Journal of Global Optimization **30:** 435–456, 2004.
© 2004 *Kluwer Academic Publishers. Printed in the Netherlands.*

A New Inclusion Function for Optimization: Kite – The One Dimensional Case*

TAMÁS VINKÓ[1], JEAN-LOUIS LAGOUANELLE[2] and TIBOR CSENDES[3]
[1]*Research Group on Artificial Intelligence of the Hungarian Academy of Sciences and University of Szeged, Szeged, Hungary (e-mail: tvinko@inf.u-szeged.hu)*
[2]*Lab. LIMA, Institut de Recherche en Informatique de Toulouse, France (e-mail: lagouane@irit.fr)*
[3]*University of Szeged, Institute of Informatics, Szeged, Hungary (e-mail: csendes@inf.u-szeged.hu)*

(Received: 9 November 2001; revised: 13 May 2003; accepted: 25 January 2004)

Abstract. In this paper the kite inclusion function is presented for branch-and-bound type interval global optimization using at least gradient information. The basic idea comes from the simultaneous usage of the centered forms and the linear boundary value forms. We will show that the new technique is not worse and usually considerably better than these two. The best choice for the center of the kite inclusion will be given. The isotonicity and at least quadratical convergence hold and there is a pruning effect of the kite which is derived from the construction of the inclusion, thus more function evaluations are not needed to use it. A numerical investigation on large standard multiextremal test functions has been done to show the performance.

Key words: centered form, inclusion function, interval global optimization, pruning.

1. Introduction

Interval global optimization algorithms based on branch-and-bound methods [8, 9, 19] provide guaranteed and reliable solutions for the problem

$$\min_{x \in X} f(x),$$

where the objective function $f\colon D \subseteq \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable and $X \subseteq D$ is the search box representing bound constraints for x. After studying some properties of the interval global optimization methods [4–6], the aim of this work is to improve the efficiency by a tighter interval inclusion function, in particular we deal with *lower bounds* of f. The quality of an enclosure method is important in the implementation of interval global optimization algorithms, because narrower inclusion of f may provide faster convergence.

In the following we denote real numbers by lower case letters $(a, b, \dots)$ and real bounded and closed intervals by capital letters $(X, Y, \dots)$. In general the lower and the upper bounds of an interval X are denoted by $\underline{X}$ and $\overline{X}$, respectively. For the sake of simplicity if we deal with only one interval, say Y, we will denote its bounds by a and b. The set of compact intervals is denoted by $\mathbb{I} := \{[a, b] \mid a \leqslant b;\ a, b \in \mathbb{R}\}$. If $X \subseteq \mathbb{R}$, then $\mathbb{I}(X) := \{Y \mid Y \in \mathbb{I}, Y \subseteq X\}$. The width

*This work has been supported by the grant OTKA T 034350.

and the midpoint of an interval X are denoted by $w(X)$ and $\mathrm{mid}(X)$, respectively. The range of the function f on X is denoted by $f(X)$. The restriction of f to the interval Y is indicated by $f|Y$.

DEFINITION 1. We call a function F: $\mathbb{I}(X) \to \mathbb{I}$ an *inclusion function* of f in X if $x \in Y$ implies $f(x) \in F(Y)$ for all $Y \in \mathbb{I}(X)$.

F' denotes an inclusion function of the derivative f' of f. $L := \min F'(X)$, $U := \max F'(X)$.

By interval arithmetic [1, 9] inclusion functions can be computed not only for given expressions but also for almost all functions specified by a finite algorithm. Applying automatic differentiation [7, 9, 15] we are able to compute inclusion functions for the derivatives without previous human interaction. In the following we assume $L < 0 < U$. If $U \leqslant 0$ or $L \geqslant 0$ then f is monotonic and $f(X) = [f(b), f(a)]$ or $f(X) = [f(a), f(b)]$.

The framework of the branch-and-bound type algorithms used in this paper is described in the following.

ALGORITHM 1. Branch-and-bound interval global optimization algorithm

```
Step 1. Let $X$ be the starting interval, $\mathcal{L}$ the working list and $\mathcal{Q}$ the final list.
        Compute $F(X)$, set $\mathcal{L} := \{X, \underline{F}(X)\}$, $\mathcal{Q} := \{\}$ and the guaranteed upper
        bound $\tilde{f} = \overline{F}(c)$  $(c \in X)$ for $f^* := \min_{x \in X} f(x)$.
Step 2. While $\mathcal{L}$ not empty do the following steps.
Step 3. Select an element $\{Y, \underline{F}(Y)\}$ from $\mathcal{L}$ and delete it from the working list.
        Divide $Y$ into two subsets $U_1 \cup U_2 = Y$ such that $\mathrm{int}(U_1) \cap \mathrm{int}(U_2) = \emptyset$,
        where 'int' denotes the interior of a set.
Step 4. Compute $F(U_i)$, apply accelerating tools to eliminate $U_i$ or some part
        of it and update $\tilde{f}$ if it is possible.
Step 5. If some criteria are fulfilled then $\mathcal{Q} = \mathcal{Q} + \{U_i, \underline{F}(U_i)\}$ else $\mathcal{L} = \mathcal{L} +$
        $\{U_i, \underline{F}(U_i)\}$. Go to Step 2.
```

It works with two lists of intervals, one for the candidate and another for the result intervals. In Step 4 some well-known accelerating tools (*midpoint test, monotonicity test, concavity test, interval Newton step, etc.*) are used to discard subintervals in which no minimizer may occur. A detailed description of these tools can be found in [7–9, 19]. The aim of this paper is to give a new inclusion function, better than the foregoing ones, and to use some new accelerating tools. These investigations yield faster convergence in global optimization algorithms.

2. Centered Forms and Their Improvements

Centered forms derived from mean-value theorems are used very often to compute interval enclosures of f [16, 18]. Namely, $f(x)=f(c)+f'(\xi)(x-c)$ holds with $c, x \in Y$ and $\xi \in [\min\{c,x\}, \max\{c,x\}]$. Therefore

$$f(x) \in F_{\mathrm{CF}}(Y,c) := f(c) + F'(Y)(Y-c). \tag{1}$$

Here, f is expanded w.r.t. every $x \in Y$, since $F'(Y)$ is an interval evaluation of the derivative of f over the entire interval Y. Note that (1) can also be computed using interval slopes [17, 20] instead of derivatives, which yields sometimes a better enclosure of $f(Y)$. Often the center c in (1) is chosen to be the midpoint of the interval Y. In the following subsection the optimal c is given.

2.1. OPTIMAL CENTERED FORMS

Consider the lower bound of $F_{\mathrm{CF}}(Y,c)$ given by (1). In Figure 1 it can be seen that for any $c \in [a,b]$ the two lines defined by the point $P(c, f(c))$ and the slopes L and U, respectively, give a lower bound for f on Y:

$$\min\{y_M(c), y_N(c)\} \leqslant \inf_{x \in Y} f(x),$$

where

$$y_M(c) := f(c) + U(a-c) \quad \text{and} \quad y_N(c) := f(c) + L(b-c).$$

From this we can compute the optimal c for the lower bound. In [2], it has been proved that the best choice for c is when $y_M(c) = y_N(c)$, i.e. the point $c^- \in Y = [a,b]$ which maximizes $\min\{y_M(c), y_N(c)\}$. The following lemma gives the formulae for this optimal case.

