An Almost Embarrasingly Parallel Interval Global Optimization Method

Ole Caprani and Kaj Madsen

Extended Abstract

The well known Branch and Bound interval search methods of Skelboe, Moore and Hansen can be applied to find the global minimum and all of the global minimizers of a non-linear function $f : D \rightarrow \mathcal{R}$, where D is an n-dimensional interval. During the search we have:

- a finite number of subintervals S_i of D (called *boxes*). This set of boxes is called the *candidate set* and denoted S.
- lower bounds, $LB(S_i)$, on $min\{f(\mathbf{x}) \mid \mathbf{x} \in S_i\}$, e.g. obtained as inf $F(S_i)$ where F is an interval extension of f.
- an upper bound f_{bound} on the global minimum $f^* = min\{f(\mathbf{x}) \mid \mathbf{x} \in D\}$.

Any global minimizer is contained in the union of all the boxes from S :

$$G = \{\mathbf{x} \in D \mid f^* = f(\mathbf{x})\} \subseteq \bigcup_S S_i \subseteq D$$

and furthermore we have for all S_i that:

$$LB(S_i) \leq f^* \leq f_{bound}$$

A subbox S_i of S is marked as *terminated* if either the width of S_i is less than a given accuracy ϵ or the width of $F(S_i)$ is less than ϵ. The purpose of the search is to try to reduce the total volume of S until S consists of only terminated subboxes. This is done as follows: At each iteration of the search the box with the smallest lower bound is selected from the non-terminated subboxes of S (Best-First), split into two subboxes (bisection), for each subbox a lower bound is obtained, each subbox is either eliminated (because the lower bound is greater than f_{bound}), marked as terminated and inserted into S or put back into the set S. Also f_{bound} may be updated and further elements of the candidate set eliminated. The search stops when S consists of only terminated subboxes.

This method was first suggested by Skelboe, [1]. The convergence properties of this method have been extensively studied by Ratschek and Rokne, [3]. This search method has been refined by Moore, [2] and Hansen, [4], among others, with what Ratschek and Rokne, [3], have called *accelerating devices*. This is monotonicity test, concavity test, and search for stationary points by means of e.g. an interval Newton method.

Attempts to parallize the Branch and Bound method have been reported e.g. in [6], [7], [8] and [9]. All of these attempts use dynamic exchange of subboxes during the search to ensure a uniform workload on all processors and furthermore when a processor finds a better value of f_{bound} this value is broadcast to the other processors.

A parallel Branch and Bound method is described in [10] with an initial static distribution of work and no communication among processors during the search. When applied to three combinatorial optimization problems, an average processor utilization of 0.70-0.85 was observed. This suggests that it is worth to consider the method of [10] as a simple parallel Branch and Bound method also for interval global optimization. The method of [10] works as follows:

1. Find a good initial value for the upper bound f_{bound}.

2. Subdivide the original box D by use of a Best-First serial Branch and Bound method to obtain a candidate set S with a sufficient number of non-terminated subboxes, e.g. an order of magnitude higher than the number of processors available, p.

3. Create p candidate sets from S of approximately equal difficulty.

4. Distribute the p candidate sets and the upper bound to the p processors.

5. Perform serial Branch and Bound on each processor until the local candidate sets consist of only terminated subboxes.

6. Collect the terminated candidate sets from the p processors.

Three factors can influence the efficiency obtained by this parallel method. First, the fraction of time spent in the serial portion, i.e. step 1,2,3,4 and 6, should be small compared to the parallelized portion in step 5.

Second, if the initial value for f_{bound} is larger than f^* the number of subboxes processed in step 5 of the different processors may be larger than the number of subboxes processed by the serial Best-First Branch and Bound, hence, more work is performed by the parallel method.

Third, the workload at the processors may not be evenly distributed. For hard global optimization problems, e.g. induced by high dimensionality or hugh numbers of local and global minimizers, the first factor can be ignored since the time it takes to find a good upper bound can be assumed to be only a few percent of the total time it takes to process the candidate set, [8]. Furthermore, for hard problems the number of subboxes generated in step 2 is only a small fraction of the total number of subboxes processed in step 5. The inefficiency caused by the second factor cannot be ignored. For global optimization this was reported in [8] and [6], for combinatorial problems in [11]. The number of extra subboxes seems to explode exponentially even for a small difference between f_{bound} and f^*. In [11] it is argued that two things can be done to prevent this growth: Use a little more time to find a good initial bound and broadcast better bounds when obtained during step 5. The third factor can be ignored for the reported combinatorial optimization problems. The results given in [11] show that if the number of subboxes generated in step 2 is increased to several magnitudes higher than the number of processors, the inefficiencies caused by uneven workload can be ignored. There is no reason to believe that things are different for interval global optimization.

Thus, the parallel interval global optimization method that this discussion suggest is to find a good upper bound and enough subboxes in an initial phase, to avoid dynamic workload distribution during the parallel search, and to exchange better values of the upper bound during the search. The advantages of this almost embarrassingly parallel Branch and Bound with static workload is ease of programming, low communication volume, easy termination detection, easy adaption to different artitectures and scalability with the number of processors, [11].

References

[1] S. Skelboe, *Computation of Rational Interval Functions*, BIT, 14, pp. 87-95, 1974.

[2] R. E. Moore, *Methods and Applications of Interval Analysis*, SIAM, Philadelphia, 1979.

[3] H. Ratschek and J. Rokne, *New Computer Methods for Global Optimization*, John Wiley & Sons, 1988.

[4] E. Hansen, *Global Optimization Using Interval Analysis - the Multidimensional Case*, Numerische Mathematik, 34, pp. 247-270, 1980.

[5] E. Hansen, *Global Optimization Using Interval Analysis*, Marcel Dekker, Inc., 1992.

[6] K. Madsen, *Parallel Algorithms for Global Optimization*, Report 91-07, Institute for Numerical Analysis, The Technical University of Denmark, Lyngby, Denmark,1991.

[7] T. Henriksen and K. Madsen, *Use of Depth-First Strategy in Parallel Global Optimization*, Institute for Numerical Analysis, Report 92-10, The Technical University of Denmark, Lyngby, Denmark, 1992.

[8] O. Caprani, Brian Godthaab and K. Madsen, *Use of a Real-Valued Local Minimum in Parallel Interval Global Optimization*, Interval Computations, No 2, 1993.

[9] S. Berner, *Parallel Methods for Verified Global Optimization Practice and Theory*, Journal of Global Optimization, 9, 1-22, 1996.

[10] P. S. Laursen, *Can Parallel Branch and Bound without Communication be Effective ?*, SIAM J. Optimization, Vol. 4, No. 2, 1994.

[11] J. Clausen, *Parallel Branch and Bound - Principles and Personal Experiences*, in A. Migdalas et al. (eds.), Parallel Computing in Optimizations, Kluwer Academic Publishers, 1997.