Numerical experiences with a new generalized subinterval selection criterion for interval global optimization *

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

30.06.2002

Abstract. The convergence properties of interval global optimization algorithms are studied that select the next subinterval to be subdivided with the largest value of the indicator $pf(f_k, X) = \frac{f_k - \underline{F}(X)}{\overline{F}(X) - \underline{F}(X)}$. This time the more general case is investigated, when the global minimum value is unknown, and thus its estimation f_k in the iteration k has an important role.

Extensive numerical tests on 40 problems confirm that substantial improvements can be achieved both on simple and sophisticated algorithms by the new method (not utilizing the minimum value).

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

1. Introduction

The present paper extends the results of an earlier one, (Csendes, 2001) for the more general case, when the global minimum value is not previously known. Consider the bound constrained global optimization problem (Horst and Pardalos, 1995; Törn and Žilinskas, 1987)

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

where the n-dimensional interval X is the search region, and $f(x) : \mathbb{R}^n \to \mathbb{R}$ is the objective function. The algorithm considered is based on inclusion function calculated by interval arithmetic (Ratschek and Rokne, 1988): 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 $\forall y \in Y \quad f(y) \in F(Y)$, where $\mathbb{I}$ stands for the set of all closed real intervals. 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). Three important types or possible properties of inclusion functions are: F is said to be an *isotone inclusion function*

* This work was supported by the Grants OMFB D-30/2000, OMFB E-24/2001, OTKA T 032118 and T 034350.

over X if for $\forall Y, Z \in \mathbb{I}(X)$, $Y \subseteq Z$ implies $F(Y) \subseteq F(Z)$. We call the inclusion function F an *α-convergent inclusion function* over X if for $\forall Y \in \mathbb{I}(X)$ $w(F(Y)) - w(f(Y)) \leq C w^{\alpha}(Y)$ holds, where α and C are positive constants. 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). 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 can be used that delete subintervals not containing a global minimizer point or shrink subintervals. 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. It is important that no global minimizer point is lost in this process.

In (Csendes, 2001) the effects of the application of

$$p(f_k, Y) = \frac{f_k - \underline{F}(Y)}{\overline{F}(Y) - \underline{F}(Y)},$$

as an indicator that gives which interval is to be selected for subdivision was investigated. In the related algorithm that interval Y was chosen which had the maximal $p(\hat{f}, Y)$ value. Here f_k is the approximation of the global minimum value in the iteration k, and in this way $p(f_k, Y)$ is a variant of the *RejectIndex*, $pf^* = \frac{f^* - \underline{F}(Y)}{\overline{F}(Y) - \underline{F}(Y)}$ (Casado and García, 1998; Casado et al., 2000a; Casado et al., 2000b; Casado et al., 2002).

In (Csendes, 2001) the author investigated the convergence properties, and it was found that the necessary and sufficient conditions for the convergence to a set of global minimizer point were 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^*.

Now we investigate such an algorithm that can utilize the new interval selection rule without knowing the global minimum value. Its approximation f_k in the k-th iteration must be obviously between the known best lower and upper bounds of f^* :

$$\underline{f}_k = \min\{\underline{F}(Y^l), l = 1, ..., |L_k|\} \leq f_k < \tilde{f}_k = \overline{f}_k,$$

where $|L|$ stands for the cardinality of the elements of the list L. Here the list L_k is always the actual list, i.e. the one available at the iteration number k.

In (Csendes, 2002) the convergence properties of the introduced algorithm were investigated. It has been found that 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 converges exclusively to global minimizer points if

$$\underline{f}_k \leq f_k < \delta(\overline{f}_k - \underline{f}_k) + \underline{f}_k$$

holds for each iteration number k, where $0 < \delta < 1$.

Notice that if we set $f_k = \min\{\underline{F}(Y^l), l = 1, ..., |L|\}$ then we have actually the Moore-Skelboe algorithm, since then always that subinterval is selected which has the minimal lower bound, since for these intervals the pf value will be zero while for each other interval it will be negative. In this sense the present theorem is a generalization of the convergence assertion on the Moore-Skelboe algorithm.

In the following numerical test we set always f_k equal to the midpoint of the interval $[\underline{f}_k, \overline{f}_k] = [\min\{\underline{F}(Y^l), l = 1, ..., |L_k|\}, \tilde{f}_k]$. This value can be determined without the knowledge of the global minimum value.

2. Numerical Tests

The numerical tests were carried out on a Pentium-IV computer (1,4 Ghz, 1 Gbyte) 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 0.00076 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 the new interval selection rule compared to the Moore-Skelboe type (denoted by $\underline{F}$) and to the pf^* based one that requires the known global optimum value (denoted by pf^*). It was the only difference between the algorithm variants. The new algorithm does not utilized the minimum value to determine a global minimizer point.

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. This phenomenon is not there for the new interval selection rule, since in these cases the f_k values are not zeros.