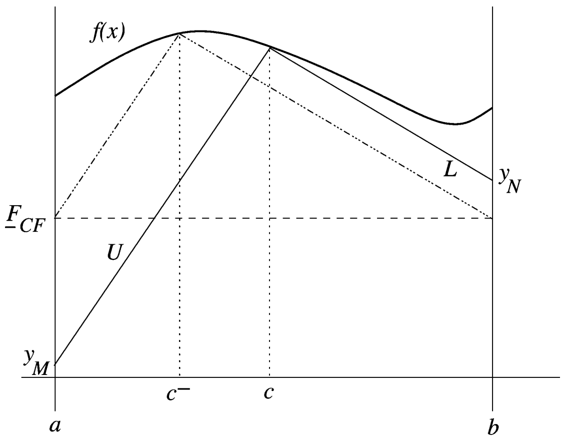

Figure 1. Centered form with the midpoint as centerpoint and with the optimal center.

LEMMA 1 (Baumann). *The optimal choice of c and the corresponding value for the lower bound is*

$$c^- = \frac{aU - bL}{U - L}$$

and

$$\underline{F}_{\mathrm{CF}}(Y, c^-) = f(c^-) + (b - a)\frac{LU}{U - L}. \tag{2}$$

Note that c^- is independent of the values of f and in global optimization algorithms the values L and U are usually available since they are computed for the monotonicity test. Hence the usage of Baumann centered form does not need extra function or gradient calls. A numerical investigation can be found in [21] for the one dimensional case.

With similar arguments one can easily obtain the corresponding formulae for the upper bound c^+ (it is the symmetric point of c^- with respect to $\mathrm{mid}(Y)$). Thus if one is interested in the best inclusion given by centered forms then both formulae have to be computed, and this increases the total computational effort for an optimization problem. However, in the interval global optimization algorithms the lower bound of the enclosure function F has a special role.

The usage of the Baumann centered form in the multidimensional case is also discussed in [2]. Note that this generalization can be done easily.

2.2. LINEAR BOUNDARY VALUE FORMS

When a centered form is applied to both lower and upper bounds a and b of the interval Y it is called a *linear boundary value form* [17]. This case is presented in Figure 2. The straightforward calculation of the intersection $S = (x_s, y_s)$ of the two lines $y = f(a) + L(x - a)$ and $y = f(b) + U(x - b)$ provides the formulae for the computation of a lower bound of $F(Y)$. This is claimed in the following lemma.

LEMMA 2 (Neumaier). *The two lines defined by the points $A(a, f(a))$ and $B(b, f(b))$ and the slopes L and U, respectively, provide a lower bound for f:*

$$x_s = \frac{f(a) - f(b)}{U - L} + \frac{bU - aL}{U - L}, \tag{3}$$

$$\underline{F}_{\mathrm{LBVF}}(Y) = y_s = \frac{Uf(a) - Lf(b)}{U - L} + (b - a)\frac{LU}{U - L}, \tag{4}$$

which is called the lower bound of the linear boundary value form.

It is clear that the inequality $\underline{F}_{\mathrm{LBVF}}(Y) \leqslant \underline{f}(Y)$ always holds because the two lines $y = f(a) + L(x - a)$ and $y = f(b) + U(x - b)$ are under the curve of the function f and they never cross.

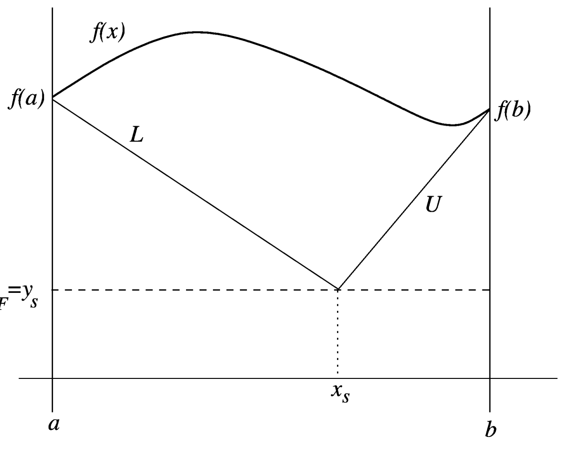

Figure 2. Linear boundary value form for the function $f|[a,b]$.

From these results the question arises: What is the better evaluation of the lower bound of F, the Baumann centered form or the linear boundary value form? The simple observation that the computation of (2) and (4) differs only by the expressions $f(c^-)$ and $(Uf(a)-Lf(b))/(U-L)$, provides the following statement.

PROPOSITION 1. $\underline{F}_{\mathrm{CF}}(Y,c^-)\leqslant\underline{F}_{\mathrm{LBVF}}(Y)$ *if and only if* $f(c^-)\leqslant\dfrac{Uf(a)-Lf(b)}{U-L}$.

As we can see, in some cases the linear boundary value form gives a better result than the Baumann centered form. Proposition 1 claims that it is the case, e.g. when the objective function f is convex.

Note that all of the values in the formulae (3) and (4) are fixed, there is no possibility to optimize the inclusion. The computation of $\underline{F}_{\mathrm{LBVF}}$ needs more information since $f(a)$ and $f(b)$ have to be computed, thus in a global optimization algorithm it may lead to a higher computational effort. However, when the function values have been computed at the extremal points of the current interval Y then these values can be used later, when subintervals of Y are considered. In Section 5 we use such a technique to reduce the computations. The formula for the upper bound of $f(x)$ is similar to (4) and it uses the previously computed values ($L,U,f(a)$ and $f(b)$). For the multidimensional case some investigations can be found in [13, 14].

3. The Kite Inclusion Function

3.1. SIMULTANEOUS USAGE OF CENTERED FORMS AND LINEAR BOUNDARY VALUE FORMS

Is there any advantage to use the two formulae simultaneously [12]? The answer is given in Figure 3 (here the centered form is not necessarily according to

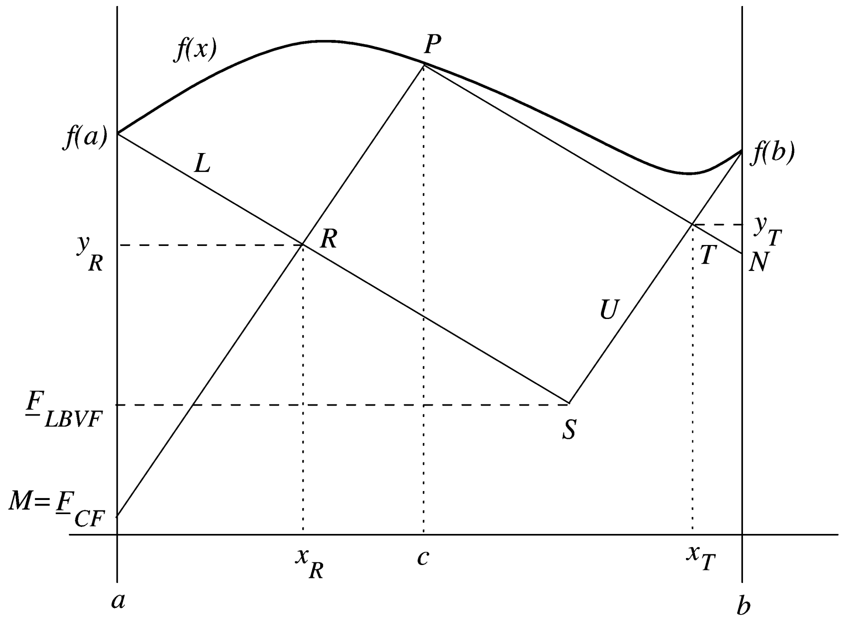

Figure 3. Simultaneous usage of the centered form (based on the midpoint of the current interval) and the linear boundary value form.

Baumann's suggestion, in this figure the midpoint of the current interval is used as the center), where the graph of f is within the convex inclusion cone determined by the points $(a, f(a)), S$ and $(b, f(b))$ and outside the concave exclusion cone MPN. This leads to the following assertion, which claims that the simultaneous usage provides a not worse (and usually considerably better) inclusion of the objective function.

