New subinterval selection criteria for interval global optimization *

Tibor Csendes
University of Szeged, Institute of Informatics, Szeged, Hungary

30.12.1999

Abstract. The theoretical convergence properties of interval global optimization algorithms that select the next subinterval to be subdivided according to a new class of interval selection criteria are investigated. The latter are based on variants of the *RejectIndex*: $pf^*(X) = \frac{f^* - \underline{F}(X)}{\overline{F}(X) - \underline{F}(X)}$, a recently thoroughly studied indicator, that can quite reliably show which subinterval is close to a global minimizer point. Extensive numerical tests on 40 problems confirm that substantial improvements can be achieved both on simple and sophisticated algorithms by the new method (utilizing the known minimum value), and that these improvements are larger when hard problems are to be solved.

Keywords: Convergence properties, interval methods, global optimization, interval selection

1. Introduction

Consider the bound constrained global optimization problem (Horst and Pardalos, 1995; Törn and Žilinskas, 1987)

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

where the n-dimensional interval X is the search region, and $f(x) : \mathbb{R}^n \to \mathbb{R}$ is the objective function. We assume that there exists at least one global minimizer point in X, that is also a stationary point. Problems that have only nonstationary global minimizer points (necessarily on the boundary of the search region), can be recognized by interval optimization methods, and they can be solved usually in a relatively easy way, e.g. by the monotonicity test. The algorithm considered is based on inclusion functions calculated by interval arithmetic (Ratschek and Rokne, 1988):

Definition 1. A function $F : \mathbb{I}^n \to \mathbb{I}$ is an *inclusion function* of the objective function f if for $\forall Y \in \mathbb{I}^n$ and $\forall y \in Y \;\; f(y) \in F(Y)$, where $\mathbb{I}$ stands for the set of all closed real intervals.

* This work was supported by the Grants FKFP 0739/97, 0449/99, OTKA T 016413 and T 017241.

In other words, $f(Y) \subseteq F(Y)$ where $f(Y)$ is the range of f over Y. The lower and upper bounds of an interval $Y \in \mathbb{I}^n$ are denoted by $\underline{Y}$ and $\overline{Y}$, respectively. The width of an interval is $w(Y) = \overline{Y} - \underline{Y}$, and $w(Y) = \max_i(\overline{Y_i} - \underline{Y_i})$ if $Y \in \mathbb{I}^n$ is an n-dimensional interval vector (also called a box). $\mathbb{I}(X)$ stands for all intervals in X. Three important types or possible properties of inclusion functions are:

Definition 2. F is said to be an *isotone inclusion function* over X if for $\forall Y, Z \in \mathbb{I}(X),\ \ Y \subseteq Z$ implies $F(Y) \subseteq F(Z)$.

Definition 3. We call the inclusion function F an *α-convergent inclusion function* over X if for $\forall Y \in \mathbb{I}(X)\ \ w(F(Y)) - w(f(Y)) \leq Cw^\alpha(Y)$ holds, where α and C are positive constants.

Definition 4. We say that the inclusion function F has the *zero convergence property*, if $w(F(Z_i)) \to 0$ holds for all the $\{Z_i\}$ interval sequences for which $Z_i \subseteq X$ for all $i = 1, 2, \ldots$ and $w(Z_i) \to 0$.

Denote the global minimum value of the function $f(x)$ on the search region X by f^*. Assume that there exists an isotone inclusion function $F(X)$ for $f(x)$.

Several Branch-and-Bound (B&B) type algorithms have been suggested and studied for the solution of (1) utilizing inclusion function information on the problem (Hansen, 1992; Kearfott, 1996; Ratschek and Rokne, 1988). To allow a general discussion, we study the following algorithm framework that can incorporate most of the features of the present procedures.

Algorithm

Step 1 Let L be an empty list, the leading box $A := X$, and the iteration counter $k := 1$. Set $\tilde{f} = \overline{F}(X)$.

Step 2 Subdivide A into s subsets $A_i, (i = 1, \ldots, s)$ satisfying $A = \cup A_i$ so that $\mathrm{int}(A_i) \cap \mathrm{int}(A_j) = \emptyset$ for all $i \neq j$ where int denotes the interior of a set. Evaluate the inclusion function $F(X)$ for all the new subintervals, and update the upper bound $\tilde{f}$ of the global minimum.

Step 3 Let $L := L \cup \{(A_i, F(A_i))\}$.

Step 4 Discard certain elements from L that cannot contain a global minimum point.

Step 5 Choose a new $A \in L$ and remove the related pair from the list.

Step 6 While termination criteria do not hold let $k := k+1$ and go to Step 2.

In Step 4, so called accelerating devices that delete subintervals not containing a global minimizer point can be used. Such accelerating devices can be for example the cut-off test (for some implementations it is called midpoint test), the monotonicity test, the interval Newton step and the concavity test.

Recently many papers (Casado and García, 1998; Casado et al., 2000a; Casado et al., 2000b) have studied possible ways of utilizing of a new, easy to evaluate indicator, the *RejectIndex*:

$$pf^*(X) = \frac{f^* - \underline{F}(X)}{\overline{F}(X) - \underline{F}(X)}.$$

These studies have proven that this indicator can provide reasonably reliable information on how close a subinterval in the search region is to a global minimizer point. The $pf^*(X)$ parameter proved to be a good basis for effective decisions in load balancing for parallel algorithms (Casado and García, 1998), for a heuristic rejection criterion solving hard optimization problems that have large memory complexity (Casado et al., 2000a), and also for finding the most suitable form of multisection to speed up interval optimization procedures (Casado et al., 2000b; Csendes and Ratz, 1997; Ratz and Csendes, 1995). Now we shall select that interval with the largest pf^* value for subdivision (in Step 5).

The original formulation of this indicator was based on the known value of the global minimum, which is not always available. We introduce here a new expression for a wider class of indicators that still keep the basic properties of $pf^*(X)$:

$$p(\hat{f}, X) = \frac{\hat{f} - \underline{F}(X)}{\overline{F}(X) - \underline{F}(X)},$$

where the $\hat{f}$ value will be a kind of approximation of the global minimum in the subsequent theoretical analysis. In the present investigation we assume that $\hat{f} \in F(X)$, i.e. this estimation is realistic in the sense that $\hat{f}$ is within the known bounds of the objective function on the search region. According to our numerical experience, we need a good approximation of the f^* value to improve the efficiency of the algorithm.

For both expressions, the denominator has a crucial role, and it is where the new selection rules differ from the earlier 'choose the subinterval with the lowest lower bound' rule (Moore and Ratschek, 1988; Skelboe, 1974). The denominator, $\overline{F}(X) - \underline{F}(X)$ is nonnegative due to the properties of the inclusion functions. Theoretically it can also be zero, if either the width of the argument interval is zero, or if the objective function is constant on a positive measure interval. If the search region is a positive measure interval, then the subdivision may produce only positive measure subintervals. It still may happen that one of the accelerating devices (like the monotonicity test) will decrease an interval to a lower dimensional one. Except in constrained problems, not considered here, the possibility that the objective function is constant on a positive measure interval cannot be excluded, although it seems to be not very likely in real life (Moore and Ratschek, 1988).