2.1. Derivative free algorithms

First the basic algorithm with only the cut-off test was run in these three 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 Tables I and II. Notice that the summarizing figures in the last two rows are calculated only for those problems which can be solved with less than 20,000 memory units (by all algorithm variants).

Summarizing the results we can point out that although the new technique proves to be better than the Moore-Skelboe method in terms of computation time, it is caused basically by only one single exceptional figure for the otherwise difficult to solve Schwefel 2.7 problem. Also, the new algorithm fails to solve the same problems for which the pf^* method fails, with one exception, for which the new method succeeds. According to the memory complexity we have a mixed picture regarding the achievement of the new method. On the one hand it is the best according to the compound figures (Σ, %), but with the exception of again the Sch27 problem the results are quite scattered. Regarding the more balanced AoP values, we can state that both the CPU time and the memory complexity figures of the new method are between the respective values of the other two techniques, which is understandable considering that the new interval selection method is based on the pf^* principle but without the valuable information of the global minimum value.

In the Tables I, III, and VI, the given efficiency indicators are the required CPU time in seconds (CPUt) and the maximal list length necessary (MLL). In the Tables II, IV, and VII, the given efficiency indicators are the number of objective function evaluations (NFE), and the number of gradient evaluations (NGE). In the Tables V and VIII, the given efficiency indicators are the number of Hessian evaluations (NHE), and the number of iterations (NIT). The last two lines contain summarized figures for those problems which could be solved by all the

Table I. Numerical results for the basic algorithm with only the cut-off test.

Problem		CPUt					MLL				
name	dim.	$\underline{F}$	new		pf^*		$\underline{F}$	new		pf^*	
				%		%			%		%
S5	4	0.09	0.11	122	0.08	89	18	18	100	18	100
S7	4	0.33	1.19	361	0.22	67	67	186	278	52	78
S10	4	0.53	2.49	470	0.23	43	77	214	278	31	40
H3	3	347.64	431.98	124	5.59	2	20,000	20,000	100	2,383	12
H6	6	444.75	439.99	99	368.55	83	20,000	20,000	100	20,000	100
GP	2	474.79	1,760.60	371	3.48	1	20,000	20,000	100	2,145	11
SHCB	2	362.53	298.12	82	0.54	0	20,000	17,643	88	762	4
THCB	2	2.59	1.45	56	1.28	49	1,128	655	58	1,348	120
BR	2	0.03	0.02	67	0.01	33	54	27	50	16	30
RB	2	0.01	0.00	0	0.01	100	21	19	90	21	100
RB5	5	2.68	0.06	2	2.86	107	896	55	6	896	100
L3	2	387.02	443.24	115	0.09	0	20,000	20,000	100	72	0
L5	2	381.78	319.82	84	0.05	0	20,000	20,000	100	37	0
L8	3	0.02	0.02	100	0.02	100	13	13	100	13	100
L9	4	0.04	0.03	75	0.04	100	17	17	100	17	100
L10	5	0.07	0.06	86	0.07	100	22	22	100	22	100
L11	8	0.23	0.25	109	0.24	104	38	38	100	38	100
L12	10	0.43	0.45	105	0.42	98	48	48	100	48	100
L13	2	0.01	0.01	100	0.01	100	10	11	110	10	100
L14	3	0.02	0.03	150	0.02	100	15	16	107	15	100
L15	4	0.04	0.05	125	0.04	100	20	21	105	20	100
L16	5	0.06	0.06	100	0.06	100	25	26	104	25	100
L18	7	0.15	0.16	107	0.15	100	35	36	103	35	100
Sch21	2	0.08	0.08	100	0.08	100	44	44	100	44	100
Sch31	3	0.02	0.02	100	0.01	50	13	18	138	13	100
Sch25	2	0.01	0.01	100	0.01	100	25	19	76	25	100
Sch27	3	114.40	0.06	0	115.27	101	5,706	16	0	5,706	100
Sch214	4	0.17	0.11	65	0.17	100	160	178	111	160	100
Sch218	2	2.65	1.08	41	0.01	0	678	342	50	78	12
Sch32	3	0.02	0.03	150	0.02	100	32	64	200	32	100
Sch37	5	0.01	0.00	0	0.01	100	2	2	100	2	100
Sch37	10	0.01	0.01	100	0.01	100	2	2	100	2	100
G5	5	2.43	0.11	5	2.41	99	32	32	100	32	100
G7	7	0.18	0.24	133	0.17	94	58	82	141	58	100
R4	2	0.64	0.38	59	0.03	5	820	473	58	81	10
R5	3	0.10	0.01	10	0.10	100	98	15	15	98	100
R6	5	0.33	0.06	18	0.34	103	168	23	14	168	100
R7	7	1.10	0.13	12	1.11	101	314	35	11	314	100
R8	9	2.51	0.28	11	2.57	102	460	46	10	460	100
EX2	5	358.43	354.16	99	328.91	92	20,000	20,000	100	20,000	100
Σ		131.99	9.05	7	128.08	97	11,116	2,813	25	9,898	89
	AoP			92		86			94		91

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 methods (denoted by new and pf^*) compared to that of the old one as per cents. The last line contains the average of the percentages (AoP) for the respective columns. In the Tables I - V the search is stopped after the first candidate interval good enough (i.e. when the stopping criterion is met, when the diameter of a candidate interval was smaller than 0.01). For the last three tables the search is *not* stopped in this case.