PROPOSITION 2. *Let* $\underline{F}_K(Y,c) := \min\{y_R(c), y_T(c)\}$, *where*

$$y_R(c) := \frac{Uf(a) - Lf(c) + LU(c-a)}{U-L}, \tag{5}$$

and

$$y_T(c) := \frac{Uf(c) - Lf(b) + LU(b-c)}{U-L}. \tag{6}$$

Then the inequalities

$$\max\{\underline{F}_{\text{LBVF}}(Y), \underline{F}_{\text{CF}}(Y,c)\} \leqslant \underline{F}_K(Y,c) \leqslant \underline{f}(Y) \tag{7}$$

hold.

Proof. The point R is the intersection of the lines $y = f(a) + L(x-a)$ and $y = f(c) + U(x-c)$:

$$x_R(c) = \frac{f(a) - f(c) + Uc - La}{U-L} \tag{8}$$

and $y_R(c)$ is defined above in (5). The point T is the intersection of the lines $y=f(b)+U(x-b)$ and $y=f(c)+L(x-c)$:

$$x_T(c)=\frac{f(c)-f(b)+Ub-Lc}{U-L} \tag{9}$$

and $y_T(c)$ is defined above in (6).

In the following we have to consider four cases.

(i) Suppose that $\underline{F}_{\mathrm{CF}}(Y,c)\leqslant\underline{F}_{\mathrm{LBVF}}(Y)$. We have to prove that $\underline{F}_{\mathrm{LBVF}}(Y)\leqslant y_R(c)$, i.e.

$$\begin{aligned}\frac{Uf(a)-Lf(b)+LU(b-a)}{U-L} &\leqslant^? \frac{Uf(a)-Lf(c)+LU(c-a)}{U-L}\\ -Lf(b)+LUb &\leqslant^? -Lf(c)+LUc\\ L(f(c)-f(b)) &\leqslant^? -LU(b-c)\\ f(c)-f(b) &\geqslant^? U(c-b)\\ \frac{f(c)-f(b)}{c-b} &\leqslant U.\end{aligned}$$

The last inequality is always true, because its left hand side is the steepness of the line determined by the points $(c,f(c))$ and $(b,f(b))$, while U is an upper bound of $f'(x)$ on [a, b].

(ii) Now we prove that if $\underline{F}_{\mathrm{CF}}(Y,c)\leqslant\underline{F}_{\mathrm{LBVF}}(Y)$ then $\underline{F}_{\mathrm{LBVF}}(Y)\leqslant y_T$:

$$\begin{aligned}\frac{Uf(a)-Lf(b)+LU(b-a)}{U-L} &\leqslant^? \frac{Uf(c)-Lf(b)+LU(b-c)}{U-L}\\ Uf(a)-LUa &\leqslant^? Uf(c)-LUc\\ f(a)-f(c) &\leqslant^? L(a-c)\\ \frac{f(c)-f(a)}{c-a} &\geqslant L.\end{aligned}$$

The last inequality is always true, because its left hand side is the steepness of the line given by the points $(c,f(c))$ and $(a,f(a))$, while L is a lower bound of $f'(x)$ on $[a,b]$.

(iii) Suppose now that $\underline{F}_{\mathrm{LBVF}}(Y)\leqslant\underline{F}_{\mathrm{CF}}(Y,c)$. First $\underline{F}_{\mathrm{CF}}(Y,c)\leqslant y_R$ is checked. The proof of this case is similar to that of (i), one can show that $f(c)+U(a-c)\leqslant y_R(c)$ is satisfied for the same reason as in case (i).

(iv) Finally let us see the case: if $\underline{F}_{\mathrm{LBVF}}(Y)\leqslant\underline{F}_{\mathrm{CF}}(Y,c)$ then $\underline{F}_{\mathrm{CF}}(Y,c)\leqslant y_T(c)$. The proof of this case is similar to that of (ii), one can show that $f(c)+L(b-c)\leqslant y_T(c)$ is satisfied for the same reason as for the case (ii).

With these four considerations the inequality $\max\{\underline{F}_{\mathrm{LBVF}},\underline{F}_{\mathrm{CF}}\}\leqslant y_K$ has been proved.

Now we show that $y_K\leqslant\underline{f}(Y)$ also holds. Consider the intervals $Y_1=[a,c]$ and $Y_2=[c,b]$, where $c\in[a,b]$. The values y_R and y_T are the lower bounds of the linear boundary value form for the function f on the intervals Y_1 and Y_2, respectively. From Lemma 2 it is known that $y_R \leqslant \underline{f}(Y_1)$ and $y_T \leqslant \underline{f}(Y_2)$ hold. Consequently, the inequality $y_K = \min\{y_R, y_T\} \leqslant \underline{f}(X)$ also holds.

□

Notice that $\underline{F}_{\mathrm{CF}}(Y, c_1) \leqslant \underline{F}_K(Y, c_2)$ does not necessary hold if $c_1 \neq c_2$: for instance if $c_2 = a$ or $c_2 = b$ then $\underline{F}_K(Y, c_2) = \underline{F}_{\mathrm{LBVF}}(Y)$ and if $c_1 = c^-$ then by Proposition 1 $\underline{F}_{\mathrm{CF}}(Y, c^-)$ could be better than $\underline{F}_{\mathrm{LBVF}}(Y)$.

3.2. OPTIMAL CENTER OF THE KITE

From the above results it can be seen that the simultaneous usage of the Baumann centered form and the linear boundary value form gives an at least as good lower bound for the inclusion function as the better of them. Now – as for centered forms – we are interested in the best choice for the center c, that is a point c^* such that

$$\underline{F}_K(Y, c^*) = \max_{c \in [a,b]} \underline{F}_K(Y, c) = \max_{c \in [a,b]} \min\{y_R(c), y_T(c)\}. \tag{10}$$

In the following theorem we consider this optimal choice for the center of the kite.

THEOREM 1. *The following statements hold.*

1. *There is a unique* $c^* \in [a,b]$ *such that* $y_R(c^*) = y_T(c^*)$,
2. c^* *is a maximizer of* $\underline{F}_K(Y, c)$.

Proof. 1. We consider the difference $\Delta := y_T - y_R$. Applying derivation we obtain $y_R'(c) = \frac{-Lf'(c)}{U-L} + \frac{LU}{U-L} \leqslant 0$ and $y_T'(c) = \frac{Uf'(c)}{U-L} - \frac{LU}{U-L} \geqslant 0$ for all $c \in [a,b]$, which means that y_R monotonically decreasing and y_T monotonically increasing. Together with $L < 0 < U$ this implies $\Delta'(c) > 0$ for all $c \in [a,b]$. Hence Δ is strictly increasing. But since obviously $\Delta(a) \leqslant 0$ and $\Delta(b) \geqslant 0$, Δ has exactly one zero c^* in $[a,b]$, i.e.

$$\frac{Uf(a) - Lf(c^*) + (c^* - a)LU}{U - L} = \frac{Uf(c^*) - Lf(b) + (b - c^*)LU}{U - L}. \tag{11}$$

From Equation (11) we obtain

$$c^* = \frac{f(c^*) - f(a)}{2L} + \frac{f(c^*) - f(b)}{2U} + \frac{a+b}{2},$$

which means that c^* is the unique fixed point of the function

$$\varphi := \alpha f + \beta, \tag{12}$$

where

$$\alpha = \frac{L+U}{2LU}, \quad \beta = \frac{LU(a+b) - Uf(a) - Lf(b)}{2LU}.$$

2. We have seen that the function y_R monotonically decreasing while the function y_T monotonically increasing. Take a point d where $d \neq c^*$. If $d < c^*$ then

$$y_R(d) \geqslant y_R(c^*) = y_T(c^*) \geqslant y_T(d),$$

where one of the inequalities is strict because $d \neq c^*$. Thus,

$$\underline{F}_K(Y,d) = \min\{y_R(d), y_T(d)\} = y_T(d) \leqslant y_T(c^*) = y_R(c^*) = \underline{F}_K(Y,c^*)$$

holds. If $d > c^*$ then

$$y_R(d) \leqslant y_R(c^*) = y_T(c^*) \leqslant y_T(d),$$

where one of the inequalities is strict because $d \neq c^*$. Hence

$$\underline{F}_K(Y,d) = \min\{y_R(d), y_T(d)\} = y_R(d) \leqslant y_T(c^*) = y_R(c^*) = \underline{F}_K(Y,c^*)$$

holds. Now since for all $d \neq c^*$ the inequality $\underline{F}_K(Y,d) \leqslant \underline{F}_K(Y,c^*)$ holds, the maximum value of $\underline{F}_K$ is attained in c^*.

