	(20%)
Powell	4	563	577	(102%)	591	(105%)	535	(95%)
Matyas	2	765	765	(100%)	765	(100%)	789	(103%)
SC37	5	27	27	(100%)	27	(100%)	27	(100%)
SC37	10	43	43	(100%)	43	(100%)	43	(100%)
SC37	30	91	91	(100%)	91	(100%)	91	(100%)
SC32	3	85	75	(88%)	71	(84%)	85	(100%)
EX1	2	55	53	(96%)	53	(96%)	63	(115%)
EX2	5	971 103	197 529	(20%)	174 009	(18%)	1 551 483	(160%)
EX3	4	169	169	(100%)	169	(100%)	169	(100%)
Sum		1 095 219	306 085	(28%)	285 829	(26%)	5 743 559	(524%)
AoP				(93%)		(93%)		(202%)
Sum / sig.		1 090 002	301 076	(28%)	280 810	(26%)	5 737 792	(526%)
AoP / sig.				(70%)		(69%)		(744%)

TABLE 3
Number of derivative evaluations required by the four methods for the solution of the test problems

Problem		Direction selection rules						
Name	n	A	B	(B/A)	C	(C/A)	D	(D/A)
S5	4	305	303	(99%)	305	(100%)	305	(100%)
S7	4	332	329	(99%)	330	(99%)	332	(100%)
S10	4	340	338	(99%)	330	(97%)	340	(100%)
H3	3	1 000	468	(47%)	424	(42%)	2 131	(213%)
H6	6	8 245	5 242	(64%)	4 445	(54%)	22 934	(278%)
GP	2	199 566	176 292	(88%)	181 494	(91%)	7 129 277	(3 572%)
SHCB	2	618	558	(90%)	554	(90%)	768	(124%)
RCOS	2	184	134	(73%)	134	(73%)	258	(140%)
RB	2	94	75	(80%)	75	(80%)	94	(100%)
THCB	2	474	429	(91%)	430	(91%)	653	(138%)
L1	1	299	299	(100%)	299	(100%)	299	(100%)
L3	2	5 395	3 770	(70%)	3 758	(70%)	5 657	(105%)
L5	2	2 145	1 668	(78%)	1 659	(77%)	2 207	(103%)
L8	3	117	115	(98%)	115	(98%)	117	(100%)
L9	4	204	201	(99%)	201	(99%)	204	(100%)
L10	5	315	311	(99%)	311	(99%)	315	(100%)
L11	8	801	801	(100%)	801	(100%)	801	(100%)
L12	10	1 241	1 241	(100%)	1 241	(100%)	1 241	(100%)
L13	2	82	74	(90%)	74	(90%)	82	(100%)
L14	3	177	162	(92%)	161	(91%)	177	(100%)
L15	4	308	284	(92%)	280	(91%)	308	(100%)
L16	5	475	436	(92%)	433	(91%)	475	(100%)
L18	7	917	832	(91%)	839	(91%)	917	(100%)
SC12	50	31 111	34 184	(110%)	34 723	(112%)	31 111	(100%)
Beale	2	523	809	(155%)	843	(161%)	488	(93%)
SC31	3	166	166	(100%)	165	(99%)	166	(100%)
SC31P	3	166	166	(100%)	165	(99%)	166	(100%)
Booth	2	206	186	(90%)	200	(97%)	166	(81%)
Box	3	259	531	(205%)	525	(203%)	269	(104%)
Kowalik	4	597	254	(43%)	254	(43%)	116	(19%)
Powell	4	1 974	1 986	(101%)	2 041	(103%)	1 901	(96%)
Matyas	2	1 346	1 346	(100%)	1 346	(100%)	1 384	(103%)
SC37	5	108	108	(100%)	111	(103%)	108	(100%)
SC37	10	365	365	(100%)	365	(100%)	365	(100%)
SC37	30	2 355	2 355	(100%)	2 355	(100%)	2 355	(100%)
SC32	3	209	189	(90%)	181	(87%)	209	(100%)
EX1	2	100	98	(98%)	98	(98%)	108	(108%)
EX2	5	3 906 446	800 050	(20%)	708 987	(18%)	6 151 948	(157%)
EX3	4	663	663	(100%)	663	(100%)	663	(100%)
Sum		4 170 228	1 037 818	(25%)	951 715	(23%)	13 362 026	(320%)
AoP				(93%)		(93%)		(198%)
Sum / sig.		4 152 908	1 021 206	(25%)	935 066	(23%)	13 343 134	(321%)
AoP / sig.				(72%)		(70%)		(719%)