Table II. Numerical results for the basic algorithm with only the cut-off test.

Problem		NFE					NIT				
name	dim.	$\underline{F}$	new		pf^*		$\underline{F}$	new		pf^*	
				%		%			%		%
S5	4	139	157	113	130	94	46	52	113	43	93
S7	4	352	1,273	362	253	72	117	424	362	84	72
S10	4	403	1,900	471	184	46	134	633	472	61	46
H3	3	66,817	120,055	180	7,159	11	22,272	40,018	180	2,386	11
H6	6	64,105	68,074	106	60,202	94	21,368	22,691	106	20,067	94
GP	2	70,663	636,550	901	6,433	9	23,554	212,183	901	2,144	9
SHCB	2	78,883	129,211	164	2,329	3	26,294	43,070	164	776	3
THCB	2	16,660	16,705	100	4,108	25	5,553	5,568	100	1,369	25
BR	2	382	277	73	94	25	127	92	72	31	24
RB	2	115	67	58	115	100	38	22	58	38	100
RB5	5	6,874	211	3	6,874	100	2,291	70	3	2,291	100
L3	2	70,774	118,153	167	283	0	23,591	39,384	167	94	0
L5	2	69,223	87,520	126	121	0	23,074	29,173	126	40	0
L8	3	64	70	109	64	100	21	23	110	21	100
L9	4	85	91	107	85	100	28	30	107	28	100
L10	5	106	112	106	106	100	35	37	106	35	100
L11	8	172	178	103	172	100	57	59	104	57	100
L12	10	214	220	103	214	100	71	73	103	71	100
L13	2	64	70	109	64	100	21	23	110	21	100
L14	3	94	100	106	94	100	31	33	106	31	100
L15	4	124	130	105	124	100	41	43	105	41	100
L16	5	139	145	104	139	100	46	48	104	46	100
L18	7	193	199	103	193	100	64	66	103	64	100
Sch21	2	913	913	100	913	100	304	304	100	304	100
Sch31	3	88	112	127	88	100	29	37	128	29	100
Sch25	2	205	85	41	205	100	68	28	41	68	100
Sch27	3	60,535	88	0	60,535	100	20,178	29	0	20,178	100
Sch214	4	1,117	772	69	1,117	100	372	257	69	372	100
Sch218	2	14,770	15,415	104	232	2	4,923	5,138	104	77	2
Sch32	3	169	376	222	169	100	56	125	223	56	100
Sch37	5	4	4	100	4	100	1	1	100	1	100
Sch37	10	4	4	100	4	100	1	1	100	1	100
G5	5	6,142	283	5	6,142	100	2,047	94	5	2,047	100
G7	7	274	346	126	274	100	91	115	126	91	100
R4	2	3,226	2,812	87	280	9	1,075	937	87	93	9
R5	3	388	58	15	388	100	129	19	15	129	100
R6	5	574	94	16	574	100	191	31	16	191	100
R7	7	1,048	130	12	1,048	100	349	43	12	349	100
R8	9	1,522	172	11	1,522	100	507	57	11	507	100
EX2	5	62,407	78,151	125	63,823	102	20,802	26,050	125	21,274	102
Σ		117,159	43,569	37	86,508	74	39,042	14,512	37	28,825	74
	AoP			102		87			102		87

The number of function evaluations and the number of iterations go understandably in parallel according to Table II. Here again the new method is the best according to the summarized figures (Σ) — but again mainly just due to the exceptional efficiency achieved on the Sch27 problem. Also, remember, that these compound indicators are calculated for those problems which were solved with less than 20,000 memory units, and the pf^* variant could solve all the problems with only one exception. This fact that the pf^* method solves more problems is more important for the user than the efficiency figures alone. On the other hand, the behaviour of the algorithm with the new method is easily explained by the less information it can rely on.

Summarizing the numerical experiences with the simple algorithm variants, we can conclude that the new method is definitely better than the old one at least on the whole set of the given test problems. We have found the tendency 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.

2.2. 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 all the three algorithm variants use 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 III-V.

