INTERVAL ANALYSIS: SUBDIVISION DIRECTIONS IN INTERVAL B&B METHODS

The selection of *subdivision direction* is one of the points where the efficiency of the basic *branch-and-bound algorithm* for unconstrained global optimization can be improved (see **Interval analysis: unconstrained and constrained optimization**). The traditional approach is to choose that direction for subdivision in which the actual box has the largest width. If the inclusion function $\boldsymbol{\phi}(x)$ is the only available information about the problem

$$\min_{x \in \boldsymbol{x}_0} \phi(x),$$

then it is usually the best possible choice. If, however, other information like the inclusion of the gradient ($\boldsymbol{\nabla\phi}$), or even the inclusion of the Hessian ($\boldsymbol{H}$) is calculated, then a better decision can be made.

Subdivision directions.

All the rules select a direction with a merit function:

$$k := \arg\max_{i=1}^{n} D(i), \tag{1}$$

where $D(i)$ is determined by the given rule. If many such optimal k indices exist then the algorithm can chose the smallest one, or it can select an optimal direction randomly.

Rule A. The first rule was the interval-width oriented rule. This rule chooses the coordinate direction with

$$D(i) := w(\boldsymbol{x}_i). \tag{2}$$

This rule is justified by the idea that, if the original interval is subdivided in a uniform way, then the width of the actual subintervals goes to zero most rapidly.

The algorithm with Rule A is convergent both with and without the monotonicity test [8]. This rule allows a relatively simple analysis of the convergence speed (as in [8], Chapter 3, Theorem 6).

Rule B. `E. Hansen` described another rule (initiated by `G. W. Walster`) [5]. The direct aim of this heuristic direction selection rule is to find the component for which $W_i = \max_{t \in \boldsymbol{x}_i} \phi(m_1, \ldots, m_{i-1}, t, m_{i+1}, \ldots, m_n) - \min_{t \in \boldsymbol{x}_i} \phi(m_1, \ldots, m_{i-1}, t, m_{i+1}, \ldots, m_n)$ is the largest (where $m_i = (\underline{\boldsymbol{x}}_i + \overline{\boldsymbol{x}}_i)/2$ is the midpoint of the interval $\boldsymbol{x}_i$). The factor W_i, that should reflect how much ϕ varies as x_i varies over $\boldsymbol{x}_i$, is then approximated by $w(\boldsymbol{\nabla\phi}_i(\boldsymbol{x}))w(\boldsymbol{x}_i)$ (where $\boldsymbol{\nabla\phi}_i(\boldsymbol{x})$ denotes the i-th component of $\boldsymbol{\nabla\phi}(\boldsymbol{x})$). The latter is not an upper bound for W_i (cf. [5] page 131 and Example 2 in Section 3 of [4]), yet it can be useful as a merit function.

Rule B selects the coordinate direction, for which (1) holds with

$$D(i) := w(\boldsymbol{\nabla\phi}_i(\boldsymbol{x}))w(\boldsymbol{x}_i). \tag{3}$$

It should be noted that the basic bisection algorithm represents only one way in which Rule B was applied in [5]. There the subdivision was, e.g., also carried out for many directions in a single iteration step.

Rule C. The next rule was defined by `D. Ratz` [9]. The underlying idea was to minimize the width of the inclusion: $w(\boldsymbol{\phi}(\boldsymbol{x})) = w(\boldsymbol{\phi}(\boldsymbol{x}) - \boldsymbol{\phi}(m(\boldsymbol{x}))) \approx w(\boldsymbol{\nabla\phi}(\boldsymbol{x})(\boldsymbol{x} - m(\boldsymbol{x}))) = \sum_{i=1}^{n} w(\boldsymbol{\nabla\phi}_i(\boldsymbol{x})(\boldsymbol{x}_i - m(\boldsymbol{x}_i)))$. Obviously, that component is to be chosen for which the term $w(\boldsymbol{\nabla\phi}_i(\boldsymbol{x})(\boldsymbol{x}_i - m(\boldsymbol{x}_i)))$ is the largest. Thus, Rule C can also be formulated with (1) and

$$D(i) := w(\boldsymbol{\nabla\phi}_i(\boldsymbol{x})(\boldsymbol{x}_i - m(\boldsymbol{x}_i))). \tag{4}$$