For both cases the computational implementation may provide a kind of remedy due to the generally applied outward rounding technique, that always produces a positive width inclusion function value. Still the correct solution of these types of problems is to introduce a result list containing those subintervals for which either the width of the interval $w(X)$ or the width of the inclusion $F(X)$ is smaller than a preset positive constant according to the actual machine precision. In the sequel we assume that such a technique exists, and in this way the denominator in the expression of the $p(\hat{f}, X)$ parameter cannot be zero.

The numerators of $pf^*(Y)$ and $p(\hat{f}, Y)$ are not as dangerous, still they deserve some discussion. For some rare but significant problems the lower bound of the objective function on nearly all relevant subintervals can be equal to the global minimum causing $pf^*(Y)$ and thus also $p(f^*, Y)$ to be zero, inhibiting an effective interval selection. Such problems arise for example when the zeros of a function $g(x)$ must be determined by minimizing $f(x) = g(x)^2$, like in some phase equilibrium problems (Stateva and Tsvetkov, 1994). For such problems the suggested interval selection technique is obviously not advantageous.

In Section 2 the convergence properties of a class of interval branch-and-bound optimization methods that use some variants of this indicator as selection criteria for choosing the next leading subinterval to be subdivided are investigated.

2. Theoretical results

2.1. Convergence of algorithm variants

To investigate the convergence properties of the introduced Algorithm, we assume that the stopping conditions cannot be fulfilled. For example, if these are the often used $w(F(Y)) < \varepsilon_1$ and $w(Y) < \varepsilon_2$ for a leading interval Y, or for all the intervals on the list L, then we set $\varepsilon_1 = \varepsilon_2 = 0$. The introduced algorithm will be studied without accelerating devices. Wherever it is applicable, the properties of the algorithm with accelerating tests will be discussed separately.

The global minimum value is not always available. Thus we use an updated approximation f_k of f^*, where k is the iteration number, for the generalized $p(f_k, X)$ algorithm parameter. The following theorem investigates the case when the interval selection criterion parameter f_k converges to a value $\hat{f}$ larger than the global minimum.

Theorem 1. Assume that the inclusion function of the objective function is isotone, it has the zero convergence property, and the $p(f_k, Y)$ parameters are calculated with the f_k parameters converging to $\hat{f} > f^*$, for which there exists a point $\hat{x} \in X$ with $f(\hat{x}) = \hat{f}$. Then the branch-and-bound algorithm that selects that interval Y from the working list which has the maximal $p(f_i, Z)$ value can converge to a point $\hat{x} \in X$ for which $f(\hat{x}) > f^*$, i.e. to a point which is not a global minimizer point of the given problem.

Proof. 1. Consider first the case when $f_k = \hat{f}$ is constant. Assume that a problem $\min_{x \in X} f(x)$ is given for which the studied algorithm has an infinite number of iterations. Such problems exist, see e.g. the problem

$$\min_{x \in [-1,1]} x^2.$$

Since the accelerating devices are switched off, no subinterval will be deleted.

Choose an arbitrary point $\hat{x} \in X$ such that $f(\hat{x}) = \hat{f}$, and consider the interval sequence $\{X_i\}$ that starts from X ($X_0 = X$), and converges to $\hat{x}$ according to the interval subdivision rule of our algorithm. The interval sequence $\{X_i\}$ is unique having a fixed, deterministic interval subdivision rule in our algorithm (with the exception when $\hat{x}$ is on the boundary of a generated subinterval).

Obviously $f_k = \hat{f} \in F(X_k)$ holds for all X_k members of the interval sequence $\{X_i\}$, and thus $f_k - \underline{F}(X_k) \geq 0$. In a similar way, $f_k - \underline{F}(Y) \geq 0$ for all the subintervals that contain global minimizer points. Thus both sets of subintervals have the same signs of the respective $p(f_k, X)$ values.

The inclusion functions of the original problem will be constructed in such a way, that while keeping the usual properties of the inclusion functions, the $p(\hat{f}, X_i)$ values will be 0.9 for all of the intervals of the subsequence $\{X_i\}$ defined above, and 0.5 for all other subintervals.

For the starting box X, set $F(X)$ to be $[\hat{f} - 9d(X), \hat{f} + d(X)]$, where d is a suitable value: $d(X) = \max(\overline{f}(X) - \hat{f}, \hat{f} - \underline{f}(X))$. For this interval

$f(X) \subseteq F(X)$ obviously holds. For those X_i subintervals which contain $\hat{x}$, the inclusion function will be $F(X_i) = [\hat{f} - 9d(X_i), \hat{f} + d(X_i)]$. For all other subintervals the inclusion function will be set to $F(Y) = [\hat{f} - d(Y), \hat{f} + d(Y)]$.

Check the properties of the defined function $F(Y)$:

Inclusion: These definitions give inclusion functions in each case, since for all $x \in Y$ the value $f(x)$ differs from $\hat{f}$ at most by $d(Y)$. While in the last mentioned case, i.e. when $\hat{x} \notin Y$, $f(Y) = F(Y)$ can happen, for subintervals of the sequence $\{X_i\}$ the range of the function is always substantially overestimated by $F(X_i)$. On the other hand, this overestimation depends directly on the width of the range of $f(X)$.

Isotonicity: The defined inclusion function will be isotone, since it will change together with the range $f(X_i)$ of the related intervals if they contain $\hat{x}$: $X_{i+k} \subseteq X_i \to F(X_{i+k}) \subseteq F(X_i)$. When passing from such intervals that contain $\hat{x}$ to subintervals Y for which $\hat{x} \notin Y$, isotonicity will be ensured by the construction:

$$[\hat{f} - d(Y), \hat{f} + d(Y)] \subseteq [\hat{f} - 9d(X_i), \hat{f} + d(X_i)]$$

if $Y \subseteq X_i$ due to the factor 9 in the definition and to the inclusion property of the range of a function. For two subintervals Y_1 and Y_2, for which $\hat{f} \notin Y_i$ $(i = 1, 2)$, and $Y_1 \subseteq Y_2$, the respective d values are also nondecreasing, i.e. $d(Y_1) \leq d(Y_2)$. This ensures the inclusion isotonicity for such intervals.

Zero convergence of $w(F(X_i))$ for interval sequences with $F(X_i) \to \hat{f}$: for such interval sequences the values of $d(X_i)$ converge to zero, since $X_i \to \hat{x}$, and thus $w(X_i) \to 0$. For all $\{Y_i\}$ interval sequences for which $\lim F(Y_i) \neq \hat{f}$, this property cannot hold in our example, since a positive distance to $\hat{x}$ causes positive d values, and thus $w(F(Y_i))$ cannot converge to zero — not even for cases when $w(Y_i) \to 0$.