The first observation is that with the accelerating devices all three algorithm variants were able to solve all the test problems even with the much more strict stopping criterion. The memory requirements were substantially smaller. The efficiency measures for the new interval selection rule are quite scattered: e.g. the CPU time used for the solution ranges from 7% to 821% as compared to the similar values for the Moore-Skelboe algorithm version. The sum of the CPU time used by the new technique was 218.79 seconds, which is 81 percent of that of the basic method, indicating a substantial time saving for the whole set of problems. On the other hand, if we calculate the average relative change for the test problems (the average of the percentages for the singular problems), then the new method means a 53% worse achievement. These two indicators together state that the new interval

Table III. Numerical results with all of the available accelerating devices.

Problem		CPUt					MLL				
name	dim.	$\underline{F}$	new	%	pf^*	%	$\underline{F}$	new	%	pf^*	%
S5	4	0.09	0.10	111	0.10	111	10	10	100	10	100
S7	4	0.13	0.14	108	0.14	108	14	14	100	14	100
S10	4	0.20	0.20	100	0.19	95	16	16	100	16	100
H3	3	0.11	0.11	100	0.11	100	9	10	111	20	222
H6	6	2.00	2.01	101	2.05	103	78	64	82	65	83
GP	2	3.01	2.65	88	0.10	3	470	308	66	153	33
SHCB	2	0.07	0.06	86	0.01	14	49	35	71	22	45
THCB	2	0.02	0.03	150	0.02	100	16	13	81	20	125
BR	2	0.01	0.02	200	0.01	100	9	10	111	11	122
RB	2	0.01	0.02	200	0.02	200	11	9	82	11	100
RB5	5	1.32	0.49	37	1.31	99	72	45	63	72	100
L3	2	0.85	0.85	100	0.12	14	138	99	72	57	41
L5	2	0.27	0.27	100	0.27	100	31	40	129	32	103
L8	3	0.02	0.03	150	0.03	150	9	9	100	9	100
L9	4	0.06	0.06	100	0.06	100	13	13	100	13	100
L10	5	0.09	0.10	111	0.09	100	15	15	100	15	100
L11	8	0.30	0.41	137	0.31	103	28	30	107	28	100
L12	10	0.61	0.92	151	0.61	100	36	38	106	36	100
L13	2	0.01	0.03	300	0.01	100	9	14	156	9	100
L14	3	0.03	0.03	100	0.03	100	12	12	100	12	100
L15	4	0.05	0.13	260	0.05	100	19	22	116	19	100
L16	5	0.07	0.44	629	0.07	100	20	66	330	20	100
L18	7	0.19	1.56	821	0.19	100	26	118	454	26	100
Sch21	2	0.08	0.08	100	0.08	100	25	25	100	25	100
Sch31	3	0.01	0.02	200	0.01	100	6	6	100	6	100
Sch25	2	0.02	0.02	100	0.02	100	4	11	275	4	100
Sch27	3	2.43	0.20	8	2.44	100	236	19	8	236	100
Sch214	4	0.37	0.41	111	0.37	100	78	70	90	78	100
Sch218	2	0.02	0.02	100	0.00	0	7	6	86	4	57
Sch32	3	0.02	0.04	200	0.02	100	7	10	143	7	100
Sch37	5	0.05	0.05	100	0.06	120	32	32	100	32	100
Sch37	10	3.20	3.16	99	3.20	100	818	818	100	818	100
G5	5	2.32	0.16	7	2.30	99	32	18	56	32	100
G7	7	0.23	1.33	578	0.23	100	58	211	364	58	100
R4	2	0.10	0.10	100	0.02	20	40	33	83	31	78
R5	3	0.31	0.05	16	0.32	103	57	10	18	57	100
R6	5	1.08	0.33	31	1.07	99	30	20	67	30	100
R7	7	3.41	1.15	34	3.41	100	41	27	66	41	100
R8	9	9.21	2.91	32	9.23	100	59	55	93	59	100
EX2	4	237.32	198.10	83	82.03	35	1,968	1,225	62	2,733	139
Σ		269.70	218.79	81	110.71	41	4,608	3,606	78	4,941	107
	AoP			153		92			116		99

selection rule caused improvements better on hard to solve problems, that required more CPU time. For the pf^* based algorithm version, that could utilize the known global minimum value the efficiency improvements were larger (41%), and even for the AoP value we could find a better value than that of the basic method (92%).

The other efficiency indicators give a similar picture: the new technique is between the basic one and that of based on pf^* with the exception of the maximal list length needed and the number of iterations. In the earlier case the new method proved to be the best on the

Table IV. Numerical results with all of the available accelerating devices.