An illustrative example can be seen in Figure 4. □

It may happen that $\underline{F}_K(Y,\cdot)$ has more than one maximizer. If either $f'(c^*) = L$ or $f'(c^*) = U$ and $f'(d) = L$ or $f'(d) = U$ holds for all of the values $d \in [c^* - \varepsilon, c^* + \delta]$, $(\varepsilon, \delta > 0)$ then there can be infinitely many maximizers of the function $\underline{F}_K(Y,\cdot)$ in the interval $[c^* - \varepsilon, c^* + \delta]$. A simple example for this case can be seen in the Figure 5, where the function $\underline{F}_K(Y,\cdot)$ has an infinite number of maximizer points in the interval $[c^* - \varepsilon, c^*]$.

On the other hand if $f'(Y)$ lies in the open set (L,U), then there is one and only one optimal point. This is satisfied usually if L, U are computed by outward rounded interval arithmetic on a computer. Also, it is always possible to take $[L - \varepsilon, U + \varepsilon]$ with $\varepsilon > 0$ for enclosing the derivative.

COROLLARY 1. *The kite inclusion function with its optimal center always yields an at least as good lower bound as the Baumann centered form, i.e. the inequality*

$$\underline{F}_{\mathrm{CF}}(Y,c^-) \leqslant \underline{F}_K(Y,c^*)$$

always holds.

Proof. By Proposition 2 the inequality $\underline{F}_{\mathrm{CF}}(Y,c^-) \leqslant \underline{F}_K(Y,c^-)$ holds. We have seen that $\underline{F}_K(Y,c) \leqslant \underline{F}_K(Y,c^*)$ always holds, especially for $c = c^-$. □

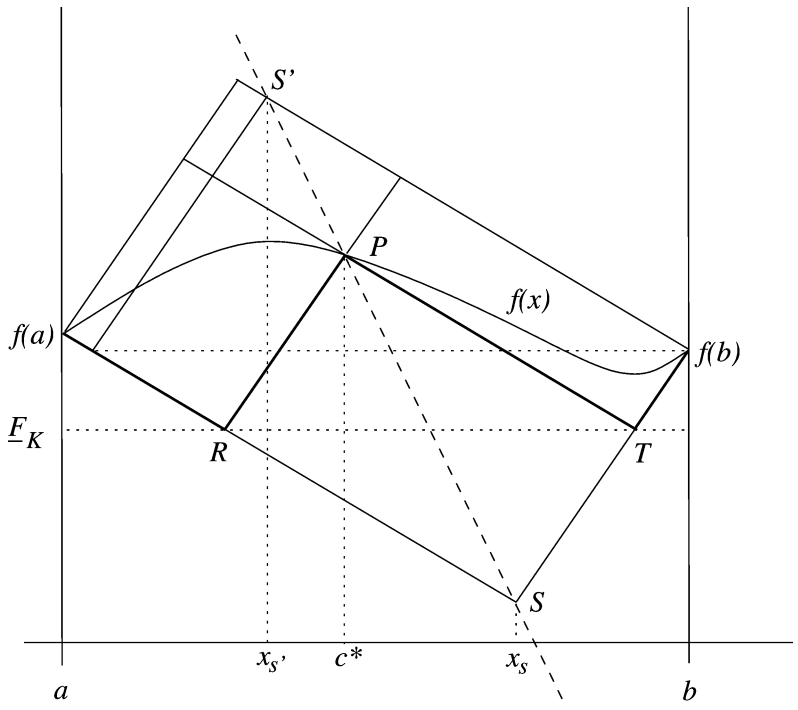

Figure 4. The optimal position c^* of c to obtain the best lower bound of $f|[a,b]$.

The information available in the current step can be used to obtain c^* by fixed point iterations in the interval given by the points x_s and $x_{s'}$, where x_s is given in (3) and

$$x_{s'} = \begin{cases} \dfrac{U(Ua-Lb)-L(f(a)-f(b))}{U(U-L)} & \text{if } f(a) \leqslant f(b), \quad \text{and} \\[2ex] \dfrac{L(Lb-Ua)-U(f(b)-f(a))}{L(L-U)} & \text{if } f(a) \geqslant f(b). \end{cases}$$

It is clear that c^* is included in the interval defined by x_s and $x_{s'}$, because the point $(c^*, f(c^*))$ is the crossing point of the graph of $f|[a,b]$ and the line determined by the points $(x_s, f(x_s))$ and $(x_{s'}, f(x_{s'}))$.

For faster convergence an interval Newton type method on the equation

$$\Phi(c) = \alpha f(c) + \beta - c = 0 \tag{13}$$

could be used. In this case the interval evaluation of $f'(c)$ has to be computed too. Moreover, one can apply a quasi Newton-method on the Equation (13) with the usage of the previously computed inclusion function of the derivative as a constant. In both methods usually one step is enough to provide a sufficiently good approximation of c^*. However, from Proposition 2 we know that the kite

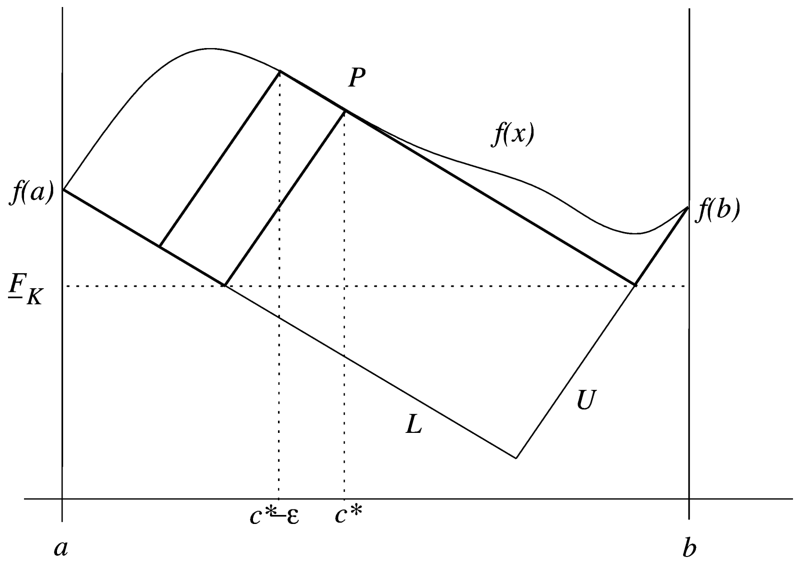

Figure 5. There can be many maximizers of the function $\underline{F}_K(Y,\cdot)$.

algorithm always yields an at least as good lower bound as the other two methods, thus for the application of this new method in global optimization algorithms the computation of c^* is not necessary with high precision, i.e. *any* $c \in Y$ could do, especially one from the interval determined by x_s and $x_{s'}$.