α-convergence: According to the previous paragraph, α-convergence can only be expected for interval sequences for which $F(X_i) \to \hat{f}$. The α-convergence requires either assumptions on the interval arithmetic, or on the properties of the underlying function. In the present case, e.g. α-convergence can be proved with $\alpha = 1$ if $f(x)$ is Lipschitz continuous with the Lipschitz constant M. Then $|f(y_1) - f(y_2)| \leq M|y_1 - y_2|$ for $y_1, y_2 \in Y$ implies $w(F(Y)) \leq Mw(Y)$, and for an interval sequence $\{Y_i\}$ with $\lim_{i\to\infty} w(Y_i) = 0$, the inequalities $w(F(Y_i)) \leq Mw(Y_i)$ hold for all $i = 1, 2, \ldots$

Having all the necessary properties, we can conclude that the defined inclusion function will result in interval selection decisions in the algorithm, such that only those intervals will be processed further that have $p(\hat{f}, Y) = 0.9$ values, and these converge to $\hat{x}$, for which $f(\hat{x}) \neq f^*$.

Since there exists no subsequence of the leading subintervals that converges to a point with an objective function value different from $\hat{f}$, the missing zero convergence property for these subsequences does not cause any theoretical difficulties.

2. For the case when f_k is not constant, we can use the same construct, and to have the necessary properties, the inclusion function $F(X)$ will be defined again on the basis of $\hat{f}$ (and not on f_k). After a finite number of iterations the subintervals containing the target point x' will be selected as before since the difference between f_k and $\hat{f}$ will be small enough to ensure larger $p(f_k, Y)$ values for these subintervals. □

Together with the constructed problem, there exist obviously an infinity of problems for which the algorithm does not converge to a global minimizer point: e.g. the constructed problem can be transformed to a new one by shifting the objective function (and also the respective inclusion function) by a constant: $f_{\text{new}}(x) = f(x) + f_0$, where $f_0 \neq 0$ is an arbitrary real constant.

Notice that the easy to implement and cheap cut-off test and the zero convergence property can be sufficient to ensure convergence to global optimizer points even if the sequence $\{f_i\}$ does not converge to f^*: for all interval subsequences of the leading intervals, if the sequences converge to a point for which the objective function value is greater than f^*, then the zero convergence property will imply $\underline{f}(Z_i) > \tilde{f}$, i.e. such boxes can be deleted.

To consider the case when f_k converges to $\hat{f} < f^*$, we must assume that the inclusion function $F(Y)$ has the zero convergence property. This is a natural assumption, all the usual techniques have this theoretical feature, and at least the convergence of the lower bounds to the respective real function value is indispensable for the convergence of branch-and-bound algorithms.

Theorem 2. Assume that the inclusion function of the objective function has the zero convergence property, and f_k converges to $\hat{f} < f^*$. Then the optimization branch-and-bound algorithm will produce an everywhere dense sequence of subintervals converging to each point of the search region X regardless of the objective function value.

Proof. Since f_k converges to $\hat{f} < f^*$, $f_k < f^*$ holds with the exception of a finite number of iterations. The convergence of the algorithm will not be affected by those f_k values larger than the global minimum. The first iterations of our algorithm will process all of those subintervals on which the inclusion function $F(Y)$ may excessively overestimate the respective ranges, and thus $\underline{F}(Y) \leq f_k < f^*$ may occur. Later

$\underline{F}(Y) > f_k$ holds for each subintervals, and hence $p(\hat{f}, Y)$ will be necessarily negative.

Assume that the algorithm has an interval subsequence converging to a point $x \in X$. According to our earlier assumption, such a sequence must exist. Then due to the zero convergence property of the inclusion function $F(Y)$ the denominator of the $p(f_k, Y)$ parameter will converge to zero, and thus $p(f_k, Y)$ will converge to minus infinity. This is the reason why no subinterval Y' of the search region X may remain in the list L with a positive width: the value $p(f_k, Y')$ must become the largest in a certain iteration cycle, and thus it will be selected for further subdivision according to the interval selection rule, choosing the subinterval with the maximal $p(f_k, Y)$ value. □

Theorem 3. Assume that the inclusion function of the objective function is isotone and it has the zero convergence property. Consider the interval branch-and-bound optimization algorithm that uses the cut-off test, the monotonicity test, the interval Newton step and the concavity test as accelerating devices, and that selects as next leading interval that interval Y from the working list which has the maximal $p(f_i, Z)$ value. A necessary and sufficient condition for the convergence of this algorithm to a set of global minimizer points is that the sequence $\{f_i\}$ converges to the global minimum value f^* and there exist at most a finite number of f_i values below f^*.

Proof. 1. Consider first the sufficiency of the conditions. Assume that the conditions of the theorem are fulfilled, and still the sequence of leading intervals generated by the interval optimization algorithm has a subsequence, $\{X_i\}$ $(i = 1, 2, \ldots)$, that converges to a point x' for which $f(x') > f^*$, i.e. x' is not a global minimizer point.

Since $w(X_i) \to 0$, and $w(X_i) \to 0$ implies $w(F(X_i)) \to 0$, and $f_i \to f^*$, there exists such a positive integer k that $\underline{F}(X_i) > f_i$ holds for all $i > k$. In this way the value of $p(f_i, X)$ is negative for such i indices.

It is assumed that the search region X contains a global minimizer point that is also a stationary point. An interval containing such a stationary point can be subdivided, but cannot be deleted by any accelerating device, since

a, the lower bound $\underline{F}(Y)$ cannot be larger than $\tilde{f} \geq f^*$ (cutoff-test),

b, the objective function cannot be strictly monotonic on Y (monotonicity test),

c, the interval Newton step must give a result $N(Y)$ such that $N(Y) \cap Y \neq \emptyset$ due to the existence of a minimizer point in Y, and finally d, the objective function cannot be strictly concave on the whole interval Y (concavity test).

Since after subdivision one of the resulting subintervals contains the stationary point, there must exist at each iteration a subinterval Y containing a stationary point y^* such that $f(y^*) = f^*$. For this interval $\underline{F}(Y) \le f^*$ must hold due to the properties of the inclusion function, and hence we obtain $f^* - \underline{F}(Y) \ge 0$, i.e. a nonnegative $p(f_i, Y)$ value for f_i parameters not smaller than f^*. For $f_i < f^*$ the corresponding $p(f_i, Y)$ values can be negative.

Consequently an interval with a nonnegative $p(f_i, Y)$ value must be always on the work list (with the exceptions of at most a finite number of iteration cycles), and in each such iteration it must be compared with other intervals during the selection of the next leading interval. For some iterations when $j > k$, the algorithm preferred members of the $\{X_i\}$ interval sequence (having negative $p(f_j, X_j)$ values for $j > k$) compared to Y. Since $p(f_i, Y)$ is almost always nonnegative, this fact contradicts the selection rule choosing always the subinterval with the maximal $p(f_i, X)$ value.