The important difference between (3) and (4) is that in Rule C the width of the multiplied intervals is maximized, not the multiplied widths of the respective intervals (and these are in general not equal). After a short calculation, the right-hand side of (4) can be written as $\max\{|\min \boldsymbol{\nabla\phi}_i(\boldsymbol{x})|, |\max \boldsymbol{\nabla\phi}_i(\boldsymbol{x})|\}w(\boldsymbol{x}_i)$. This

subdivision direction
branch-and-bound algorithm
Interval analysis: unconstrained and constrained optimization
`E. Hansen`
`G. W. Walster`
`D. Ratz`
maximum smear
`R.B. Kearfott`

corresponds to the *maximum smear* defined by R.B. Kearfott (used as a direction selection merit function solving systems of nonlinear equations [6, 7]) for the case $\phi : \mathbb{R}^n \to \mathbb{R}$. It is easy to see that the Rules B and C give the same merit function value if and only if either $\underline{\nabla \phi_i}(\boldsymbol{x}) = 0$ or $\overline{\nabla \phi_i}(\boldsymbol{x}) = 0$.

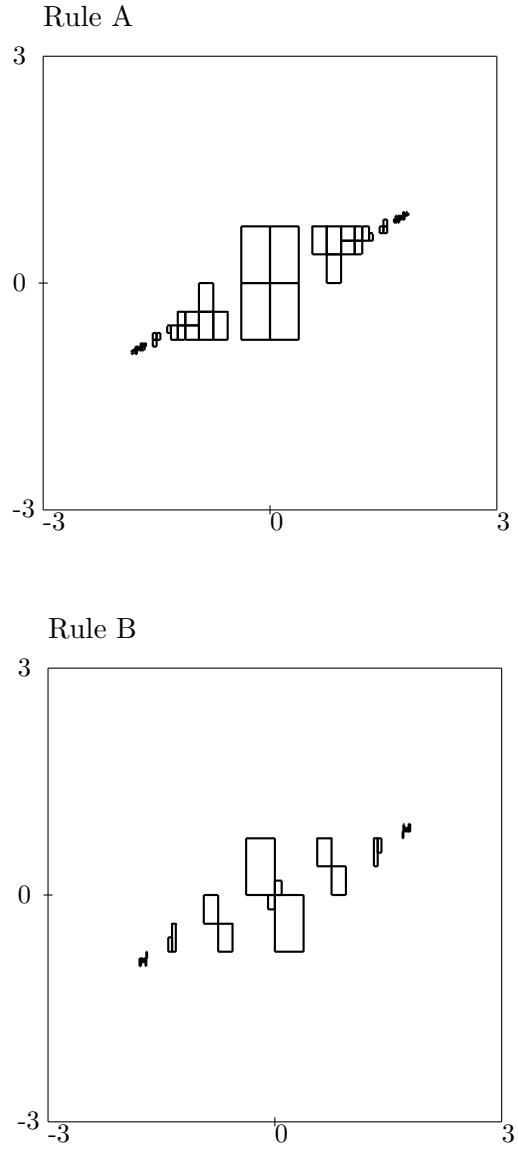

Figure 1. Remaining subintervals after 250 iteration steps of the model algorithm with the direction selection rules A, and B for the Three-Hump-Camel-Back problem [3].

Rule D. The fourth rule, Rule D is derivative-free like Rule A, and reflects the machine representation of the inclusion function $\boldsymbol{\phi}(\boldsymbol{x})$ (see [5]). It is again defined by (1) and by

$$D(i) := \begin{cases} w(\boldsymbol{x}_i) & \text{if } 0 \in \boldsymbol{x}_i, \\ w(\boldsymbol{x}_i)/<\boldsymbol{x}_i> & \text{otherwise,} \end{cases} \tag{5}$$

where $<\boldsymbol{x}>$ is the *mignitude* of the interval $\boldsymbol{x}$: $<\boldsymbol{x}> := \min_{x \in \boldsymbol{x}} |x|$.