Problem		NFE					NGE				
name	dim.	$\underline{F}$	new	%	pf^*	%	$\underline{F}$	new	%	pf^*	%
S5	4	144	148	103	144	100	88	91	103	88	100
S7	4	149	155	104	149	100	86	89	103	86	100
S10	4	158	161	102	141	89	90	96	107	80	89
H3	3	303	303	100	297	98	193	193	100	190	98
H6	6	1,681	1,699	101	1,725	103	1,137	1,160	102	1,190	105
GP	2	15,694	15,317	98	536	3	9,195	8,968	98	231	3
SHCB	2	734	730	99	122	17	435	433	100	62	14
THCB	2	347	347	100	291	84	215	215	100	177	82
BR	2	202	164	81	91	45	123	96	78	47	38
RB	2	278	435	156	278	100	153	257	168	153	100
RB5	5	4,234	1,616	38	4,234	100	2,878	978	34	2,878	100
L3	2	2,225	2,219	100	342	15	1,326	1,354	102	172	13
L5	2	676	690	102	681	101	406	402	99	408	100
L8	3	93	93	100	93	100	57	57	100	57	100
L9	4	122	136	111	122	100	75	85	113	75	100
L10	5	142	157	111	142	100	88	99	113	88	100
L11	8	214	284	133	214	100	130	182	140	130	100
L12	10	288	431	150	288	100	179	282	158	179	100
L13	2	86	196	228	86	100	49	115	235	49	100
L14	3	137	153	112	137	100	79	91	115	79	100
L15	4	155	361	233	155	100	89	222	249	89	100
L16	5	163	890	546	163	100	90	532	591	90	100
L18	7	235	1,851	788	235	100	132	1,159	878	132	100
Sch21	2	918	918	100	918	100	582	582	100	582	100
Sch31	3	112	119	106	112	100	66	69	105	66	100
Sch25	2	345	338	98	345	100	207	202	98	207	100
Sch27	3	3,597	306	9	3,597	100	2,217	191	9	2,217	100
Sch214	4	2,587	2,897	112	2,587	100	1,599	1,780	111	1,599	100
Sch218	2	290	308	106	26	9	176	188	107	15	9
Sch32	3	217	428	197	217	100	126	260	206	126	100
Sch37	5	374	374	100	374	100	228	228	100	228	100
Sch37	10	5,087	5,087	100	5,087	100	2,685	2,685	100	2,685	100
G5	5	5,869	377	6	5,869	100	3,648	221	6	3,648	100
G7	7	339	1,901	561	339	100	175	988	565	175	100
R4	2	1,053	997	95	201	19	650	616	95	102	16
R5	3	1,105	182	16	1,105	100	750	114	15	750	100
R6	5	1,658	537	32	1,658	100	1,223	365	30	1,223	100
R7	7	2,934	1,023	35	2,934	100	2,295	730	32	2,295	100
R8	9	5,035	1,660	33	5,035	100	4,065	1,205	30	4,065	100
EX2	4	274,229	271,217	99	101,555	37	171,409	169,989	99	62,359	36
Σ		334,209	317,205	95	142,625	43	209,394	197,569	94	89,072	43
	AoP			140		86			145		85

whole set of problems. Still the earlier pattern according to which the pf^* based method gave several times the same efficiency figure as the basic algorithm is now changed: the new technique seems to follow a different search path.

In the next round of tests we have allowed the approximation of several global minimizer points, i.e. the search was not stopped after the first found good enough candidate interval. This algorithm setting is obviously not advantageous for the new interval selection rules, since their strength lies in the quick convergence to one of the global mini-

Table V. Numerical results with all of the available accelerating devices.

Problem		NHE					NIT				
name	dim.	$\underline{F}$	new		pf^*		$\underline{F}$	new		pf^*	
				%		%			%		%
S5	4	8	8	100	8	100	16	17	106	16	100
S7	4	8	8	100	8	100	18	19	106	18	100
S10	4	8	8	100	7	88	18	20	111	17	94
H3	3	23	23	100	23	100	39	39	100	38	97
H6	6	83	86	104	89	107	193	190	98	192	99
GP	2	623	601	96	1	0	2,256	2,266	100	76	3
SHCB	2	27	26	96	4	15	121	121	100	17	14
THCB	2	24	24	100	19	79	49	49	100	41	84
BR	2	14	9	64	4	29	29	62	214	12	41
RB	2	12	27	225	12	100	38	67	176	38	100
RB5	5	330	94	28	330	100	419	718	171	419	100
L3	2	105	113	108	9	9	289	287	99	47	16
L5	2	31	26	84	32	103	87	94	108	86	99
L8	3	6	6	100	6	100	11	11	100	11	100
L9	4	8	9	113	8	100	13	14	108	13	100
L10	5	9	10	111	9	100	15	16	107	15	100
L11	8	10	14	140	10	100	23	28	122	23	100
L12	10	13	19	146	13	100	30	42	140	30	100
L13	2	5	10	200	5	100	10	24	240	10	100
L14	3	8	9	113	8	100	15	17	113	15	100
L15	4	8	20	250	8	100	17	35	206	17	100
L16	5	7	36	514	7	100	19	93	489	19	100
L18	7	9	69	767	9	100	27	186	689	27	100
Sch21	2	54	54	100	54	100	113	190	168	113	100
Sch31	3	6	6	100	6	100	14	15	107	14	100
Sch25	2	28	27	96	28	100	50	56	112	50	100
Sch27	3	252	27	11	252	100	430	421	98	430	100
Sch214	4	161	164	102	161	100	307	609	198	307	100
Sch218	2	21	23	110	2	10	44	46	105	3	7
Sch32	3	13	30	231	13	100	28	63	225	28	100
Sch37	5	34	34	100	34	100	45	53	118	45	100
Sch37	10	202	202	100	202	100	696	1,213	174	696	100
G5	5	331	20	6	331	100	351	667	190	351	100
G7	7	9	46	511	9	100	40	187	468	40	100
R4	2	52	49	94	5	10	154	154	100	28	18
R5	3	72	10	14	72	100	107	132	123	107	100
R6	5	101	27	27	101	100	140	206	147	140	100
R7	7	169	47	28	169	100	204	309	151	204	100
R8	9	262	68	26	262	100	310	498	161	310	100
EX2	4	18,999	18,801	99	7,126	38	39,585	39,600	100	14,055	36
Σ		22,145	20,890	94	9,456	43	46,370	48,834	105	18,118	39
	AoP			138		85			166		85