TABLE 4

Space complexity of the four methods for the solution of the test problems in terms of the necessary length of the list

Problem		Direction selection rules						
Name	n	A	B	(B/A)	C	(C/A)	D	(D/A)
S5	4	13	14	(108%)	14	(108%)	13	(100%)
S7	4	16	16	(100%)	17	(106%)	16	(100%)
S10	4	16	17	(106%)	17	(106%)	16	(100%)
H3	3	26	17	(65%)	15	(58%)	48	(185%)
H6	6	182	99	(54%)	87	(48%)	405	(223%)
GP	2	9 648	9 381	(97%)	9 463	(98%)	486 382	(5 041%)
SHCB	2	37	36	(97%)	34	(92%)	44	(119%)
RCOS	2	15	11	(73%)	11	(73%)	19	(127%)
RB	2	7	4	(57%)	4	(57%)	7	(100%)
THCB	2	21	21	(100%)	21	(100%)	23	(110%)
L1	1	16	16	(100%)	16	(100%)	16	(100%)
L3	2	820	530	(65%)	525	(64%)	692	(84%)
L5	2	253	200	(79%)	196	(77%)	250	(99%)
L8	3	9	9	(100%)	9	(100%)	9	(100%)
L9	4	12	12	(100%)	12	(100%)	12	(100%)
L10	5	15	15	(100%)	15	(100%)	15	(100%)
L11	8	24	24	(100%)	24	(100%)	24	(100%)
L12	10	30	30	(100%)	30	(100%)	30	(100%)
L13	2	8	9	(113%)	8	(100%)	8	(100%)
L14	3	12	13	(108%)	12	(100%)	12	(100%)
L15	4	16	19	(119%)	19	(119%)	16	(100%)
L16	5	21	23	(110%)	23	(110%)	21	(100%)
L18	7	32	31	(97%)	31	(97%)	32	(100%)
SC12	50	50	120	(240%)	131	(262%)	50	(100%)
Beale	2	53	62	(117%)	63	(119%)	56	(106%)
SC31	3	5	5	(100%)	7	(140%)	5	(100%)
SC31P	3	5	5	(100%)	7	(140%)	5	(100%)
Booth	2	12	13	(108%)	11	(92%)	16	(133%)
Box	3	26	79	(304%)	77	(296%)	26	(100%)
Kowalik	4	69	25	(36%)	25	(36%)	13	(19%)
Powell	4	69	52	(75%)	58	(84%)	85	(123%)
Matyas	2	24	24	(100%)	24	(100%)	26	(108%)
SC37	5	5	5	(100%)	5	(100%)	5	(100%)
SC37	10	10	10	(100%)	10	(100%)	10	(100%)
SC37	30	30	30	(100%)	30	(100%)	30	(100%)
SC32	3	10	13	(130%)	13	(130%)	10	(100%)
EX1	2	14	10	(71%)	10	(71%)	9	(64%)
EX2	5	68 714	13 898	(20%)	12 855	(19%)	121 623	(177%)
EX3	4	80	80	(100%)	80	(100%)	80	(100%)
Maximum		68 714	13 898	(20%)	12 855	(19%)	486 382	(708%)
Average		2 062	640	(31%)	616	(30%)	15 645	(759%)
AoP				(101%)		(103%)		(234%)
Average / sig.		13 278	4 038	(30%)	3 876	(29%)	101 567	(765%)
AoP / sig.				(93%)		(95%)		(954%)