This contradiction implies that the sequence of leading intervals generated by our algorithm cannot contain such a subsequence that would converge to a point having larger objective function value than f^*.

2. The necessity of the first condition (i.e. that the sequence $\{f_i\}$ must converge to the global minimum value f^*) was proved in Theorem 1. We prove here that it is necessary to have the convergence of the algorithm to the set of global minimizer points that at most a finite number of f_k values can be below f^*. Without this a counter example could be constructed.

Consider a problem for which $f(x') > f^* = 0$ for an $x' \in X$, and there exists a subinterval $Y \subseteq X$, for which $f(y) = f^* = 0$ holds for each $y \in Y$. It is obvious that the $p(f_i, Y_i')$ values will converge to minus infinity as the subintervals Y_i' converge to x', since the numerator of $p(f_i, Y_i')$ will remain negative for $i > k$, and the denominator converges to zero from above. Having a fixed sequence of $\{f_i\}$, the lower bounds $\underline{F}(Y_i)$ of the subintervals Y_i within Y can be set in such a way that the resulting $p(f_i, Y_i)$ values will be less than the $p(f_i, Y_i')$ value for the subinterval containing the x' point. This can be done even in accordance with the required inclusion isotonicity and zero convergence properties of the related inclusion function. □

As an immediate consequence of the statement of Theorem 3, the following corollaries can be stated:

Corollary 1. If our algorithm applies the interval selection rule of maximizing the $p(f^*, X) = pf^*(X)$ values for the members of the list L (i.e. if we can use the known exact global minimum value), then the algorithm converges exclusively to global minimizer points.

Corollary 2. If our algorithm applies the interval selection rule of maximizing the $p(\tilde{f}, X)$ values for the members of the list L where $\tilde{f}$ is the best available upper bound for the global minimum, *and its convergence to f^* can be ensured*, then the algorithm converges exclusively to global minimizer points.

Remark 1. Notice that maximizing the $p(f_i, X)$ value when selecting the next subinterval to be subdivided is similar to minimizing the $\underline{F}(X)$ value for the subintervals at hand. For constant denominator $\overline{F}(X) - \underline{F}(X)$ and for fixed $f_i = f^*$ values the same subinterval sequence will be selected by the two algorithm versions.

Remark 2. In real life situations, either the global minimum value is known at least approximately, or the updated best upper bound of it, the $\tilde{f}$ parameter can be used in the $p(\hat{f}, X)$ expression. The first case is the less probable, still there exist problems where the optimum value is a priori known, only the location of the global minimizer points is to be determined. Both approaches for $\hat{f}$ are realistic and allow convergence in the above strong sense.

Remark 3. Notice that Theorem 3 was proved without α-convergence assumed for the inclusion function, although the typical implementations (natural extension or higher order central forms) have this property. On the other hand, the zero convergence property was directly used in the proof.

The consequences of Theorems 1, 2 and 3 are that the user must either set $f_k = \hat{f}$ to be the known exact global minimum (when available), or use the updated best current upper limit of it provided by the algorithm itself: $f_k = \tilde{f}$. Larger set constants as $f_k = \hat{f} > f^*$ may cause convergence to not optimal points x' with function values $\hat{f} \geq f(x') > f^*$. When f_k are set to a constant smaller than the global minimum: $f_k = \hat{f} < f^*$, the algorithm will simply subdivide every subinterval, converging thus to all points within the search region X.

According to earlier numerical experiences (Casado et al., 2000a), but also according to other preliminary computational tests on the usability of the $pf^*(\hat{f}, X)$ parameter in the selection criterion, the new rule is used best together with the Moore-Skelboe selection criterion (Moore and Ratschek, 1988; Skelboe, 1974). In other words, that subinterval must be selected for the next iteration which has large $pf^*(f_i, X)$ value and nearly minimal $\underline{f}(X)$ lower bound. To combine the two criteria, a smooth utility function $u(x, y) : \mathbb{R}^2 \to \mathbb{R}$ will be applied, and that subinterval will be considered for the next iteration, for which $u(pf^*(f_i, X), -\underline{F}(X))$ is maximal. Theorem 4 characterizes the convergence properties of the studied interval branch-and-bound algorithm with such a compound selection criterion.

Theorem 4. Assume that the inclusion function of the objective function is isotone and it has the zero convergence property. Consider the interval branch-and-bound optimization algorithm that selects as next leading interval that interval Y from the working list which has the maximal $u(p(f_i, X), -\underline{F}(X))$ value.

1. Sufficient conditions for the convergence of this algorithm to a set of global minimizer points are that the sequence $\{f_i\}$ converges to the global minimum value f^*, there exist at most a finite number of f_i values below $f^*$and that the utility function $u(x, y)$ is strictly monotonically increasing in both of its arguments.

2. On the other hand, $f_i > f^* + \delta$ for a $\delta > 0$ allows convergence to a nonoptimal point even for the class of defined utility functions.

Proof. 1. Following the lines of thought of the proof of Theorem 3, assume that although the conditions of Theorem 4 are fulfilled, there exists a subsequence $\{X_i\}$ of the leading intervals that converges to a point $x' \in X$ for which $f(x') > f^*$ holds, i.e. x' is not a global minimizer point.

It is again assumed that the search region X contains a global minimizer point that is also a stationary point. A subinterval containing such a stationary point cannot be deleted according to the proof of Theorem 3. Thus after every iteration cycle, there must exist a subinterval Y_i that contains such a stationary point y with $f(y) = f^*$. For these subintervals $\underline{F}(Y_i) \leq f^*$ must hold due to the properties of the inclusion function, and hence we obtain $f^* - \underline{F}(Y_i) \geq 0$, i.e. a nonnegative $p(f_i, Y_i)$ value for f_i parameters not smaller than f^*. For a finite number of iteration cycles when $f_i < f^*$, the respective $p(f_i, Y_i)$ values can be negative. Thus after a sufficiently large number of iterations

$$u(p(f_i, Y_i), -\underline{F}(Y_i)) > u(p(f_i, Y'), -\underline{F}(Y'))$$

holds for the leading subinterval Y_i' containing x' due to the monotonicity property of u. However, this inequality does not allow selection of subintervals containing the supposed accumulation point x' .

Thus the sequence of leading intervals generated by our algorithm cannot contain such a subsequence that would converge to a point having larger objective function value than f^*.

