Improving Interval Methods for Global Optimization[3]

A.E. Csallner and M. Cs. Markót

The global optimization problem can be defined in general as follows:

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

where X is a — possibly multidimensional — interval. If we denote the set of real intervals by $\mathbb{I}$ and $f : \mathbb{R}^n \to \mathbb{R}$ is the objective function of the problem, then $X \in \mathbb{I}^n$. Note, that a great class of real-life bound-constrained global optimization problems are covered by (1), e.g., problems where the parameters are given with tolerances or if the optimizers are supposed to be inside a parameter region [2].

Problem (1) can be solved with verified accuracy with the aid of interval methods (see, e.g., [1, 3, 4, 5, 6]). These methods are based on the well-known branch-and-bound principle. Thus, a search tree is built where the whole search region — the interval X — is the root and the particular levels consist of subintervals which are partitions of their parents in the tree. Those branches that cannot be pruned have to be stored for later treatment. The kind of the storage method used can be of great importance in performance if the increase of the number of intervals to be stored is considerable. The current presentation deals with these possibilities, i.e., the list handling of interval methods for global optimization.

Because the efficiency of a branch-and-bound method depends highly on which branch, i.e., stored list element is chosen as next to be treated, two basically different principles can be applied to handle the list. The first is to keep the list ordered and always pick up the first (or last) element, the second is to let the list be unordered and search for the next element in each step a new one is needed. The former saves computational time at picking up the elements, the latter at putting them onto the list. The time necessary for list handling can be calculated in both cases.

There are different principles to realize interval subdivision methods independently from the list handling. These variations can influence also the choice of the kind of list handling. There exist algorithms where the natural ordering of the list is the proper ordering for choosing the new elements to be picked up, hence, the list becomes a FIFO list.

A further possibility is to keep a part consisting of a constant number of list elements ordered, and refresh this part from time to time, e.g., after each list operation or when the ordered part becomes empty.

The oral presentation discusses all possibilities mentioned above thoroughly.

References

[1] Alefeld G. and Herzberger J. (1983), *Introduction to Interval Computations*, Academic Press, New York.

[2] Csallner A.E. (1993), *Global Optimization in Separation Network Synthesis*, Hungarian Journal of Industrial Chemistry, 21, pp. 303–308.

[3] Hansen E.R. (1992), *Global Optimization Using Interval Analysis*, Marcel Dekker, New York.

[4] Kearfott R.B. (1996), *Rigorous Global Search: Continuous Problems*, Kluwer, Dordrecht.

[5] Moore, R.E. (1966), *Interval Analysis*, Prentice Hall, Englewood Cliffs NJ.

[6] Ratschek H. and Rokne J. (1993), *Interval Methods*, In: Handbook of Global Optimization, Horst R. and Pardalos P.M. (eds.), Kluwer, Dordrecht, pp. 751–828.

[3]This work has been supported by the Grants OTKA T017241, F025743, and FKFP 0739/1997.