We call the procedure that computes an approximation c of c^* and an inclusion of $\underline{F}_K(Y,c)$ the *kite algorithm*. As we will see later, this procedure can be adopted into a global optimization algorithm and also used as an accelerating tool. We call this extension the extended kite algorithm. It is described in Subsection 3.5.

For an upper bound the corresponding center c' of the kite can be calculated from $y_{R'} = y_{T'}$, where R' is the crossing point of the lines $y = f(a) + U(x-a)$ and $y = f(c) + L(x-c)$, and T' is the crossing point of the lines $y = f(c) + U(x-c)$ and $y = f(b) + L(x-b)$. One can obtain the corresponding formulae:

$$x_{R'} = \frac{f(c)-f(a)+Ua-Lc}{U-L}, \quad y_{R'} = \frac{Uf(c)-Lf(a)+(a-c)LU}{U-L},$$
$$x_{T'} = \frac{f(b)-f(c)-Lb+Uc}{U-L}, \quad y_{T'} = \frac{Uf(b)-Lf(c)+(c-b)LU}{U-L}.$$

These formulae together with (5), (6), (8) and (9) give the lower and upper bounds of the inclusion function of $f(Y)$, respectively. Proposition 2 claims that the kite method provides an at least as good lower bound as the linear boundary value form and the Baumann centered form. A similar statement holds for the upper bound $\overline{F}_K(Y,c')$. Also Theorem 1 could be adapted to this case.

3.3. PROPERTIES OF THE KITE INCLUSION FUNCTION

In this subsection some properties of the kite inclusion function are considered.

DEFINITION 2. We call F an *inclusion isotone function* over X, if for all $Y,Z\in\mathbb{I}(X)$ $Y\subseteq Z$ implies $F(Y)\subseteq F(Z)$.

THEOREM 2. *Let us suppose that F' is an isotone function and let the inclusion function F of f be given by the kite algorithm, i.e. $F(Y)=[\underline{F}_K(Y,c^*),\overline{F}_K(Y,c')]$ for all $Y\in\mathbb{I}(X)$. Then F is an isotone function.*

Proof. Let $Y\subset Z=[a,b]$ and c_Z^* be a maximizer of the kite on Z. First, we have to prove that $\underline{F}_K(Y,c)\geqslant\underline{F}_K(Z,c_Z^*)$ holds for any $c\in Y$. Let $F'(Z)=[L,U]$ and $F'(Y)=[L',U']$. If $L'\geqslant 0$ then let $\underline{F}_K(Y,c):=[f(\underline{Y}),f(\overline{Y})]$ for all $c\in Y$ or if $U'\leqslant 0$ then let $\underline{F}_K(Y,c):=[f(\overline{Y}),f(\underline{Y})]$ for all $c\in Y$. In both cases $\underline{F}_K(Y,c)\geqslant\underline{F}_K(Z,c_Z^*)$ holds because the values $f(\underline{Y})$ and $f(\overline{Y})$ cannot be under the lines $y=f(a)+L(x-a)$ and $y=f(b)+U(x-b)$.

For the case $L'<0<U'$ an indirect proof is given. The idea comes from Figure 6, let the values L and U be given, i.e. the lines $y=f(a)+L(x-a)$ and $y=f(b)+U(x-b)$ are fixed. We try to construct such a function f which contradicts the isotonicity. In Figure 6 the dashed line signed by $\underline{F}(Z,c_Z^*)$ is the lower bound of the inclusion function given by the maximizer c_Z^* of the kite on Z. We try to construct an interval $Y\subset Z$ where the center of the kite c_Y is such a point $(c_Y,f(c_Y))$ which results in an inclusion function that contradicts the isotonicity. It is clear, that such a point should exist only in the quad $PTSR$, since only such points can produce a lower $\underline{F}$ value. However, it is a contradiction, because the slope of the line that passes through the points $(c_Y,f(c_Y))$ and $(c_Z^*,f(c_Z^*))$ is not in $[L,U]$. Consequently, there is no such a point c_Y which satisfies $\underline{F}_K(c_Y)<\underline{F}_K(c_Z^*)$.

The inequality $\overline{F}_K(c_Y)\leqslant\overline{F}_K(c_Z')$ can be proved with a similar argument, where c_Z' is a maximizer of the kite for upper bounds on Z and $c_Y\in Y$. □

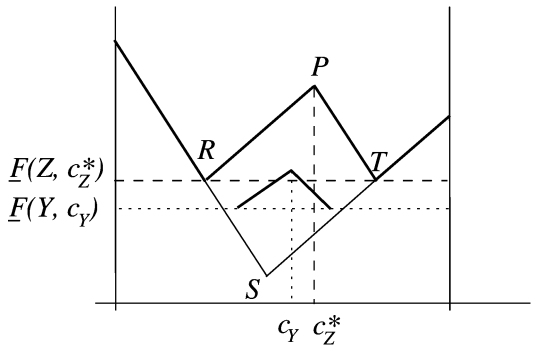

Figure 6. In this picture we try to construct such a function f which does not allow the isotonicity property of the kite.

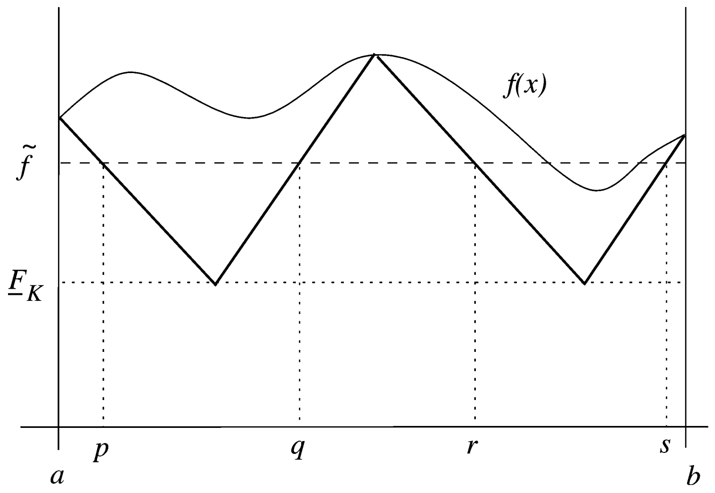

Figure 7. Pruning effect of the kite: the deletion of the intervals $[a,p)$, (q,r) and $(s,b]$ is possible (case (a) in Lemma 4).

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

LEMMA 3. *If the inclusion function of the derivative satisfies a Lipschitz condition, then the inclusion function given by the kite algorithm is α-convergent for at least $\alpha = 2$.*

Proof. From Proposition 2 we know that the kite algorithm always yields an at least as good lower bound for the inclusion function as the Baumann centered form. The same is true for the upper bound. It is also known that the centered forms are quadratically convergent in the case when $F'(X)$ satisfies a Lipschitz condition [11]. Consequently, the inclusion function given by the kite algorithm is at least quadratically convergent too. At least the same C and α values can be used for F_K as for F_{CF}. □

3.4. PRUNING EFFECTS OF THE KITE

In branch and bound based interval global optimization methods there are several well-known accelerating tools – for example the midpoint test and the monotonicity test. The aim of these tests is to delete an as large as possible part of the search space which does not contain a global optimizer point. In [20] a pruning technique based on slopes and in [21] – in geometrical sense a similar one – for derivatives have been developed. In this subsection we show a pruning technique based on the new kite inclusion function.