mizer points — while others may remain hidden in the list of boxes to be processed. Although this approach is acceptable in many practical situations, in other cases all the global minimizer points are of interest. The numerical results for these algorithm variants are compiled in Tables VI to VIII.

We can observe the same trends as in the Tables III to V, for the algorithm variants that approximate only one global minimizer point, but the differences are less between the methods. Still the pf^* based algorithm is the most efficient, followed by the new technique. In other

Table VI. Numerical results with all of the available accelerating devices.

Problem		CPUt					MLL				
name	dim.	$\underline{F}$	new		pf^*		$\underline{F}$	new		pf^*	
				%		%			%		%
S5	4	0.10	0.09	90	0.10	100	10	10	100	10	100
S7	4	0.13	0.14	108	0.14	108	14	14	100	14	100
S10	4	0.21	0.21	100	0.19	90	16	16	100	16	100
H3	3	0.12	0.12	100	0.11	92	9	10	111	20	222
H6	6	2.03	2.02	100	2.09	103	78	64	82	65	83
GP	2	3.05	2.75	90	3.28	108	470	308	66	1,111	236
SHCB	2	0.07	0.07	100	0.06	86	49	35	71	49	100
THCB	2	0.02	0.03	150	0.02	100	16	13	81	20	125
BR	2	0.02	0.03	150	0.02	100	9	7	78	11	122
RB	2	0.01	0.03	300	0.01	100	11	9	82	11	100
RB5	5	1.70	2.09	123	1.70	100	72	45	63	72	100
L3	2	0.87	0.87	100	0.85	98	138	99	72	78	57
L5	2	0.27	0.28	104	0.27	100	31	40	129	32	103
L8	3	0.03	0.03	100	0.03	100	9	9	100	9	100
L9	4	0.06	0.06	100	0.06	100	13	13	100	13	100
L10	5	0.09	0.10	111	0.09	100	15	15	100	15	100
L11	8	0.31	0.41	132	0.30	97	28	30	107	28	100
L12	10	0.61	0.93	152	0.62	102	36	38	106	36	100
L13	2	0.01	0.03	300	0.01	100	9	14	156	9	100
L14	3	0.04	0.04	100	0.03	75	12	12	100	12	100
L15	4	0.06	0.13	217	0.05	83	19	22	116	19	100
L16	5	0.08	0.44	550	0.08	100	20	66	330	20	100
L18	7	0.19	1.58	832	0.19	100	26	118	454	26	100
Sch21	2	0.12	0.13	108	0.12	100	25	25	100	25	100
Sch31	3	0.01	0.02	200	0.01	100	6	6	100	6	100
Sch25	2	0.02	0.02	100	0.02	100	4	11	275	4	100
Sch27	3	70.10	3.27	5	69.92	100	3,890	37	1	3,890	100
Sch214	4	0.43	0.69	160	0.44	102	78	88	113	78	100
Sch218	2	0.01	0.02	200	0.02	200	7	6	86	23	329
Sch32	3	0.02	0.05	250	0.02	100	7	10	143	7	100
Sch37	5	0.20	0.19	95	0.19	95	32	32	100	32	100
Sch37	10	7.67	7.75	101	7.70	100	914	914	100	914	100
G5	5	4.62	4.59	99	4.61	100	32	22	69	32	100
G7	7	0.22	1.34	609	0.23	105	58	211	364	58	100
R4	2	0.10	0.10	100	0.10	100	40	33	83	67	168
R5	3	0.54	0.49	91	0.55	102	68	11	16	68	100
R6	5	1.43	1.72	120	1.48	103	30	20	67	30	100
R7	7	3.82	5.23	137	3.81	100	41	27	66	41	100
R8	9	10.55	14.92	141	10.54	100	59	55	93	59	100
EX2	4	239.73	203.93	85	96.13	40	1,968	1,225	62	4,253	216
Σ		349.67	256.94	73	206.19	59	8,369	3,740	45	11,283	135
	AoP			170		100			116		117

words, for problems where the global minimum value is available, the pf^* based interval selection is the best choice, while if it is not the case, then the new rule is better than the Moore-Skelboe method even for the problems when all the global minimizers are to be found.