2. Consider now the case when $f_i > f^* + \delta$ for a $\delta > 0$. Choose such a $x' \in X$, for which $f(x') = f^* + \delta$, and a utility function, for which an arbitrary large function value can be reached by a proper first argument (as e.g. $u(x,y) = ax + by$ with $a, b > 0$). Although an interval sequence converging to x' has smaller u values than that belonging to the global minimizer point, these values can be increased with the redefinition of the inclusion function as in the proof of Theorem 1. In this way, it can be achieved that the interval selection step will choose the interval sequence converging to x'. $\square$

The mild condition on the utility function allows u to be linear with positive coefficients: $u(x,y) = ax + by$ with $a, b > 0$, or e.g. to use $u(x,y) = xy$. With the first utility function we can weight the two underlying criteria: since $pf^*(Y)$ is scaled to have values between zero and one, $-\underline{F}(Y)$ should be divided e.g. by $w(F(X))$ to have a balanced decision. With this weighting our decision will be invariant against linear transformations of the objective function. Utility functions that are constants in one of their arguments do not fit Theorem 4, albeit these ensure convergence, since they realize simply one of the underlying interval selection criteria.

In an earlier paper (Casado et al., 2000a) the following theorem was proved for the $pf^*(X)$ parameter and for the traditional Moore-Skelboe algorithm variant:

Theorem 5. Assume that for an optimization problem $\min_{x \in X} f(x)$ the inclusion function $F(X)$ of $f(x)$ is isotone, and α-convergent with given positive constants α and C. Assume further that the pf^* parameter is less than 1 for all the subintervals of X. Then an arbitrary large number $N(> 0)$ of consecutive leading intervals of the basic B&B Algorithm that selects the subinterval with the smallest lower bound as next leading interval may have the properties that:

1. none of these processed intervals contains a stationary point, and
2. during this phase of the search the pf^* values are maximal for these intervals.

Although Theorem 5 investigates a different algorithm with another interval selection rule, its statement is still interesting for our problem: using the overestimation property typical for inclusion functions, one can construct problems that have an arbitrary long sequence of consecutive leading intervals with maximal $pf^*(X)$ values. In other words, it is possible to find problems for which the $pf^*(X)$ value can be diverted for an arbitrary long sequence of leading intervals.

2.2. Approximations of the global minimum

According to the last subsection, the availability of a sequence converging to the global minimum is indispensable, the present subsection is devoted to investigations how such sequences can be obtained. The common estimator of the global minimum is the updated upper bound of it:

$$\tilde{f}_k = \min\{\tilde{f}_{k-1}, \overline{F}(A_1), ..., \overline{F}(A_i)\}, \tag{2}$$

where $A_1, A_2, \ldots, A_i$ are the result subintervals after subdividing the leading interval A in the iteration cycle k, and $\tilde{f}_0 = \overline{F}(X)$. In the Moore-Skelboe algorithm $\tilde{f}_k$ converges to the global minimum from above (Ratschek and Rokne, 1988).

Another source of information on the global minimum value is the lower bound of the objective function on the leading interval. It can be formulated as

$$\underline{F}(A_k), \tag{3}$$

where A_k is again the leading interval in the iteration cycle number k. The Moore-Skelboe algorithm always selects a subinterval from the working list on which the objective function has the minimal lower bound, and these lower bounds converge to the global minimum (Ratschek and Rokne, 1988).

For the further investigation we call our optimization algorithm as defined in the introduction with the interval selection based purely on the $pf(f_i, X)$ parameter as *Algorithm 1*, and that with the interval selection according to the utility function u as *Algorithm 2*.

Assertion 1. The $\tilde{f}_k$ and the $\underline{F}(A_k)$ parameters as defined in (2) and (3), do not necessarily converge to the global minimum value f^* for Algorithm 1 and Algorithm 2.

Proof. As it was shown in the proofs of Theorems 1 and 4, Algorithms 1 and 2 may converge exclusively to nonoptimal points (to x' such that $f(x') > f(x^*)$). In these cases $\tilde{f}_k$ can converge to $f(x')$, or can remain constant (below $f(x')$) after a certain iteration index. On the other hand, $\underline{F}(A_k)$ converges exclusively to $f(x')$. Anyway, these parameters do not necessarily converge to the global minimum value. □

In spite of the negative results of Assertion 1, there remain plenty of ways to have an updated estimate of the global minimum value: we can use a random sampling of the search region, we can utilize the lower bounds of the intervals in the working list, and we can also run local search procedures to improve the available estimate of the global minimum value. Also a proper combination of the above techniques can give a converging estimate.

3. Numerical Tests

The numerical tests were carried out on a dual processor Pentium-II computer (233 Mhz, 128 Mbyte) under the Linux operating system. The programs were coded in C++. The inclusion functions were implemented via the PROFIL/BIAS routines (Knüppel, 1993), and the basis algorithm was that of the C++ Toolbox for Verified Computing (Hammer et al., 1995). The standard time unit (the CPU time required to evaluate the Shekel 5 test function 1000 times at $(4.0, 4.0, 4.0, 4.0)^T$) was 0.00209 seconds.

In our computational experiments we have used the following global optimization test problems: Shekel-5 (abbreviated as S5), Shekel-7 (S7) and Shekel-10 (S10), Hartman-3 (H3), Hartman-6 (H6), Goldstein-Price (GP), Six-Hump-Camel-Back (SHCB), Three-Hump-Camel-Back (THCB), Branin RCOS (BR), Rosenbrock (RB), 5-dimensional Rosenbrock (RB5), Levy-3 (L3), Levy-5 (L5), Levy-8 (L8) to Levy-16 (L16), Levy-18 (L18), Schwefel-2.1 (Sch21), Schwefel-3.1 (Sch31), Schwefel-2.5 (Sch25), Schwefel-2.7 (Sch27), Schwefel-2.14 (Sch214), Schwefel-2.18 (Sch218), Schwefel-3.2 (Sch32), Schwefel-3.7 (Sch37 in the 5 and 10 dimensional form), Griewank-5 (G5), Griewank-7 (G7), Ratz-4 to Ratz-8 (R4 to R8), and EX2 from (Csendes and Ratz, 1997). The search regions were the same as in other numerical tests (Csendes and Ratz, 1997; Hansen, 1992; Ratz and Csendes, 1995; Törn and Žilinskas, 1987). Although we used standard global optimization test problems, the test set is of course limited, and thus the computational results must be interpreted with care.

The goal of the computational test was to demonstrate the effect of, changing the old interval selection rule to the new one, selecting that subinterval which has the maximal pf^* parameter value. It was the only difference between the algorithm variants (denoted by old and new). The new algorithm utilized the known minimum value to determine a global minimizer point. The numerical testing of those algorithms variants which use certain $\hat{f}$ approximations of the f^* value remains for a later study.

For some test problems (e.g. L8, to L11) the minimum value was zero, and the minimum of the lower bounds of the inclusion functions was also zero due to the sum of squares function forms. In the case of these problems, the old algorithm will select only such subintervals for which $\underline{F}(Y) = 0$. For the same subintervals the respective pf^* values are also zero, and thus larger than the negative pf^* values of all other subintervals. Hence the two algorithm versions produce the same results in terms of the maximal list length necessary and of the number of function evaluations too. In these cases also the required CPU time was nearly the same.