LEMMA 4. *Let $Y = [a,b] \subseteq X$ be the current considered subinterval, $c^* \in [a,b]$ be a maximizer of $\underline{F}_K(Y,\cdot)$, and $\tilde{f}$ be the current guaranteed upper bound for* *the global minimum value of* $f|[a,b]$. *Let us define the following values*:

$$p=a+\frac{\tilde{f}-f(a)}{L},\quad q=c^*+\frac{\tilde{f}-f(c^*)}{U},$$
$$r=c^*+\frac{\tilde{f}-f(c^*)}{L},\quad s=b+\frac{\tilde{f}-f(b)}{U}.$$

If $L<0<U$, *then a pruning technique based on the kite algorithm can be used in the sense that*

(a) *If* $\tilde{f}<\min\{f(a),f(b),f(c^*)\}$ *then all the global minimizer points of* Y *are contained in the intervals* $[p,q]$ *and* $[r,s]$.
(b) *If* $f(b)\leqslant\tilde{f}<\min\{f(a),f(c^*)\}$ *then all the global minimizer points of* Y *are contained in the intervals* $[p,q]$ *and* $[r,b]$.
(c) *If* $f(a)\leqslant\tilde{f}<\min\{f(b),f(c^*)\}$ *then all the global minimizer points of* Y *are contained in the intervals* $[a,q]$ *and* $[r,s]$.
(d) *If* $f(c^*)\leqslant\tilde{f}<\min\{f(a),f(b)\}$ *then all the global minimizer points of* Y *are contained in the interval* $[p,s]$.
(e) *If* $\max\{f(b),f(c^*)\}\leqslant\tilde{f}<f(a)$ *then all the global minimizer points of* Y *are contained in the interval* $[p,b]$.
(f) *If* $\max\{f(a),f(c^*)\}\leqslant\tilde{f}<f(b)$ *then all the global minimizer points of* Y *are contained in the interval* $[a,s]$.
(g) *If* $\max\{f(a),f(b)\}\leqslant\tilde{f}<f(c^*)$ *then all the global minimizer points of* Y *are contained in the intervals* $[a,q]$ *and* $[r,b]$.

Proof. (a) Let $z\in[a,b]$ such that $f(z)=\min_{x\in[a,b]}f(x)$ and $\tilde{f}\geqslant f(z)$. We have to prove that

$$a+\frac{\tilde{f}-f(a)}{L}\leqslant z. \tag{14}$$

We know that for all $x\in[a,b]$ the inequality $f(a)+L(x-a)\leqslant f(x)$ holds. If $x=z$, then $L(z-a)\leqslant f(z)-f(a)$ which is equivalent to

$$L\leqslant\frac{f(z)-f(a)}{z-a}\leqslant\frac{\tilde{f}-f(a)}{z-a}, \tag{15}$$

if $z\neq a$. If $z=a$ then (14) is true since $\tilde{f}<f(a)$. From (15) since $z>a$ we obtain

$$Lz\leqslant La+\tilde{f}-f(a)$$

which proves (14) because $L<0$. To prove that

$$z\leqslant c^*+\frac{\tilde{f}-f(c^*)}{U} \tag{16}$$

we use the inequality $f(c^*)+U(x-c^*)\leqslant f(x)$ which holds for all $x\in[a,c^*]$, $c^*\in[a,b]$. If $x=z$ then

$$U\leqslant\frac{f(z)-f(c^*)}{z-c^*}\leqslant\frac{\tilde{f}-f(c^*)}{z-c^*},$$

in the case when $z\neq c^*$. From this inequality we have

$$Uz\leqslant Uc^*+\tilde{f}-f(c^*),$$

which proves (16) because $U>0$. The case $z=c^*$ is not possible because $\tilde{f}<f(c^*)$ was supposed and $f(z)\leqslant\tilde{f}$.
The proof of the case $x^*\in[r,s]$ can be given with similar arguments.
(b)–(g) These cases can be proven in a similar way. □

Note that the usage of the pruning does not need extra information, all the values applied in the formulae of Lemma 4 have been computed previously. In Section 4 the efficiency of this pruning technique is demonstrated by several numerical tests.

EXAMPLE. We demonstrate the above considerations on a simple function. Let $f(x)=x^2-x, X=[0,0.75]$. Then the global minimum is $f^*=-0.25$ for $x^*=0.5$. By automatic differentiation (but also by hand calculation) one can obtain $F'(X)=[-1,0.5]$. Using the formulae above we get the following lower bounds for $F(X)$:

$$\underline{F}_{\mathrm{CF}}(c^-)=-0.5$$
$$\underline{F}_{\mathrm{LBVF}}=-0.375$$
$$\underline{F}_K(c^*)=-0.31066,$$

where c^* is approximately 0.43934. Thus the kite algorithm with the optimal c^* provides the best lower bound for $F(X)$.

Using the pruning effect of the kite, the subintervals $X_1=[0,0.25)$ and $X_2=(0.63,0.75]$ can be eliminated, only the subinterval $X'=[0.25,0.63]$ have to be considered further.

3.5. EXTENDED KITE ALGORITHM

This subsection describes the extended kite algorithm in detail, which can easily be built into an interval branch-and-bound optimization algorithm using at least first order information of the objective function f. In our implementation the algorithm described in Section 1 was used.

ALGORITHM 2. Extended kite algorithm

Step i. Compute $f(a), f(b)$, and $F'(X) = [L, U]$.

Step ii. If $L < 0 < U$ then using the computed values determine c^* (usually approximately). Then evaluate $f(c^*)$ and compute $\underline{F}_K(c^*)$.

Step iii. If $\tilde{f} > \min\{\overline{F}(c^*), \overline{F}(a), \overline{F}(b)\}$ then update $\tilde{f}$. Apply the midpoint test with $\overline{F}(c^*)$.

Step iv. Use the kite pruning technique with the intervals to be deleted in Lemma 4.

The extended kite algorithm can be used not only to get a better lower bound for F, but – as we could see in the previous subsection – for pruning and also in the midpoint test. Thus Algorithm 2 should be inserted at Step 4 into Algorithm 1.

In this detailed algorithm Step ii may have large computational complexity – depending on what kind of method is used for determining c^*. For the implementation of this algorithm in a global optimization method the optimal center of the kite is approximated by the mean value of the interval given by $x_{s'}$ and x_s.

4. Numerical Tests

This section presents numerical tests based on the above results. The test problems are given in [3], the computations were carried out on a dual processor Pentium-II computer (233 MHz, 256 Mbyte) under Linux operating system. The C++ Toolbox for Verified Computing [7] and C-XSC [10] was used as a computational environment.

In the implementation it was an important aspect that the construction of the optimal kite needs much information, thus we should not do computations redundantly. Our considerations were the following:

- The optimal center c^* can be well approximated as the mean value of x_s and $x_{s'}$ instead of applying the interval Newton method to compute the root of the equation (13). By this technique the computational effort decreases because the intervals $F(c)$ and $F'(c)$ do not have to be computed. The center given by this method is a good approximation of the optimal center c^*, especially when the considered subinterval is narrow.
- The pruning step is applied only at the end of the branching iteration because it moves the endpoint(s) of the current interval and then the new function value(s) of the new endpoint(s) would have to be computed again. It remained still to be cleared whether it is the most efficient way of the implementation.

The next subsections discuss the numerical results for two kinds of implementations. The first one uses only gradient information of f, the other one uses also second derivative information. Both algorithms were stopped when the width of the interval given by the lower bound of the kite inclusion and the current upper bound for f^* or the width of the candidate interval was smaller than a preset ε, i.e.

$$w([\underline{F}_K,\tilde{f}])\leqslant\varepsilon \quad \text{or} \quad w(Y)\leqslant\varepsilon.$$

In the comparisons the standard algorithm from [7] was used, which uses centered forms for inclusion functions of f. To measure the performance we use the following indicator: the number of function evaluations $+2\times$(the number of the derivative evaluations) $+3\times$(the number of the second derivative evaluations). Our experiments show that this is a correct weighting.

For the numerical tests the total CPU time used was below one second, thus the comparison of the computation time does not make much sense.