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

Table VII. Numerical results with all of the available accelerating devices.

Problem		NFE					NGE				
name	dim.	$\underline{F}$	new		pf^*		$\underline{F}$	new		pf^*	
				%		%			%		%
S5	4	144	148	103	144	100	88	91	103	88	100
S7	4	149	155	104	149	100	86	89	103	86	100
S10	4	158	161	102	145	92	90	96	107	83	92
H3	3	303	303	100	297	98	193	193	100	190	98
H6	6	1,681	1,699	101	1,725	103	1,137	1,160	102	1,190	105
GP	2	15,788	15,788	100	15,788	100	9,261	9,261	100	9,261	100
SHCB	2	737	733	99	721	98	437	435	100	429	98
THCB	2	356	356	100	356	100	221	221	100	221	100
BR	2	243	369	152	310	128	150	208	139	189	126
RB	2	316	457	145	316	100	180	273	152	180	100
RB5	5	5,450	6,688	123	5,450	100	3,685	4,442	121	3,685	100
L3	2	2,249	2,279	101	2,271	101	1,342	1,394	104	1,393	104
L5	2	676	690	102	681	101	406	402	99	408	100
L8	3	93	93	100	93	100	57	57	100	57	100
L9	4	122	136	111	122	100	75	85	113	75	100
L10	5	142	157	111	142	100	88	99	113	88	100
L11	8	214	284	133	214	100	130	182	140	130	100
L12	10	288	431	150	288	100	179	282	158	179	100
L13	2	86	196	228	86	100	49	115	235	49	100
L14	3	137	153	112	137	100	79	91	115	79	100
L15	4	155	361	233	155	100	89	222	249	89	100
L16	5	163	890	546	163	100	90	532	591	90	100
L18	7	235	1,851	788	235	100	132	1,159	878	132	100
Sch21	2	1,456	1,456	100	1,456	100	919	919	100	919	100
Sch31	3	112	119	106	112	100	66	69	105	66	100
Sch25	2	352	387	110	352	100	212	237	112	212	100
Sch27	3	70,803	4,825	7	70,803	100	44,937	3,497	8	44,937	100
Sch214	4	3,073	4,727	154	3,073	100	1,923	2,840	148	1,923	100
Sch218	2	320	320	100	320	100	196	196	100	196	100
Sch32	3	236	468	198	236	100	140	289	206	140	100
Sch37	5	1,097	1,097	100	1,097	100	896	896	100	896	100
Sch37	10	11,446	11,446	100	11,446	100	7,368	7,368	100	7,368	100
G5	5	11,020	11,020	100	11,020	100	6,824	6,824	100	6,824	100
G7	7	339	1,901	561	339	100	175	988	565	175	100
R4	2	1,068	1,068	100	1,068	100	660	660	100	660	100
R5	3	1,878	1,703	91	1,878	100	1,288	1,165	90	1,288	100
R6	5	2,188	2,621	120	2,188	100	1,627	1,946	120	1,627	100
R7	7	3,279	4,477	137	3,279	100	2,567	3,484	136	2,567	100
R8	9	5,748	8,089	141	5,748	100	4,627	6,490	140	4,627	100
EX2	4	279,075	274,682	98	117,745	42	174,918	172,617	99	71,773	41
Σ		423,375	364,784	86	262,148	62	267,587	231,574	87	164,569	62
	AoP			157		99			161		99

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.

Table VIII. Numerical results with all of the available accelerating devices.