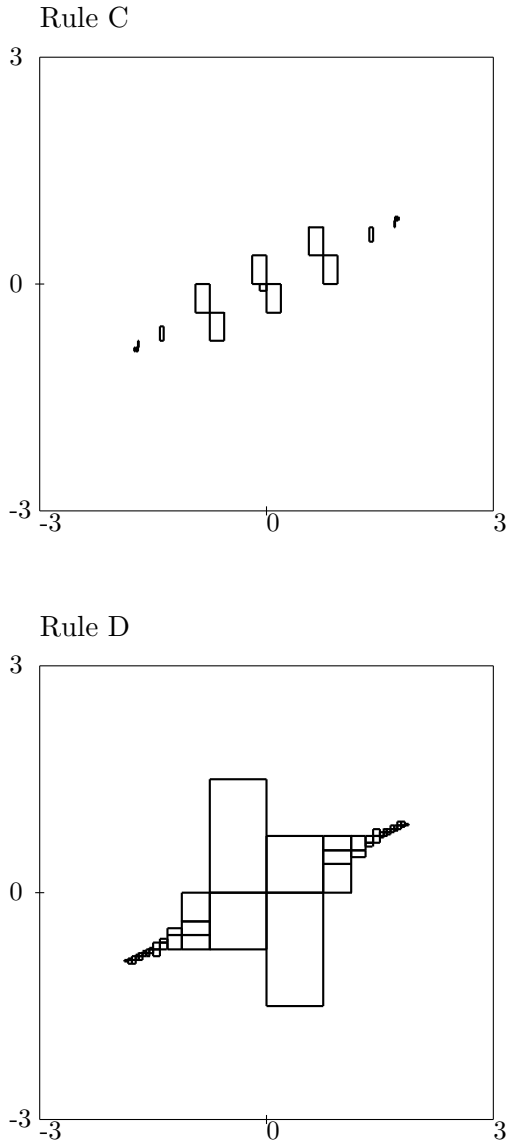

Figure 2. Remaining subintervals after 250 iteration steps of the model algorithm with the direction selection rules C, and D for the Three-Hump-Camel-Back problem [3].

This rule may decrease the *excess width* $w(\boldsymbol{\phi}(\boldsymbol{x})) - w(\phi^u(\boldsymbol{x}))$ of the inclusion function (where $\phi^u(\boldsymbol{x})$ is the range of ϕ on $\boldsymbol{x}$) that is

mignitude
excess width

caused in part by the floating point computer representation of real numbers. Consider the case when the component widths are of similar order, and the absolute value of one component is dominant. The subdivision of the latter component may result in a worse inclusion, since the representable numbers are sparser in this direction.

Rule E. Similar to Rule C, the underlying idea of Rule E is to minimize the width of the inclusion, but this time based on second order information (suggested by D. Ratz [10]):

$$D(i) := w((\boldsymbol{x}_i - m(\boldsymbol{x}_i))(\boldsymbol{\nabla}\boldsymbol{\phi}_i(m(\boldsymbol{x})) + \frac{1}{2}\sum_{j=1}^{n}(\boldsymbol{H}_{ij}(\boldsymbol{x})(\boldsymbol{x}_i - m(\boldsymbol{x}_i))). \quad (6)$$

Many interval optimization codes use *automatic differentiation* to produce the gradient and Hessian values. For such an implementation the subdivision selection rule E requires not much overhead.

Properties of direction selection rules.

Both the theoretical and numerical properties of subdivision direction selection rules have been studied extensively [1, 3, 4, 10, 11]. The exact definitions, theorems and details of numerical comparison tests can be found in these papers. Denote the global minimum value by ϕ^*.

Theoretical properties. In [4] the property of *balanced direction selection* has been defined. A subdivision direction selection rule is balanced basically if the B&B algorithm with this direction selection rule will not be unfair with any coordinate direction: each direction will be selected an infinite number of times in each *infinite subdivision sequence* of the leading boxes generated by the optimization algorithm. A global minimizer point $x' \in \boldsymbol{x}^0$ is called *hidden global minimizer point*, if there exists a subbox $\boldsymbol{x}' \subseteq \boldsymbol{x}^0$ with positive volume for which $x' \in \boldsymbol{x}'$ and $\underline{\boldsymbol{\phi}}(\boldsymbol{x}') = \phi^*$ while there exists an other global minimizer point x'' of the same problem such that $\underline{\boldsymbol{\phi}}(\boldsymbol{x}'') < \phi^*$ holds for each subbox $\boldsymbol{x}'' \subseteq \boldsymbol{x}^0$ with positive volume that contains x'' [11]. Now the following statements can be made:

1. The basic branch-and-bound algorithm converges in the sense that $\lim_{s\to\infty} w(\boldsymbol{x}^s) = 0$ if and only if the interval subdivision selection rule is balanced [4] (where $\boldsymbol{x}^s$ is the leading box of the algorithm in the iteration step number s).

2. Assume that the subdivision direction selection rule is balanced. Then the basic B&B algorithm converges to global minimizer points in the sense that $\lim_{s\to\infty} \boldsymbol{\phi}(\boldsymbol{x}^s) = \phi^*$, the set of accumulation points A of the leading box sequence is not empty, and A contains only global minimizer points.

3. Assume that the optimization algorithm converges for a given problem in the sense that $\lim_{s\to\infty} \boldsymbol{\phi}(\boldsymbol{x}^s) = \phi^*$. Then either the algorithm proceeds on the problem as one with a balanced direction-selection rule, or there exists a box $\boldsymbol{y}$ such that $\phi(x) = \phi^*$ for all $x \in \boldsymbol{y}$, and $w(\boldsymbol{y}_i) > 0$ $(i = 1, 2, \ldots, n)$ for all coordinate directions that are selected only a finite number of times.