4.1. FIRST ORDER ALGORITHMS

First the basic algorithm with the cut-off and monotonicity test was run using centered forms and the kite algorithm with and without pruning technique. These variants use only first order information about the objective function. The numerical results for these variants that are obtained with $\varepsilon=10^{-12}$ are summarized in Table 1. All of the given 40 standard test problems were solved. For each test function the number of function evaluations (F-eval), the number of derivative evaluations (D-eval), the number of bisections (bisection) and the maximum list length necessary (list length) are reported. These indicators are given for all the three algorithm variants. At the end of the table the sum of the given indicators and the relative compound indicators for the new methods compared to that of the basic one are represented as percents.

The sum of function evaluations was 15,706 for the traditional method and 8,416 and 11,710 for the kite algorithm with and without pruning, meaning 46% and 26% improvement, respectively. The improvement in the sum of the function evaluations is mainly produced by hard problems, particularly by the last two, when the number of function evaluations were reduced by 38% and 37%, respectively. It can also be seen that the use of the pruning technique is beneficial.

The sum of derivative evaluations was 9,646 for the old method and 3,406 and 5,536 for the kite algorithm with and without using pruning meaning 65% and 43% improvement, respectively.

The performance measure is 34,998 for the old method and 15,228 and 22,728 for the kite algorithm with and without pruning. This means 56% and 35% improvement. From these indicators we can conclude that the use of the new enclosure method together with the pruning technique is important.

The sum of the number of bisections was 3,114 for the original method, 1,683 and 2,728 for the new procedure with and without pruning meaning 46% and 13% improvement, respectively. This together with the efficiency measure of the

Table 1. Numerical results for the first order algorithms on 40 one-dimensional problems.

Prob.	F-eval.			D-eval.			Bisection			List length		
no.	*cf*	$k+pr$	*k*	*cf*	$k+pr$	*k*	*cf*	$k+pr$	*k*	*cf*	$k+pr$	*k*
1	84	60	93	53	25	44	25	12	21	4	5	3
2	88	77	99	55	31	46	26	15	22	5	6	5
3	98	93	100	55	39	48	25	19	23	3	5	2
4	96	95	119	59	35	56	27	17	27	4	7	4
5	109	100	141	69	45	68	33	22	33	3	3	2
6	88	69	110	55	27	52	25	13	25	3	4	3
7	79	57	95	51	25	46	23	12	22	2	2	2
8	92	80	115	59	37	56	27	18	27	2	6	2
9	94	84	118	61	37	58	28	18	28	2	4	2
10	89	76	113	57	33	54	27	16	26	2	3	2
11	83	70	101	53	29	48	24	14	23	3	2	1
12	83	68	103	53	29	50	24	14	24	2	3	2
13	91	71	111	59	33	54	27	16	26	3	4	3
14	118	95	138	77	41	66	36	20	32	2	3	2
15	107	85	132	69	33	66	32	16	32	6	13	6
16	113	107	138	73	49	70	35	24	34	8	12	8
17	109	107	128	71	47	62	33	23	30	2	6	3
18	153	105	117	99	49	56	47	24	27	4	4	3
19	95	72	93	59	31	44	27	15	21	4	4	3
20	82	52	79	53	23	38	24	11	18	1	3	1
21	83	64	79	53	29	38	25	14	18	2	2	1
22	145	125	147	93	57	68	43	28	33	4	6	3
23	161	144	167	103	63	78	47	31	38	3	5	3
24	158	136	166	103	65	76	47	32	37	3	6	3
25	155	128	192	101	59	94	47	29	46	4	6	3
26	223	110	202	145	51	98	69	25	48	4	5	3
27	179	143	220	117	69	108	55	34	53	4	6	4
28	229	122	226	149	55	110	69	27	54	4	5	3
29	215	156	209	139	67	92	63	33	45	4	5	4
30	310	212	302	203	101	148	93	50	73	4	8	4
31	88	75	99	57	33	48	26	16	23	2	3	2
32	602	251	386	395	119	188	186	59	93	8	15	8
33	345	272	401	225	131	198	107	65	98	17	18	15
34	292	216	242	189	101	102	86	50	50	8	7	5
35	88	74	87	57	33	42	26	16	20	1	2	1
36	762	559	529	383	197	212	177	98	105	14	14	9
37	352	289	291	217	117	122	101	58	60	10	13	7
38	1,026	567	530	675	253	212	201	126	105	12	14	4
39	3,965	1,519	1,409	2,329	511	564	436	255	281	109	31	19
40	4,377	1,631	3,583	2,673	597	1,856	635	298	927	97	40	77
Σ	15,706	8,416	11,710	9,646	3,406	5,536	3,114	1,683	2,728	379	310	237
		54%	74%		35%	57%		54%	87%		82%	63%

function and the derivative evaluations indicate that the pruning has an important role in the usage of the kite enclosure method.

The sum of the maximal list lengths was 379 for the old method, 310 and 237 for the new algorithm with and without pruning, meaning 18% and 37% improvement. On the other hand, the memory complexity was very small.

From these results we can conclude that if the derivative of the objective function f is available, then the new algorithm using the kite inclusion function together with pruning is much better than the old one using only centered forms. Our numerical investigation shows that all the standard test problems can be solved with less computational effort.

4.2. THE SECOND ORDER ALGORITHM

In this subsection the numerical results for second order algorithms with cut-off test, monotonicity test, concavity test and interval Newton step are presented. The numerical results are demonstrated in Table 2, where the indicators are again the number of function (F-eval), derivative (D-eval), and second derivative (H-eval) evaluations, the number of bisections (bisection) and the maximal list length (list length).

These algorithm variants are more sophisticated, use all the well-known accelerating tools, thus we cannot expect so much improvements as in the previous subsection, because the more powerful methods we use, the less chance we have to improve their capabilities. In the implementation after bisecting the candidate interval the natural interval inclusion is used. After completing the monotonicity test, cut-off test and concavity test, also a Newton step is applied. The kite inclusion function together with its pruning procedure is used only on the subinterval(s) given by the Newton step. Our experience shows that only the (d), (e) and (f) pruning steps should be used, because these steps produce only one subinterval and this is suitable for this algorithm variant. This order of the accelerating techniques allowed a reduced computation effort.

For the second order algorithms the number of the function evaluations was $4,029$ for the original one, $3,101$ and $3,401$ for the new method with and without pruning, which means 23% and 16% improvements, respectively.

The sum of derivative evaluations was $2,747$ for the old algorithm, $1,659$ and $1,899$ for the new one with and without pruning. This means 40% and 31% improvement in the efficiency, respectively.

The sum of the second derivative evaluations was 612 for the old algorithm, 638 and 733 for the new one with and without pruning, which means that the performance of the kite algorithm was worse than the original method – according to this indicator. However, for every test function the number of evaluations of second derivatives was relatively small compared to the number of function and derivative evaluations, so it did not corrupt the performance too much.

The performance measure is $11,359$ for the traditional algorithm, $8,333$ and $9,398$ for the new one with and without pruning, which means 27% and 17%

Table 2. Numerical results for the second order algorithms on 40 one-dimensional problems.