3.1. Derivative free algorithms

First the basic algorithm with only the cut-off test was run in these two versions. For these procedures no derivative information was necessary. We used the traditional bisection and the subdivision was made along the coordinate direction with the longest edge. These simple algorithms were stopped when the diameter of a candidate interval was smaller than 0.01, or if the length of the working list reached 20,000. This memory limitation is far from the physical one, still above this level, a larger and larger part of the computation must be spent on administration in contrast to function evaluations. The numerical results are demonstrated in Table I.

The test problems can be classified into 3 groups according to Table I. The first group is of those problems for which the change in the interval selection does not cause any difference in the efficiency indicators of CPU time (CPUt), maximal list length reached (MLL) and number of function evaluations (NFE). The second group contains problems which could not be solved by the old method below MLL = 20,000. There was no such instance when the old technique would have allowed a solution with MLL $<$ 20,000, while MLL was 20,000 for the new one. In this sense, the new suggested method is certainly better. The third group contains the rest of the problems.

In the case of the second group, the efficiency indicators are not reliable, since the quality of the solutions can be quite different, depending on the phase in which the memory limitation canceled the search. Even though it is true, still we can confirm that in each case the quality of the solution provided by the new method was better, and thus the real efficiency improvements can be even better than what could be found in Table I. The implementation of the new interval selection rule resulted in more than 2 order of magnitude better CPU time for the test problem Goldstein-Price, much less memory complexity, and in 95% less function evaluations. The Six-Hump-Camel-Back problem was quite difficult for the old algorithm, but it could be solved just below the 20,000 memory limitation. Still the efficiency indicators for this problem are very similar to those of the second group.

For the first and third group, we have summarized the efficiency indicators in the last two lines of Table I. Thus altogether 1,600 seconds was necessary for the solution of these problems with the old, and 463 seconds with the new algorithm. This means a 71% improvement, i.e. so much less time is needed if a similar set of problems should be solved, and we decide to use the new algorithm. If we consider the average relative improvement on these problems (which emphasizes the easier problems), then the CPU time saving to be expected on a problem which is similar to those in the given set is about 14%.

The sum of the necessary number of list members cannot be identified easily as a physical quantity, since the problems are usually solved separately, still it can express the total memory complexity of the given set of problems. The summed up MLL numbers was 29,172 for the old, and 10,785 for the new algorithm. This means a 63% improvement. The average of the percentages for the single problems is 90% meaning a 10% improvement for the new method. It is worth mentioning that with only one exception the new algorithm always required less or the same computer memory as the old one.

The total sum of function evaluations was 267,211 for the old, and 98,007 for the new algorithm. This is 63% improvement, while the average of percentages is 85%, showing 15% relative improvement. The sum of function evaluations is one of the most important efficiency indicators, since the usual assumption is that the test problems are close to real life problems in terms of difficulty of solutions while the test problem functions are quickly computable.

Summarizing the numerical experiences with the simple algorithm variants, we can conclude that the new method is definitely much better than the old one at least on the given test problems. We have found for these instances that the more difficult a problem, the more improvement can be achieved by the new technique, while no significant worsening is to be anticipated for easier problems. We must stress that the improvements are obviously not uniform: results on a few test problems can explain the majority of improvements on the test set. It is not unusual that new techniques can improve simple algorithms. The following subsection shows the improvements caused by the new method on a sophisticated algorithm variant.

3.2. Algorithm variants with all accelerating devices

In the second part of the numerical tests we used several sophisticated accelerating devices and other algorithmic changes that were investigated recently (Markót et al., 2000). Now both algorithm variants used the cut-off test, the monotonicity test, the concavity test and also interval Newton steps. The interval arithmetic based inclusion functions are now improved (when possible) by centered forms. A multisection scheme with 3 subintervals and with the subdivision direction selection rule C (ibid.) was applied. The gradients and Hessians were generated by forward mode automatic differentiation. The algorithms were stopped when the diameter of a candidate interval was smaller than 10^{-8}. This time all test problems could be solved in the regular way, thus no additional stopping criterion had to be given on the working list length. The numerical results are demonstrated in Tables II-III.

This time all the test problems were solved by all algorithm variants (in spite of the much more difficult stopping criterion), so no grouping of the problems is necessary, the discussion of the results is simpler. The total CPU time necessary to solve the 40 test problems was 830 seconds for the old, and 334 seconds for the new algorithm, meaning a 60% improvement. The relative improvement expressed as the average of percentages was 14%. The difference in the absolute figures was basically caused by the results on the problem EX2, that proved to be the most difficult to solve.

The sum of the list lengths was 4,608 for the old, and 4,941 for the new method. That is, the old algorithm proved slightly (by 7%) better for once. The average of percentages indicates a one per cent improvement for the new method. Here again, it was the problem EX2 that explains the majority of the difference. The changes in the memory complexity are moderate.

The sum of function evaluations was 334,247 for the old, and 142,328 for the new method, meaning 57% improvement for the new technique. The relative improvement measured as the average of percentages was 15%, saying that, again, the improvements were larger on hard to solve problems. Here EX2 played also an important role, although the whole improvement cannot be explained only by this problem.

The sum of gradient evaluations was 209,421 for the old, and 89,072 for the new algorithm meaning an improvement of 57%. The relative improvement expressed as the average of percentages was 15% for the gradient evaluations. The similar figures for the Hessian evaluations were 22,151, and 9,456 (57% improvement), and 16% relative improvement. These improvement results are basically the same as the NFE figures, showing that these efficiency indicators are quite dependent.

The total number of iterations was 46,375 for the old, and 18,118 for the new algorithm with 61% improvement. The average of percentages gives a 15% improvement. The slight differences between the changes in the number of iterations and in the necessary CPU time are to be explained by the fact that the necessary computation is not simply proportional to the number of iterations, other factors such as larger list administration costs can bias the CPU times.

Summarizing the numerical results with the sophisticated algorithm variants, we can conclude that the new method proved to be advantageous on the test problems, although the improvements were smaller this time. Again, on harder to solve problems the improvements were larger, and the improvements are obviously not uniform: results on a few test problems can explain the majority of improvements on the test set. The set of test problems showed a balanced picture with many interesting details. The achieved improvements on the studied sophisticated algorithm are very valuable and surprisingly large. It must be also mentioned that when all the global minimizer points were to be located then the improvements were smaller.

4. Summary and conclusions

The interval branch-and-bound algorithm for global optimization is converging to the set of global minimizer points if the interval selection criterion is to choose that interval from the list L which has the maximal $p(\hat{f}, Y)$ parameter value. The user may set $\hat{f}$ to the global minimum value f^*, if it is a priori known, or to the iteratively updated lowest upper bound of the global minimum, $\tilde{f}$ (if additional conditions are fulfilled). The latter parameter is always available, and thus no previous information is needed to ensure convergence. Necessary and sufficient conditions of the convergence are given.