Problem		NHE					NIT				
name	dim.	$\underline{F}$	new	%	pf^*	%	$\underline{F}$	new	%	pf^*	%
S5	4	8	8	100	8	100	16	17	106	16	100
S7	4	8	8	100	8	100	18	19	106	18	100
S10	4	8	8	100	7	88	18	20	111	18	100
H3	3	23	23	100	23	100	39	39	100	38	97
H6	6	83	86	104	89	107	193	190	98	192	99
GP	2	641	641	100	641	100	2,226	2,266	102	2,266	102
SHCB	2	28	27	96	28	100	121	121	100	119	98
THCB	2	27	27	100	27	100	49	49	100	49	100
BR	2	19	33	174	23	121	34	62	182	44	129
RB	2	18	29	161	18	100	43	67	156	43	100
RB5	5	464	540	116	464	100	550	718	131	550	100
L3	2	113	125	111	126	112	289	287	99	285	99
L5	2	31	26	84	32	103	87	94	108	86	99
L8	3	6	6	100	6	100	11	11	100	11	100
L9	4	8	9	113	8	100	13	14	108	13	100
L10	5	9	10	111	9	100	15	16	107	15	100
L11	8	10	14	140	10	100	23	28	122	23	100
L12	10	13	19	146	13	100	30	42	140	30	100
L13	2	5	10	200	5	100	10	24	240	10	100
L14	3	8	9	113	8	100	15	17	113	15	100
L15	4	8	20	250	8	100	17	35	206	17	100
L16	5	7	36	514	7	100	19	93	489	19	100
L18	7	9	69	767	9	100	27	186	689	27	100
Sch21	2	105	105	100	105	100	190	190	100	190	100
Sch31	3	6	6	100	6	100	14	15	107	14	100
Sch25	2	29	34	117	29	100	51	56	110	51	100
Sch27	3	5,830	986	17	5,830	100	8,161	421	5	8,161	100
Sch214	4	231	338	146	231	100	358	609	170	358	100
Sch218	2	27	27	100	27	100	46	46	100	46	100
Sch32	3	15	34	227	15	100	31	63	203	31	100
Sch37	5	352	352	100	352	100	53	53	100	53	100
Sch37	10	1,576	1,576	100	1,576	100	1,213	1,213	100	1,213	100
G5	5	687	687	100	687	100	667	667	100	667	100
G7	7	9	46	511	9	100	40	187	468	40	100
R4	2	57	57	100	57	100	154	154	100	154	100
R5	3	196	182	93	196	100	149	132	89	149	100
R6	5	162	189	117	162	100	172	206	120	172	100
R7	7	206	277	134	206	100	222	309	139	222	100
R8	9	335	444	133	335	100	342	498	146	342	100
EX2	4	19,910	19,625	99	8,448	42	40,092	39,600	99	16,107	40
Σ		31,287	26,748	85	19,848	63	55,818	48,834	87	31,874	57
	AoP			155		99			152		99

2.3. A better approximate value of the global minimum?

In the last set of numerical tests we have followed the runs of the pf^* algorithm variant, and we have recorded what value between zero and one could be the correct one instead of the 0.5 used in the new interval selection rule. The obtained sequences show unfortunately the disappointing fact that no fixed value could provide the numerical efficiency of the pf^* based method. In detail, e.g. for the hard test problem EX2 the value of 0.3533603467547605 should have been used and this was very stable along the iterations followed. For the Goldstein-Price problem the optimal value should have been 0.999, and it has decreased slightly with the iteration number. In the case of the Hartman-3 problem the optimal value started from 0.584, and after a short decreasing period it began to increase to the value of 0.9945846045740079. For the Hartman-6 problem a similar tendency was noticed: started from 0.643, the optimal value decreased shortly, and then increased to about 0.919. In the case of the Levy-3 and 5 test functions, the optimal value started from 0.174, and increased monotonously to 0.9835267145732338 and 0.9934440693491524, respectively. For the Schwefel 2.7 problem it should be a constant zero, while for the Six-Hump-Camel-Back function the optimal value stabilized after an initial decrease at 0.9792010392442.

These very differing patterns cannot be followed without additional knowledge on the given problem instance, which is obviously reflected in a way in the structure of overestimations, and thus in the a posteriori optimal values used in the new interval selection rule. There remains only one chance to achieve closely the efficiency of the pf^* based interval selection rule: to use some additional, low computational cost procedures for the good estimation of the global optimum value in parallel to the interval optimization algorithm. One possible suggestion is to run regularly local search procedures starting from those points of the search domain that are not yet deleted by the interval algorithm.

3. Conclusion

The moral of the summarized numerical tests is that in cases when the global minimum value is not known a priori, the suggested new interval selection rule can improve the efficiency of interval global optimization algorithms. On the other hand, it seems that a good estimate of the minimum value can increase the improvements substantially. This value is not easy to approximate only on the basis of the inclusion function based available information. Hence it may be worthwhile to use a real, point evaluation based local search technique to provide it, especially for hard to solve optimization problems for which the additional computational costs of the local searches applied can be negligible.

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, BIT, accepted for publication.

Casado, L. G., I. García, and T. Csendes: A new multisection technique in interval methods for global optimization, Computing, 65(2000) 263-269.

Casado, L. G., I. García, and T. Csendes: Heuristic Rejection in Interval Global Optimization. Submitted for publication, available at http://www.inf.u-szeged.hu/~csendes/publ.html

Csendes, T.: Convergence properties of interval global optimization algorithms with a new class of interval selection criteria. J. Global Optimization 19(2001) 307-327

Csendes, T.: A new generalized interval selection criterion for interval global optimization. In preparation, available at http://www.inf.u-szeged.hu/~csendes/publ.html

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.