Prob.	F-eval.			D-eval.			H-eval.			Bisection			List length		
no.	*cf*	*k*+*pr*	*k*	*cf*	*k*+*pr*	*k*	*cf*	*k*+*pr*	*k*	*cf*	*k*+*p*	*k*	*cf*	*k*+*pr*	*k*
1	60	38	48	40	18	25	8	6	9	6	3	4	4	3	3
2	74	49	56	48	23	30	10	10	11	6	5	5	4	4	4
3	66	49	52	45	26	29	11	12	13	7	6	6	1	1	1
4	67	68	71	44	30	33	9	10	11	6	5	5	8	4	4
5	95	74	84	67	42	49	13	16	19	10	8	9	3	2	2
6	58	38	52	38	18	29	8	8	11	5	4	5	4	1	1
7	43	35	38	29	18	21	6	8	9	5	4	4	2	2	2
8	51	35	39	34	18	22	7	8	9	5	4	4	2	1	1
9	52	43	46	35	23	26	7	10	11	6	5	5	2	2	2
10	55	42	53	37	19	27	8	8	11	6	4	5	2	3	3
11	44	35	38	29	18	21	6	8	9	5	4	4	2	1	1
12	45	47	43	30	21	20	6	8	7	5	4	3	3	1	1
13	59	50	53	40	27	30	8	12	13	7	6	6	3	3	3
14	57	46	60	40	26	34	8	10	15	7	5	7	2	2	2
15	106	84	94	69	43	50	14	14	17	9	7	8	8	6	6
16	118	100	103	77	53	56	15	16	17	11	8	8	8	5	5
17	54	43	46	37	23	26	9	10	11	2	5	5	2	2	2
18	84	51	70	58	25	38	13	10	16	9	5	7	3	2	2
19	74	40	50	48	20	27	10	8	11	6	4	5	5	2	2
20	43	36	46	30	19	26	6	8	11	5	4	5	1	1	1
21	50	36	39	34	19	22	7	8	9	5	4	4	2	1	1
22	77	58	61	54	35	38	13	12	13	7	6	6	1	1	1
23	92	75	76	62	40	44	15	14	15	8	7	7	6	4	4
24	73	55	58	50	32	35	12	12	13	7	6	6	1	1	1
25	92	66	80	64	34	45	14	14	18	10	7	8	2	2	2
26	113	92	99	80	48	55	17	20	23	13	10	10	3	3	3
27	92	62	82	64	33	47	15	14	20	10	7	9	3	3	3
28	103	86	93	72	45	52	16	20	23	12	10	10	3	3	3
29	53	43	46	35	23	26	5	8	9	5	4	4	2	1	1
30	144	121	130	103	65	74	22	28	32	17	14	14	5	4	4
31	51	44	47	36	24	27	6	10	11	6	5	5	2	1	1
32	275	158	174	195	84	100	43	32	40	31	16	16	8	7	7
33	362	229	243	243	128	139	48	56	61	33	28	29	17	16	16
34	116	90	96	81	52	55	19	18	19	12	9	9	5	3	3
35	51	44	54	36	24	31	7	10	13	6	5	6	1	1	1
36	207	191	186	140	108	106	32	36	35	20	18	17	9	7	7
37	120	102	105	79	49	52	18	16	17	12	8	8	5	3	3
38	162	118	143	111	71	84	30	26	32	15	13	15	6	4	4
39	273	218	238	188	123	137	50	44	50	25	22	24	8	9	9
40	218	210	209	145	112	111	36	40	39	20	20	19	10	9	9
Σ	4,029	3,101	3,401	2,747	1,659	1,899	612	638	733	406	319	336	168	131	131
		77%	84%		60%	69%		104%	120%		79%	83%		78%	78%

improvement, respectively. These indicators emphasize again the advantage of the kite method together with its pruning effect.

The sum of the number of bisections was 406 for the original method, 319 and 336 for the new procedure with and without pruning meaning 21% and 17% improvement, respectively.

The sum of the maximal list length was 168 for the old method, and 131 for both new algorithms, meaning 22% improvement.

Concluding the results we have obtained from the numerical investigations, the use of the kite inclusion function in the second order algorithms provides a better performance. Although this improvement is not as high as in the first order case, the usage of the new inclusion technique is still recommended.

5. Summary

The interval function enclosure method proposed here is derived from the centered form and the linear boundary value form. We have proved that the kite inclusion function is at least as good as the two older methods and there is a unique optimal case for the selection of the parameter c. The new method can easily be implemented in a branch-and-bound type interval global optimization algorithm. For a single inclusion larger computation effort is needed by the kite algorithm, thus an extended numerical study was reported on the performance. We can conclude that a 27–56% performance improvement can be achieved depending on what kind of information is used in the optimization algorithm.

Acknowledgement

The authors would like to thank the anonymous referees for their careful work which resulted in a better algorithm and in sharper theoretical results.

References

1. Alefeld, G. and Herzberger, J. (1983), *Introduction to Interval Computations*, Academic Press, New York.
2. Baumann, E. (1988), Optimal centered forms, *BIT*, 28(1), 80–87.
3. Casado, L.G., García, I., Martínez, J.A. and Sergeyev, Ya.D. (2003), New interval analysis support functions using gradient information in a global minimization algorithm, *J. Global Optimization*, 25(4), 345–362.
4. Csallner, A.E., Csendes, T. and Markót, M.Cs. (2000), Multisection in interval branch-and-bound methods for global optimization – I. Theoretical results, *J. Global Optimization*, 16(4), 371–392.
5. Csendes, T. (2001), New subinterval selection criteria for interval global optimization, *J. Global Optimization*, 19(3), 307–327.
6. Csendes, T. and Ratz, D. (1997), Subdivision direction selection in interval methods for global optimization, *SIAM J. Numerical Anal.*, 34(3), 922–938.
7. Hammer, R., Hocks, M., Kulisch, U. and Ratz, D. (1995), *C++ Toolbox for Verified Computing I: Basic Numerical Problems: Theory, Algorithms, and Programs*, Springer-Verlag, Berlin.

8. Hansen, E.R. (1992), *Global Optimization Using Interval Analysis*, Marcel Dekker, New York.
9. Kearfott, R.B. (1996), *Rigorous Global Search: Continuous Problems*, Kluwer, Boston.
10. Klatte, R., Kulisch, U., Wiethoff, A., Lawo, C. and Rauch, M. (1993), *C-XSC; A C++ Class Library for Extended Scientific Computing*, Springer-Verlag, New York.
11. Krawczyk R. and Nickel, K. (1982), The centered form in interval arithmetics: Quadratic convergence and inclusion isotonicity, *Computing*, 28(2), 117–137.
12. Lagouanelle, J.-L. (2000), Kite algorithm: A new global interval method from linear evaluations. In: Dietrich, S., Facius, A. and Bierlox, N. (eds.) *Abstracts of the SCAN2000 Conference*, p. 126. Karlsruhe.
13. Lagouanelle, J.-L. and Messine, F. (1998), Algorithme d'encadrement de l'optimum global d'une fonction différentiable, *C. R. Acad. Sci. Paris Sér. I Math.*, 326(5), 629–632.
14. Messine, F. and Lagouanelle, J.-L. (1998), Enclosure methods for multivariate differentiable functions and application to global optimization, *J.UCS: J. Universal Comput. Sci.*, 4(6), 589–603.
15. Moore, R.E. (1979), *Methods and Applications of Interval Analysis*, SIAM, Philadelphia.
16. Moore, R.E. (1996), *Interval Analysis*, Prentice-Hall, Englewood Cliffs.
17. Neumaier, A. (1990), *Interval Methods for Systems of Equations*, Cambridge University Press, Cambridge.
18. Ratschek, H. and Rokne, J. (1984), *Computer Methods for the Range of Functions*, Ellis Horwood, Chichester.
19. Ratschek, H. and Rokne, J. (1988), *New Computer Methods for Global Optimization*, Ellis Horwood, Chichester.
20. Ratz, D. (1999), A nonsmooth global optimization technique using slopes—the one dimensional case, *J. Global Optimization*, 14(4), 365–393.
21. Sotiropoulos, D.G. and Grapsa, T.N. (2001), A branch-and-prune method for global optimization. In: Kraemer, W. and Gudenberg, J.W.V. (eds). *Scientific Computing, Validated Numerics, Interval Methods*, Kluwer, Boston, p. 215–226.