To improve the efficiency, the new interval selection criterion can be combined with the Moore-Skelboe criterion that chooses the subinterval with the minimal lower bound on the objective function. This algorithm converges to the set of global minimizer points under the mild condition that the applied utility function is strictly monotonically increasing in both of its arguments.

According to the present theoretical study, intervals with zero width inclusion functions must be handled with special care in the proposed algorithms, even though they are generally quite rare in real life unconstrained problems. The new algorithm class promises better efficiency utilizing the easy to obtain information of the relative place of the global minimum value within the inclusion interval of the objective function for leading intervals.

An extensive computational study was completed on 40 standard test problems [4], using both simple and sophisticated algorithm variants. According to the numerical investigations, the suggested algorithmic change improves all versions substantially. The more difficult the problems to be solved, the larger the improvements (at least on our problem set). The suggested new interval selection rules seem to be an indispensable part of future interval optimization algorithms.

Acknowledgements

The author is indebted to Mihály Csaba Markót who contributed to this paper by programming the related algorithms.

References

Casado, G. L. and I. García: New Load Balancing Criterion for Parallel Interval Global Optimization Algorithm, Proceedings of the 16th IASTED International Conference, 321–323, Garmisch-Partenkirchen, Germany, 1998.

Casado, L. G., I. García, and T. Csendes: A Heuristic Rejection Criterion in Interval Global Optimization Algorithms, submitted for publication.

Casado, L. G., I. García, and T. Csendes: A new multisection technique in interval methods for global optimization, Computing, accepted for publication.

Csendes, T. and D. Ratz: Subdivision direction selection in interval methods for global optimization. SIAM J. Numerical Analysis, 34(1997), 922-938.

Hammer, R. M. Hocks, U. Kulisch, D. Ratz: C++ Toolbox for Verified Computing. Springer, Berlin, 1995.

Hansen, E.: Global optimization using interval analysis. Marcel Decker, New, York, 1992.

Horst, R. and P. M. Pardalos (eds.): Handbook of Global Optimization. Kluwer, Dordrecht, 1995.

Kearfott, R. B.: Rigorous global search: continuous problems. Kluwer, Dordrecht, 1996.

Knüppel, O.: BIAS — Basic Interval Arithmetic Subroutines. Technical Report 93.3, University of Hamburg, 1993.

Markót, M.Cs., T. Csendes, A.E. Csallner: Multisection in Interval Branch-and-Bound Methods for Global Optimization II. Numerical Tests. J. Global Optimization 16(2000), 219-228.

Moore, R.E. and H. Ratschek: Inclusion functions and global optimization II. Mathematical Programming, 41(1988), 341–356.

Ratschek, H. and J. Rokne: New Computer Methods for Global Optimization. Ellis Horwood, Chichester, 1988.

Ratz, D. and T. Csendes: On the Selection of Subdivision Directions in Interval Branch-and-Bound Methods for Global Optimization, J. Global Optimization, 7(1995), 183–207.

Skelboe, S.: Computation of rational functions, BIT, 14(1974), 87–95.

Stateva, R. and S. Tsvetkov: A Diverse Approach for the Solution of the Isothermal Multiphase Flash Problem. Application to Vapor-Liquid-Liquid Systems. The Canadian J. of Chemical Engineering, 72(1994), 722–734.

Törn, A. and A. Žilinskas: Global Optimization. Springer-Verlag, Berlin, 1987.

Table I. Numerical results for the basic algorithm with only the cut-off test. The given efficiency indicators are the required CPU time in seconds (CPUt), the maximal list length necessary (MLL), and the number of objective function evaluations (NFE). The last two lines contain summarized figures for those problems which could be solved by both algorithms with MLL less than 20,000. Σ denotes the sum of the given efficiency indicator for the old and new algorithms, followed by the relative compound indicator for the new method compared to that of the old one as per cents. The last line contains the average of the percentages (AoP) for the respective columns.

Problem		CPUt			MLL			NFE		
name	dim.	old	new	%	old	new	%	old	new	%
S5	4	0.42	0.40	95	11	11	100	184	173	94
S7	4	1.37	1.05	77	45	40	89	434	330	76
S10	4	2.19	1.09	50	53	30	57	490	244	50
H3	3	965.08	24.89	3	20,000	2,383	12	95,723	11,930	12
H6	6	1,411.88	1,083.41	77	20,000	20,000	100	90,357	80,230	89
GP	2	2,639.30	14.00	0	20,000	2,145	11	179,045	8,579	5
SHCB	2	1,131.44	2.40	0	19,812	2,383	12	142,618	3,883	3
THCB	2	9.18	4.92	54	1,128	1,348	120	16,693	5,457	33
BR	2	0.10	0.03	30	18	11	61	415	118	28
RB	2	0.02	0.02	100	10	10	100	135	135	100
RB5	5	7.94	8.01	101	56	56	100	6,919	6,919	100
L3	2	1,279.42	0.36	0	20,000	72	0	101,919	376	0
L5	2	1,235.80	0.17	0	20,000	37	0	99,698	162	0
L8	3	0.07	0.07	100	8	8	100	83	83	100
L9	4	0.15	0.15	100	11	11	100	111	111	100
L10	5	0.29	0.29	100	14	14	100	139	139	100
L11	8	1.13	1.13	100	23	23	100	227	227	100
L12	10	2.20	2.18	100	29	29	100	283	283	100
L13	2	0.03	0.03	100	7	7	100	80	80	100
L14	3	0.08	0.08	100	10	10	100	120	120	100
L15	4	0.17	0.17	100	13	13	100	159	159	100
L16	5	0.28	0.28	100	16	16	100	183	183	100
L18	7	0.71	0.71	100	22	22	100	255	255	100
Sch21	2	0.31	0.31	100	44	44	100	927	927	100
Sch31	3	0.06	0.06	100	7	7	100	112	112	100
Sch25	2	0.05	0.05	100	7	7	100	223	223	100
Sch27	3	402.37	410.82	102	5,706	5,706	100	64,580	64,580	100
Sch214	4	0.65	0.66	102	81	81	100	1,283	1,283	100
Sch218	2	9.42	0.06	1	678	78	12	15,399	311	2
Sch32	3	0.07	0.07	100	12	12	100	198	198	100
Sch37	5	0.01	0.01	100	2	2	100	7	7	100
Sch37	10	0.01	0.01	100	2	2	100	7	7	100
G5	5	8.02	8.04	100	32	32	100	6,175	6,175	100
G7	7	0.79	0.79	100	43	43	100	349	349	100
R4	2	1.85	0.09	5	616	73	12	3,851	364	9
R5	3	0.37	0.38	103	68	68	100	483	483	100
R6	5	1.38	1.38	100	72	72	100	731	731	100
R7	7	5.04	5.10	101	202	202	100	1,363	1,363	100
R8	9	11.93	12.12	102	314	314	100	1,995	1,995	100
EX2	5	1,097.19	966.16	88	20,000	20,000	100	90,390	89,274	99
Σ		1,600.10	462.96	29	29,172	10,785	37	267,211	98,007	37
	AoP			86			90			85

Table II. Numerical results with all of the available accelerating devices. The given efficiency indicators are the required CPU time in seconds (CPUt), the maximal list length necessary (MLL), and the number of objective function evaluations (NFE). Σ denotes the sum of the given efficiency indicator for the old and new algorithms, followed by the relative compound indicator for the new method compared to that of the old one as per cents. The last line contains the average of the percentages (AoP) for the respective columns.