4. The subdivision selection rules A and D are balanced, and thus the related algorithms converge to global minimizer points.

5. Either the subdivision selection rules B and C choose each direction an infinite number of times (they behave as balanced), or the related algorithms converge to a positive width subinterval of the search region $\boldsymbol{x_0}$ that contains only global minimizer points.

6. Sonja Berner proved that the basic algorithm is convergent with Rule E in the sense of $\lim_{s\to\infty} \boldsymbol{\phi}(\boldsymbol{x}^s) = \phi^*$, if an additional condition holds for the inclusion function [1].

7. If the branch-and-bound algorithm with any of the direction selection rules A – E converges to a global minimizer point, then it converges to all non-hidden global minimizer points [11].

Numerical properties. The numerical comparison tests were carried out on a wide set of

D. Ratz
automatic differentiation
balanced direction selection
infinite subdivision sequence
hidden global minimizer point
Sonja Berner

test problems and in several computational environments. The set of numerical test problems contained the standard global optimization test problems [3, 4], the set of problems studied in [5], and also some additional ones [10, 11]. The computing environments include IBM RISC 6000-580 and HP 9000/730 workstations and Pentium PC-s. The programs were coded in FORTRAN-90, PASCAL-XSC, and also in C++. The tests were carried out both with simple natural interval extension and with more sophisticated inclusion functions involving centered forms. The derivatives were handcoded in some test [4], while they were generated by automatic differentiation in the others [3, 10, 11]. The range of the investigated algorithms included simple B&B procedures and also optimization codes with many acceleration devices (like the **Interval Newton method**).

The conclusions were essentially the same: the rules B, C, and E had similar, substantial efficiency improvements against rules A and D, and these improvements were the greater the more difficult the solved problem was. The average performance of Rule D was the worst. Rule C was usually the best, closely followed by Rule B and E. It seems that the use of Rule E is justified only if the second derivatives are calculated also for other purposes. The numerical results were diverse, thus if the user has a characteristic problem set, then it is worth to test all the subdivision direction selection rules to find the most fitting one.

A computationally intensive numerical study [2] has proven that the most efficient subdivision direction selection rules are not those that minimize the width of the objective function inclusions for the result subintervals (which was the common belief), but those that maximize the lower bound of the worse subinterval obtained or minimize the width of the intersection of the result subintervals. The decisions of these a posteriori rules coincide the most with the a priori rules B, C, and E. These findings confirm the earlier mentioned numerical efficiency results.

Interval Newton method

References

[1] Berner, S.: 'New results on verified global optimization', *Computing* **57** (1996), 323–343.

[2] Csendes, T., Klatte, R., and Ratz, D.: 'A Posteriori Direction Selection Rules for Interval Optimization Methods', *CEJOR* **8** (2000), 225–236.

[3] Csendes, T. and Ratz, D.: 'A review of subdivision direction selection in interval methods for global optimization', *ZAMM* **76** (1996), 319–322.

[4] Csendes, T. and Ratz, D.: 'Subdivision direction selection in interval methods for global optimization', *SIAM J. Numerical Analysis* **34** (1997), 922–938.

[5] Hansen, E.: *Global Optimization Using Interval Analysis*, Marcel Decker, 1992.

[6] Kearfott, R.B.: *Rigorous global search: continuous problems*, Kluwer Acad. Publ., 1996.

[7] Kearfott, R.B. and Novoa, M.: 'INTBIS, a Portable Interval Newton/Bisection Package', *ACM T. on Mathematical Software* **16** (1990), 152–157.

[8] Ratschek, H. and Rokne, J.: *New Computer Methods for Global Optimization*, Ellis Horwood, 1988.

[9] Ratz, D.: *Automatische Ergebnisverifikation bei globalen Optimierungsproblemen, Dissertation*, Universität Karlsruhe, 1992.

[10] Ratz, D.: 'On Branching Rules in Second-Order Branch-and-Bound Methods for Global Optimization', in G. Alefeld, A. Frommer, and B. Lang (eds.): *Scientific Computing and Validated Numerics*, Akademie-Verlag, 1996, pp. 221–227.

[11] Ratz, D. and Csendes, T.: 'On the selection of Subdivision Directions in Interval Branch-and-Bound Methods for Global Optimization', *J. Global Optimization* **7** (1995), 183–207.

Tibor Csendes
University of Szeged
H-6701 Szeged, Pf. 652
Hungary
E-mail address: csendes@inf.u-szeged.hu

AMS 1991 Subject Classification: 65K05, 90C30.
Key words and phrases: branch-and-bound, interval arithmetic, optimization, subdivision direction.