Problem		CPUt			MLL			NFE		
name	dim.	old	new	%	old	new	%	old	new	%
S5	4	0.27	0.28	104	10	10	100	144	144	100
S7	4	0.39	0.40	103	14	14	100	149	149	100
S10	4	0.58	0.52	90	16	16	100	158	141	89
H3	3	0.34	0.34	100	9	20	222	303	297	98
H6	6	6.13	6.27	102	78	65	83	1,681	1,725	102
GP	2	8.87	0.29	3	470	153	32	15,694	536	3
SHCB	2	0.20	0.03	15	49	22	44	734	122	16
THCB	2	0.07	0.06	86	16	20	125	347	291	83
BR	2	0.06	0.03	50	9	11	122	202	91	45
RB	2	0.06	0.05	83	11	11	100	316	278	88
RB5	5	4.05	4.09	101	72	72	100	4,234	4,234	100
L3	2	1.90	0.28	15	138	57	41	2,225	342	15
L5	2	0.62	0.63	102	31	32	103	676	681	100
L8	3	0.07	0.07	100	9	9	100	93	93	100
L9	4	0.14	0.14	100	13	13	100	122	122	100
L10	5	0.25	0.25	100	15	15	100	142	142	100
L11	8	0.89	0.90	101	28	28	100	214	214	100
L12	10	1.89	1.88	100	36	36	100	288	288	100
L13	2	0.03	0.03	100	9	9	100	86	86	100
L14	3	0.08	0.08	100	12	12	100	137	137	100
L15	4	0.14	0.15	107	19	19	100	155	155	100
L16	5	0.21	0.21	100	20	20	100	163	163	100
L18	7	0.56	0.56	100	26	26	100	235	235	100
Sch21	2	0.23	0.24	104	25	25	100	918	918	100
Sch31	3	0.05	0.04	80	6	6	100	12	112	100
Sch25	2	0.06	0.06	100	4	4	100	345	345	100
Sch27	3	6.60	6.63	100	236	236	100	3,597	3,597	100
Sch214	4	1.11	1.12	101	78	78	100	2,587	2,587	100
Sch218	2	0.05	0.00	0	7	4	57	290	26	8
Sch32	3	0.06	0.07	117	7	7	100	217	217	100
Sch37	5	0.17	0.18	106	32	32	100	374	374	100
Sch37	10	11.00	11.30	103	818	818	100	5,087	5,087	100
G5	5	6.37	6.42	101	32	32	100	5,869	5,869	100
G7	7	0.65	0.65	100	58	58	100	339	339	100
R4	2	0.23	0.04	17	40	31	77	1,053	201	19
R5	3	0.73	0.74	101	57	57	100	1,105	1,105	100
R6	5	2.70	2.71	100	30	30	100	1,658	1,658	100
R7	7	9.19	9.22	100	41	41	100	2,934	2,934	100
R8	9	26.00	26.00	100	59	59	100	5,035	5,035	100
EX2	4	729.00	251.00	34	1,968	2,733	138	274,229	101,555	37
Σ		829.76	333.96	40	4,608	4,941	107	334,247	142,328	43
	AoP			86			99			85

Table III. Numerical results with all of the available accelerating devices. The given efficiency indicators are the number of gradient evaluations (NGE), the number of Hessian evaluations (NHE), and the number of iterations (NIT). Σ denotes the sum of the given efficiency indicator for the old and new algorithms, followed by the relative compound indicator for the new method compared to that of the old one as per cents. The last line contains the average of the percentages (AoP) for the respective columns.

Problem		NGE			NHE			NIT		
name	dim.	old	new	%	old	new	%	old	new	%
S5	4	88	88	100	8	8	100	16	16	100
S7	4	86	86	100	8	8	100	18	18	100
S10	4	90	80	88	8	7	87	18	17	94
H3	3	193	190	98	23	23	100	39	38	97
H6	6	1,137	1,190	104	83	89	107	193	192	99
GP	2	9,195	231	2	623	1	0	2,256	76	3
SHCB	2	435	62	14	27	4	14	121	17	14
THCB	2	215	177	82	24	19	79	49	41	83
BR	2	123	47	38	14	4	29	29	12	41
RB	2	180	153	85	18	12	67	43	38	88
RB5	5	2,878	2,878	100	330	330	100	419	419	100
L3	2	1,326	172	12	105	9	8	289	47	16
L5	2	406	408	100	31	32	103	87	86	98
L8	3	57	57	100	6	6	100	11	11	100
L9	4	75	75	100	8	8	100	13	13	100
L10	5	88	88	100	9	9	100	15	15	100
L11	8	130	130	100	10	10	100	23	23	100
L12	10	179	179	100	13	13	100	30	30	100
L13	2	49	49	100	5	5	100	10	10	100
L14	3	79	79	100	8	8	100	15	15	100
L15	4	89	89	100	8	8	100	17	17	100
L16	5	90	90	100	7	7	100	19	19	100
L18	7	132	132	100	9	9	100	27	27	100
Sch21	2	582	582	100	54	54	100	113	113	100
Sch31	3	66	66	100	6	6	100	14	14	100
Sch25	2	207	207	100	28	28	100	50	50	100
Sch27	3	2,217	2,217	100	252	252	100	430	430	100
Sch214	4	1,599	1,599	100	161	161	100	307	307	100
Sch218	2	176	15	8	21	2	9	44	3	6
Sch32	3	126	126	100	13	13	100	28	28	100
Sch37	5	228	228	100	34	34	100	45	45	100
Sch37	10	2,685	2,685	100	202	202	100	696	696	100
G5	5	3,648	3,648	100	331	331	100	351	351	100
G7	7	175	175	100	9	9	100	40	40	100
R4	2	650	102	15	52	5	9	154	28	18
R5	3	750	750	100	72	72	100	107	107	100
R6	5	1,223	1,223	100	101	101	100	140	140	100
R7	7	2,295	2,295	100	169	169	100	204	204	100
R8	9	4,065	4,065	100	262	262	100	310	310	100
EX2	4	171,409	62,359	36	18,999	7,126	37	39,585	14,055	35
Σ		209,421	89,072	43	22,151	9,456	43	46,375	18,118	39
	AoP			85			84			85